

C2LLM Technical Report: A New Frontier in Code Retrieval via Adaptive Cross-Attention Pooling

Jin Qin^{*,1} Zihan Liao^{*,1} Ziyin Zhang^{*,1,2}
 Hang Yu^{†,1} Peng Di^{†,1} Rui Wang^{†,2}

¹Ant Group ²Shanghai Jiao Tong University

¹{qj431428,liaozihan.lzh,hyu.hugo,dipeng.dp}@antgroup.com

²{daenerystargaryen,wangrui12}@sjtu.edu.cn

👉 <https://github.com/codefuse-ai/CodeFuse-Embeddings>
 😊 <https://huggingface.co/collections/codefuse-ai/codefuse-embeddings>

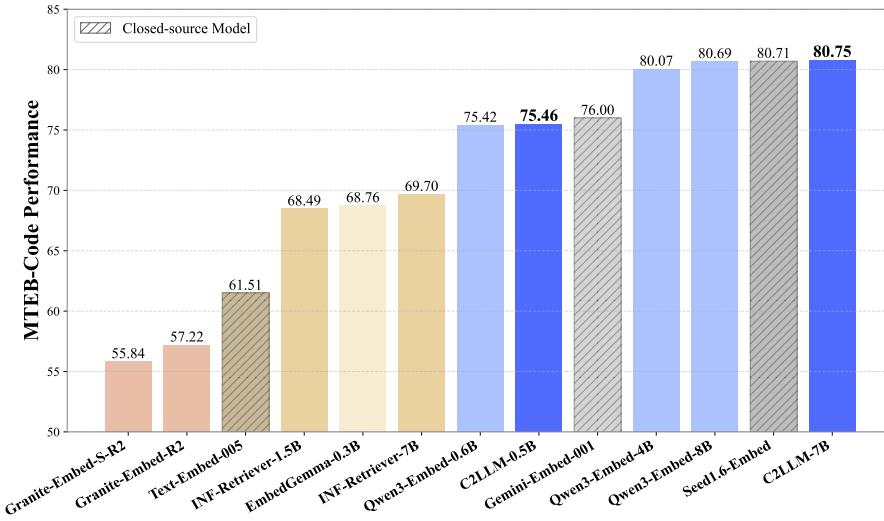


Figure 1: MTEB-Code leaderboard. C2LLM-7B ranks 1st among all models, surpassing the best closed-source models, while C2LLM-0.5B ranks 1st among models with less than 1B parameters, and 6th overall.

Abstract

We present C2LLM - Contrastive Code Large Language Models, a family of code embedding models in both 0.5B and 7B sizes. Building upon Qwen-2.5-Coder backbones, C2LLM adopts a Pooling by Multihead Attention (PMA) module for generating sequence embedding from token embeddings, effectively 1) utilizing the LLM’s causal representations acquired during pretraining, while also 2) being able to aggregate information from all tokens in the sequence, breaking the information bottleneck in EOS-based sequence embeddings, and 3) supporting flexible adaptation of embedding dimension, serving as an alternative to MRL. Trained on three million publicly available data, C2LLM models set new records on MTEB-Code among models of similar sizes, with C2LLM-7B ranking 1st on the overall leaderboard.

^{*}Equal Contribution.

[†]Correspondence to: Hang Yu <hyu.hugo@antgroup.com>, Peng Di <dipeng.dp@antgroup.com>, Rui Wang <wangrui12@sjtu.edu.cn>.

1 Introduction

Large language models (LLMs) pretrained on source code and natural language have rapidly advanced a wide spectrum of software engineering applications, including code generation, automated issue resolution, and, notably, code retrieval (Zhang et al., 2024b). In the retrieval setting, a user supplies a natural-language query (e.g., “open a jsonl file in Python and read all lines”), and the system must return the most relevant snippet among millions or even billions of candidates stored in public or private codebases. Code retrieval is not only essential for interactive developer search engines but also forms a pivotal step in the workflow of emerging code agents - autonomous systems that iteratively plan, search, and edit code to accomplish complex programming tasks (Yang et al., 2024; Gao et al., 2025; Tao et al., 2025; Wang et al., 2025).

At the core of code retrieval systems lie code embedding models. Despite the recent surge of general-purpose text embedding models (Zhang et al., 2025a; Lee et al., 2025a; Choi et al., 2025; Chen et al., 2024; Zhang et al., 2025b), directly transferring them to code embedding remains sub-optimal, as **popular pooling strategies are ill-suited to code**. State-of-the-art embedding models either adopt mean pooling over the outputs of an LLM (Lee et al., 2025a;b) or take the end-of-sequence (EOS) token representation as sequence embeddings (Choi et al., 2025; Zhang et al., 2025b). However, mean pooling is often paired with bidirectional attention, departing from the causal pretraining recipe of leading code LLMs (e.g. Qwen2.5-Coder, Hui et al., 2024) and therefore fails to unlock their full potential (Li et al., 2025b). Conversely, taking the EOS token embedding collapses all syntactic and semantic structure into one position, creating an information bottleneck that is especially harmful in the code domain, where input code files could easily contain thousands of tokens.

To address this challenge, we introduce Contrastive Code Large Language Models (C2LLM), a new code embedding model family optimized for code retrieval. **C2LLM preserves the causal attention of its backbone LLM but sidesteps the dilemma between mean pooling and EOS representation by inserting a lightweight Pooling by Multihead Attention (PMA) module** (Lee et al., 2019), which has been shown by Liao et al. (2024) to outperform both mean pooling and EOS representation. A single learnable query attends to all token representations produced by the LLM, simultaneously 1) aggregating sequence information into a single vector, and 2) providing support for dimensionality adaptation, making it ideal for real-world large-scale vector databases.

Trained on 3 million publicly available data, our 7B model achieves an average performance of 80.75 on MTEB-Code benchmark, ranking 1st among all models on the leaderboard. Our smaller model, with 0.5B parameters, scores 75.46 and pushes the frontier of models around 1B size, surpassing similar-sized competitors including Qwen3-Embedding-0.6B, EmbeddingGemma, and INF-Retriever. Our models are publicly available.

2 Related Work

In contrast to the abundance of text embedding models (Lee et al., 2025a; Zhang et al., 2025a;b), code-focused embedding research has received less attention in recent years. Most code embedding models adopt a BERT-based architecture, including CodeBERT (Feng et al., 2020), GraphCode-BERT (Guo et al., 2021), CodeSage (Zhang et al., 2024a), and CodeT5+ (Wang et al., 2023), which fail to utilize the power of Code LLMs pretrained on trillions of tokens. BGE-Code (Li et al., 2025a) and CodeXEmbed (Liu et al., 2024) represent two notable exceptions, which are based on Qwen2.5-Coder and Mistral. However, none of these models are present on the MTEB-Code leaderboard, which is dominated by general-purpose text embedding models such as Qwen3-Embedding (Zhang et al., 2025a), INF-Retriever (Yang et al., 2025), and EmbeddingGemma (Vera et al., 2025).

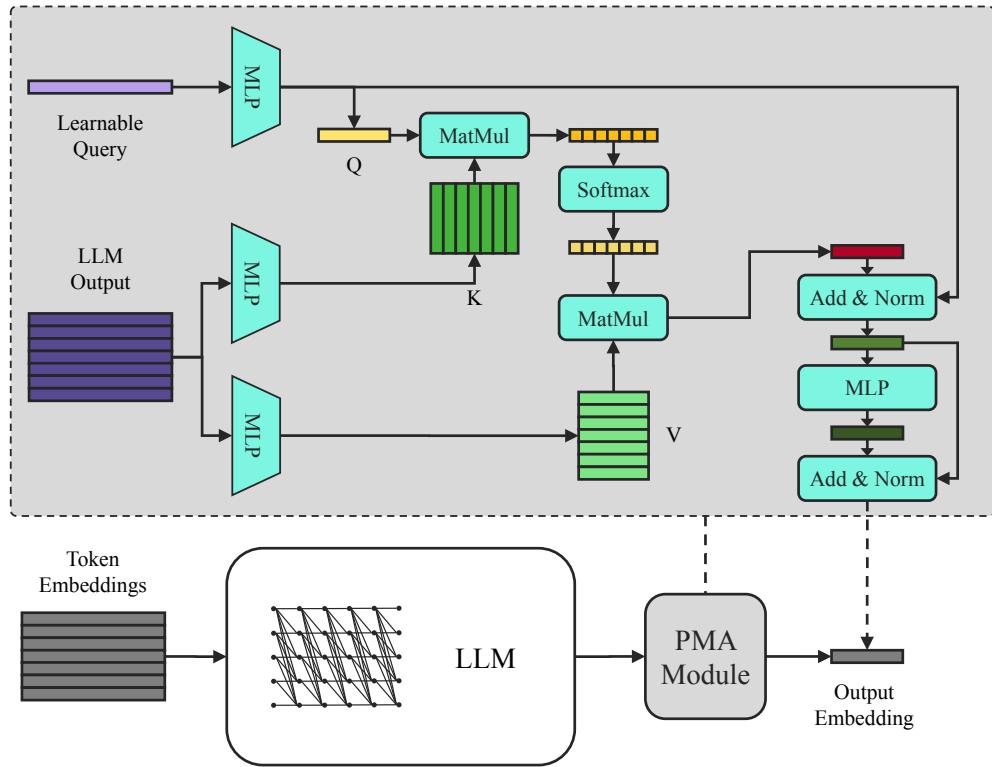


Figure 2: C2LLM Model architecture, comprising an LLM followed by a PMA (Pooling by Multihead Attention) module. PMA is a single layer of cross attention with one learnable query and takes the LLM’s last hidden states as KV, serving both to pool over the input sequence and to provide support for flexible embedding dimension. Multi-head mechanism is omitted in the illustration.

3 Model Architecture: Introducing Pooling by Multihead Attention into Embedding Models

Two of the most popular methods for obtaining an embedding from a token sequence are mean pooling (Lee et al., 2025b; Zhao et al., 2025; Yu et al., 2025) and taking the EOS token embedding (Zhang et al., 2025a; Choi et al., 2025). However, mean pooling is often paired with bidirectional attention, deviating from state-of-the-art LLMs’ pretraining design and thus being unable to fully exploit their potential (Li et al., 2025b), while EOS representation condenses information from the entire sequence into a single token, creating an information bottleneck. To circumvent this dilemma, NV-Embed (Lee et al., 2025a) introduced a latent attention layer on top of the LLM, using the LLM’s hidden states as query and a latent array of 512 vectors as key/value. This design, however, does not change the number of tokens and still requires mean pooling on the output.

In C2LLM, we propose yet another solution by introducing Pooling by Multihead Attention (PMA, Lee et al., 2019; Liao et al., 2024). As illustrated in Figure 2, PMA consists of a cross-attention layer with a single learnable query vector and takes the LLM’s last hidden states as key/value, effectively aggregating information from the token sequence into a single embedding vector. Apart from pooling over the sequence dimension, PMA can also reduce the embedding dimension at the same time, providing an alternative to MRL (Kusupati et al., 2022).

Formally, given the LLM’s hidden states for l input tokens $H \in \mathbb{R}^{l \times d_{\text{LLM}}}$ and the learnable query vector $q \in \mathbb{R}^{1 \times d_q}$, we first project them into lower dimensions:

$$Q^{1 \times d} = qW_q, \quad (1)$$

$$K^{l \times d} = HW_k, \quad (2)$$

$$V^{l \times d} = HW_v, \quad (3)$$

where $W_q \in \mathbb{R}^{d_q \times d}$, $W_k, W_v \in \mathbb{R}^{d_{\text{LLM}} \times d}$, and d is the output embedding dimension. Cross attention is then computed in this lower dimension with residual connections and layer normalization (LN):

$$O^{1 \times d} = \text{softmax}(QK^T)V, \quad (4)$$

$$\tilde{O}^{1 \times d} = \text{LN}(O + Q), \quad (5)$$

$$E^{1 \times d} = \text{LN}(\text{ReLU}(\tilde{O}W_o) + \tilde{O}). \quad (6)$$

E is then taken as the embedding for the input sequence.

Takeaway The integration of the PMA module into embedding models offers three primary advantages. First, unlike mean pooling and EOS representation, the cross-attention mechanism allows the model to learn which tokens (e.g., function signatures or key algorithmic logic) are most salient for the final representation. Second, it maintains both the foundational causal architecture and efficiency of the LLM backbone, as the PMA overhead is negligible compared to the billions of parameters in the LLM. Finally, by decoupling the LLM’s hidden dimension (d_{LLM}) from the final embedding dimension (d), PMA can produce compact embeddings suitable for vector databases without requiring the computationally expensive MRL training objective.

4 Experiments

4.1 Training Settings

Model Configurations We develop the C2LLM series by fine-tuning two state-of-the-art base models: Qwen2.5-Coder-0.5B-Instruct and Qwen2.5-Coder-7B-Instruct (Hui et al., 2024). The training data includes CodeSearchNet (including code-to-code, code-to-text, and text-to-code retrieval, Husain et al., 2019; Li et al., 2025c; Lu et al., 2021), APPS (Hendrycks et al., 2021), single-turn and multi-turn CodeFeedback (Zheng et al., 2024), CodeEditSearch (Muennighoff et al., 2024), CosQA (Huang et al., 2021), StackOverflowQA (Li et al., 2025c), SyntheticText2SQL (Meyer et al., 2024), and CodeTransOcean (Yan et al., 2023), totaling 3 million samples. For the model configurations, we employ PMA with 32 heads to aggregate token-level features into a single sequence representation. The fine-tuning process is made efficient through the use of LoRA (Hu et al., 2022), configured with a rank (r) of 64 and an alpha (α) of 32. To optimize computational throughput and memory usage, we utilize Flash Attention 2 (Dao, 2024) across all training stages.

Training Strategy The models are trained for 3 epochs with a learning rate of 1×10^{-4} and a maximum sequence length of 1024 tokens using left-padding. Our optimization strategy centers on contrastive learning. For in-batch contrastive learning, we implement a global batch strategy to synchronize samples across all distributed processes, effectively expanding the pool of negative samples. For hard-negative contrastive learning, we incorporate $K = 7$ hard negatives for each query. We apply a temperature scaling factor of $\tau = 0.05$ to both in-batch and hard-negative contrastive losses. To ensure the quality of the contrastive signals, we adopt a specialized batching strategy where data is grouped according to both the dataset source and the specific programming language before being partitioned into training batches. During the optimization process, a loss weight of 1 is assigned to all objectives, with the sole exception of the CodeEditSearch dataset, which uses a custom weight to balance its contribution. Finally, the definitive C2LLM model is produced by performing a weighted merge of four checkpoints captured at different global steps, a technique designed to enhance the stability and generalization of the final embeddings.

Prompt template The Prompt templates for each dataset are shown in Table 1.

Task Name	Query Instruction	Document Instruction
CodeEditSearchRetrieval	Retrieve the diff code that relevant the following query:	Retrieved Answer:
CodeSearchNetRetrieval	Retrieve the code that solves the following query:	Retrieved Answer:
AppsRetrieval	Given a problem description from a programming contest, retrieve code examples that can assist in solving it.	Retrieved Answer:
CodeFeedbackMT	Given a multi-turn conversation history that includes both text and code, retrieve relevant multimodal answers composed of text and code that address the ongoing discussion.	Retrieved Answer:
CodeFeedbackST	Given a single-turn question composed of text and code, retrieve suitable answers that also mix text and code to provide helpful feedback.	Retrieved Answer:
CodeSearchNetCCRetrieval	Given an initial code segment, retrieve the subsequent segment that continues the code.	Retrieved Answer:
CodeTransOceanContest	Given a Python code snippet, retrieve its semantically equivalent version written in C++.	Retrieved Answer:
CodeTransOceanDL	Given a code snippet, retrieve a semantically equivalent implementation of the same code.	Retrieved Answer:
COIRCodeSearchNetRetrieval	Given a code snippet, retrieve its corresponding document string that summarizes its functionality.	Retrieved Answer:
CosQA	Given a query from a web search, retrieve code that is helpful in addressing the query.	Retrieved Answer:
StackOverflowQA	Given a question combining text and code, retrieve relevant answers that also contain both text and code snippets and can address the question.	Retrieved Answer:
SyntheticText2SQL	Given a natural language question, retrieve SQL queries that serve as appropriate responses.	Retrieved Answer:

Table 1: Instructions for training data.

Model	Size	APPS	CodeSearchNet (CSN/CCR/CoIR)	CodeEdit	CodeFeedback (MT/ST)	CodeTransOcean (Contest/DL)	CosQA	Stack- OverflowQA	Synthetic- Text2SQL	Avg	Rank
C2LLM	7B	86.71	91.07/97.90/89.79	81.49	94.32/90.66	92.51/34.13	39.76	94.85	75.75	80.75	1
Seed1.6-Embed	NA	91.15	93.17/94.15/89.50	92.14	90.11/90.44	90.16/35.99	41.17	97.20	63.31	80.71	2
Qwen3-Embed	8B	91.07	92.66/96.35/89.51	76.97	93.70/89.93	93.73/32.81	38.04	94.75	78.75	80.69	3
Qwen3-Embed	4B	89.18	92.34/95.59/87.93	76.49	93.21/89.51	90.99/35.04	37.98	94.32	78.21	80.07	4
Gemini-Embed	NA	93.75	91.33/84.69/81.06	81.61	56.28/85.33	89.53/31.47	50.24	96.71	69.96	76.00	5
C2LLM	0.5B	61.02	89.20/96.29/86.71	71.39	92.29/88.63	84.27/33.99	38.30	89.40	74.08	75.46	6
Qwen3-Embed	0.6B	75.34	91.01/91.72/84.69	64.42	90.82/86.39	86.05/31.36	36.48	89.99	76.74	75.42	7
INF-Retriever	7B	47.37	88.77/75.71/72.27	71.79	77.64/86.63	89.16/35.18	34.18	94.22	63.51	69.70	8
EmbedGemma	0.3B	84.39	90.15/73.71/75.54	62.10	51.42/80.26	85.51/33.52	43.60	86.47	58.42	68.76	9
INF-Retriever	1.5B	38.90	90.87/75.50/78.63	67.17	77.47/84.51	85.01/33.84	33.11	91.32	65.59	68.49	10

Table 2: Top 10 models on the MTEB-Code leaderboard as of the submission date (2025-12-25). “NA” in the model size column indicates closed-source model whose size is not available.

4.2 Results

We evaluate C2LLM on the 12 retrieval tasks in MTEB-Code Benchmark (Muennighoff et al., 2023; Enevoldsen et al., 2025)¹. As shown in Table 2, C2LLM-7B achieves an average score of 80.75, surpassing the previous state-of-the-art Seed1.6-Embedding and Qwen3-Embedding-8B. Notably, C2LLM-7B shows superior performance in complex reasoning tasks such as CodeFeedback (94.32 for multi-turn, 90.66 for single turn), suggesting that the PMA module effectively captures the intent behind natural language queries directed at code.

Our smaller variant, C2LLM-0.5B, demonstrates remarkable efficiency. With only 0.5B parameters, it achieves an average score of 75.46, outperforming significantly larger models like INF-Retriever-7B (69.70). It also surpasses all other models with less than 1B parameters, establishing a new state-of-the-art in the compute-efficient regime. The consistent performance of C2LLM across both scales validates the robustness of using cross-attention as a universal pooling strategy for code embeddings.

5 Conclusion

We introduce C2LLM, a family of code embedding models that achieves state-of-the-art performance by combining the strengths of causal LLM pretraining with a flexible Pooling by Multihead Attention (PMA) module. Our results demonstrate that bypassing the historical dilemma between EOS and mean-pooling strategies allows for better information aggregation in representing code sequences, setting new records on the MTEB-Code benchmark with our 7B model.

C2LLM represents the fourth entry in the CodeFuse Embedding model family, following D2LLM (Liao et al., 2024), E2LLM (Liao et al., 2025), and F2LLM (Zhang et al., 2025b). We are

¹<https://huggingface.co/spaces/mteb/leaderboard>

dedicated to promoting open research in LLM-based embedding models, and plan to expand the series into massively multilingual and multi-domain scenarios in the near future.

References

- Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. BGE m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *CoRR*, abs/2402.03216, 2024. doi: 10.48550/ARXIV.2402.03216. URL <https://doi.org/10.48550/arXiv.2402.03216>.
- Jooyoung Choi, Hyun Kim, Hansol Jang, Changwook Jun, Kyunghoon Bae, Hyewon Choi, Stanley Jungkyu Choi, Honglak Lee, and Chulmin Yun. Lgai-embedding-preview technical report. *CoRR*, abs/2506.07438, 2025. doi: 10.48550/ARXIV.2506.07438. URL <https://doi.org/10.48550/arXiv.2506.07438>.
- Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL <https://openreview.net/forum?id=mZn2Xyh9Ec>.
- Kenneth C. Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Siblini, Dominik Krzeminski, Genta Indra Winata, Saba Sturua, Saiteja Utpala, Mathieu Ciancone, Marion Schaeffer, Diganta Misra, Shreeya Dhakal, Jonathan Rystrom, Roman Solomatin, Ömer Veysel Çagatan, Akash Kundu, and et al. MMTEB: massive multilingual text embedding benchmark. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL <https://openreview.net/forum?id=z13pfz4VCV>.
- Zhangyin Feng, Daya Guo, Duyu Tang, Nan Duan, Xiaocheng Feng, Ming Gong, Linjun Shou, Bing Qin, Ting Liu, Dixin Jiang, and Ming Zhou. Codebert: A pre-trained model for programming and natural languages. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pp. 1536–1547. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.FINDINGS-EMNLP.139. URL <https://doi.org/10.18653/v1/2020.findings-emnlp.139>.
- Pengfei Gao, Zhao Tian, Xiangxin Meng, Xinchen Wang, Ruida Hu, Yuanan Xiao, Yizhou Liu, Zhao Zhang, Junjie Chen, Cuiyun Gao, Yun Lin, Yingfei Xiong, Chao Peng, and Xia Liu. Trae agent: An llm-based agent for software engineering with test-time scaling. *CoRR*, abs/2507.23370, 2025. doi: 10.48550/ARXIV.2507.23370. URL <https://doi.org/10.48550/arXiv.2507.23370>.
- Daya Guo, Shuo Ren, Shuai Lu, Zhangyin Feng, Duyu Tang, Shujie Liu, Long Zhou, Nan Duan, Alexey Svyatkovskiy, Shengyu Fu, Michele Tufano, Shao Kun Deng, Colin B. Clement, Dawn Drain, Neel Sundaresan, Jian Yin, Dixin Jiang, and Ming Zhou. Graphcodebert: Pre-training code representations with data flow. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL <https://openreview.net/forum?id=jLoC4ez43PZ>.
- Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, and Jacob Steinhardt. Measuring coding challenge competence with APPS. In Joaquin Vanschoren and Sai-Kit Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL <https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/c24cd76e1ce41366a4bbe8a49b02a028-Abstract-round2.html>.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL <https://openreview.net/forum?id=nZeVKeeFYf9>.

Junjie Huang, Duyu Tang, Linjun Shou, Ming Gong, Ke Xu, Daxin Jiang, Ming Zhou, and Nan Duan. Cosqa: 20, 000+ web queries for code search and question answering. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *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. 5690–5700. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.ACL-LONG.442. URL <https://doi.org/10.18653/v1/2021.acl-long.442>.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Kai Dang, An Yang, Rui Men, Fei Huang, Xingzhang Ren, Xuancheng Ren, Jingren Zhou, and Junyang Lin. Qwen2.5-coder technical report. *CoRR*, abs/2409.12186, 2024. doi: 10.48550/ARXIV.2409.12186. URL <https://doi.org/10.48550/arXiv.2409.12186>.

Hamel Husain, Ho-Hsiang Wu, Tiferet Gazit, Miltiadis Allamanis, and Marc Brockschmidt. Code-searchnet challenge: Evaluating the state of semantic code search. *CoRR*, abs/1909.09436, 2019. URL <http://arxiv.org/abs/1909.09436>.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham M. Kakade, Prateek Jain, and Ali Farhadji. Matryoshka representation learning. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*. URL http://papers.nips.cc/paper_files/paper/2022/hash/c32319f4868da7613d78af9993100e42-Abstract-Conference.html.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Nv-embed: Improved techniques for training llms as generalist embedding models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025a. URL <https://openreview.net/forum?id=lgSYLSsDRe>.

Jinhyuk Lee, Feiyang Chen, Sahil Dua, Daniel Cer, Madhuri Shanbhogue, Iftekhar Naim, Gustavo Hernández Ábrego, Zhe Li, Kaifeng Chen, Henrique Schechter Vera, Xiaoqi Ren, Shanfeng Zhang, Daniel Salz, Michael Boratko, Jay Han, Blair Chen, Shuo Huang, Vikram Rao, Paul Suganthan, Feng Han, Andreas Doumanoglou, Nithi Gupta, Fedor Moiseev, Cathy Yip, Aashi Jain, Simon Baumgartner, Shahrokh Shahi, Frank Palma Gomez, Sandeep Mariserla, Min Choi, Parashar Shah, Sonam Goenka, Ke Chen, Ye Xia, Koert Chen, Sai Meher Karthik Duddu, Yichang Chen, Trevor Walker, Wenlei Zhou, Rakesh Ghiya, Zach Gleicher, Karan Gill, Zhe Dong, Mojtaba Seyedhosseini, Yun-Hsuan Sung, Raphael Hoffmann, and Tom Duerig. Gemini embedding: Generalizable embeddings from gemini. *CoRR*, abs/2503.07891, 2025b. doi: 10.48550/ARXIV.2503.07891. URL <https://doi.org/10.48550/arXiv.2503.07891>.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3744–3753. PMLR, 2019. URL <http://proceedings.mlr.press/v97/lee19d.html>.

Chaofan Li, Jianlyu Chen, Yingxia Shao, Defu Lian, and Zheng Liu. Towards A generalist code embedding model based on massive data synthesis. *CoRR*, abs/2505.12697, 2025a. doi: 10.48550/ARXIV.2505.12697. URL <https://doi.org/10.48550/arXiv.2505.12697>.

Chaofan Li, Minghao Qin, Shitao Xiao, Jianlyu Chen, Kun Luo, Defu Lian, Yingxia Shao, and Zheng Liu. Making text embedders few-shot learners. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025b. URL <https://openreview.net/forum?id=wfLuiDjQ0u>.

Xiangyang Li, Kuicai Dong, Yi Quan Lee, Wei Xia, Hao Zhang, Xinyi Dai, Yasheng Wang, and Ruiming Tang. Coir: A comprehensive benchmark for code information retrieval models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 22074–22091. Association for Computational Linguistics, 2025c. URL <https://aclanthology.org/2025.acl-long.1072/>.

Zihan Liao, Hang Yu, Jianguo Li, Jun Wang, and Wei Zhang. D2LLM: decomposed and distilled large language models for semantic search. In Lun-Wei Ku, Andre Martins, and Vivek Srikanth (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 14798–14814. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.791. URL <https://doi.org/10.18653/v1/2024.acl-long.791>.

Zihan Liao, Jun Wang, Hang Yu, Lingxiao Wei, Jianguo Li, Jun Wang, and Wei Zhang. E2LLM: Encoder elongated large language models for long-context understanding and reasoning. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 19212–19241, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.970. URL <https://aclanthology.org/2025.emnlp-main.970/>.

Ye Liu, Rui Meng, Shafiq Joty, Silvio Savarese, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. Codexembed: A generalist embedding model family for multilingual and multi-task code retrieval. *CoRR*, abs/2411.12644, 2024. doi: 10.48550/ARXIV.2411.12644. URL <https://doi.org/10.48550/arXiv.2411.12644>.

Shuai Lu, Daya Guo, Shuo Ren, Junjie Huang, Alexey Svyatkovskiy, Ambrosio Blanco, Colin B. Clement, Dawn Drain, Dixin Jiang, Duyu Tang, Ge Li, Lidong Zhou, Linjun Shou, Long Zhou, Michele Tufano, Ming Gong, Ming Zhou, Nan Duan, Neel Sundaresan, Shao Kun Deng, Shengyu Fu, and Shujie Liu. Codexglue: A machine learning benchmark dataset for code understanding and generation. In Joaquin Vanschoren and Sai-Kit Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL <https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/c16a5320fa475530d9583c34fd356ef5-Abstract-round1.html>.

Yev Meyer, Marjan Emadi, Dhruv Nathawani, Lipika Ramaswamy, Kendrick Boyd, Maarten Van Segbroeck, Matthew Grossman, Piotr Mlocek, and Drew Newberry. Synthetic-Text-To-SQL: A synthetic dataset for training language models to generate sql queries from natural language prompts, April 2024. URL <https://huggingface.co/datasets/gretelai/synthetic-text-to-sql>.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. MTEB: massive text embedding benchmark. In Andreas Vlachos and Isabelle Augenstein (eds.), *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pp. 2006–2029. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EACL-MAIN.148. URL <https://doi.org/10.18653/v1/2023.eacl-main.148>.

Niklas Muennighoff, Qian Liu, Armel Randy Zebaze, Qinkai Zheng, Binyuan Hui, Terry Yue Zhuo, Swayam Singh, Xiangru Tang, Leandro von Werra, and Shayne Longpre. Octopack: Instruction tuning code large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL <https://openreview.net/forum?id=mw1PWNSWZP>.

Hongyuan Tao, Ying Zhang, Zhenhao Tang, Hongen Peng, Xukun Zhu, Bingchang Liu, Yingguang Yang, Ziyin Zhang, Zhaogui Xu, Haipeng Zhang, Linchao Zhu, Rui Wang, Hang Yu, Jianguo Li, and Peng Di. Code graph model (CGM): A graph-integrated large language model for repository-level software engineering tasks. *CoRR*, abs/2505.16901, 2025. doi: 10.48550/ARXIV.2505.16901. URL <https://doi.org/10.48550/arXiv.2505.16901>.

Henrique Schechter Vera, Sahil Dua, Biao Zhang, Daniel Salz, Ryan Mullins, Sindhu Raghuram Panyam, Sara Smoot, Iftekhar Naim, Joe Zou, Feiyang Chen, Daniel Cer, Alice Lisak, Min Choi, Lucas Gonzalez, Omar Sanseviero, Glenn Cameron, Ian Ballantyne, Kat Black, Kaifeng Chen, Weiyi Wang, Zhe Li, Gus Martins, Jinhyuk Lee, Mark Sherwood, Ju-yeong Ji, Renjie Wu, Jingxiao Zheng, Jyotinder Singh, Abheesht Sharma, Divyashree Sreepathihalli, Aashi Jain, Adham Elarabawy, AJ Co, Andreas Doumanoglou, Babak Samari, Ben Hora, Brian Potetz, Dahun Kim, Enrique Alfonseca, Fedor Moiseev, Feng Han, Frank Palma Gomez, Gustavo Hernández Ábreo, Hesen Zhang, Hui Hui, Jay Han, Karan Gill, Ke Chen, Koert Chen, Madhuri Shanbhogue, Michael Boratko, Paul Suganthan, Sai Meher Karthik Duddu, Sandeep Mariserla, Setareh Ariafar, Shanfeng Zhang, Shijie Zhang, Simon Baumgartner, Sonam Goenka, Steve Qiu, Tamaya Dabral, Trevor Walker, Vikram Rao, Waleed Khawaja, Wenlei Zhou, Xiaoqi Ren, Ye Xia, Yichang Chen, Yi-Ting Chen, Zhe Dong, Zhongli Ding, Francesco Visin, Gaël Liu, Jiageng Zhang, Kathleen Kenealy, Michelle Casbon, Ravin Kumar, Thomas Mesnard, Zach Gleicher, Cormac Brick, Olivier Lacombe, Adam Roberts, Qin Yin, Yun-Hsuan Sung, Raphael Hoffmann, Tris Warkentin, Armand Joulin, Tom Duerig, and Mojtaba Seyedhosseini. Embeddinggemma: Powerful and lightweight text representations. *CoRR*, abs/2509.20354, 2025. doi: 10.48550/ARXIV.2509.20354. URL <https://doi.org/10.48550/arXiv.2509.20354>.

Xingyao Wang, Boxuan Li, Yufan Song, Frank F. Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, Hoang H. Tran, Fuqiang Li, Ren Ma, Mingzhang Zheng, Bill Qian, Yanjun Shao, Niklas Muennighoff, Yizhe Zhang, Binyuan Hui, Junyang Lin, and et al. Openhands: An open platform for AI software developers as generalist agents. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL <https://openreview.net/forum?id=0Jd3ayDDoF>.

Yue Wang, Hung Le, Akhilesh Gotmare, Nghi D. Q. Bui, Junnan Li, and Steven C. H. Hoi. Codet5+: Open code large language models for code understanding and generation. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 1069–1088. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.68. URL <https://doi.org/10.18653/v1/2023.emnlp-main.68>.

Weixiang Yan, Yuchen Tian, Yunzhe Li, Qian Chen, and Wen Wang. Codetransocean: A comprehensive multilingual benchmark for code translation. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pp. 5067–5089. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-EMNLP.337. URL <https://doi.org/10.18653/v1/2023.findings-emnlp.337>.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/5a7c947568c1b1328ccc5230172e1e7c-Abstract-Conference.html.

Junhan Yang, Jiahe Wan, Yichen Yao, Wei Chu, Yinghui Xu, and Yuan Qi. inf-retriever-v1 (revision 5f469d7), 2025. URL <https://huggingface.co/infly/inf-retriever-v1>.

Peng Yu, En Xu, Bin Chen, Haibiao Chen, and Yafei Xu. Qzhou-embedding technical report. *CoRR*, abs/2506.21632, 2025. doi: 10.48550/ARXIV.2506.21632. URL <https://doi.org/10.48550/arXiv.2506.21632>.

Dejiao Zhang, Wasi Uddin Ahmad, Ming Tan, Hantian Ding, Ramesh Nallapati, Dan Roth, Xiaofei Ma, and Bing Xiang. Code representation learning at scale. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024a. URL <https://openreview.net/forum?id=vfzRRjumpX>.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *CoRR*, abs/2506.05176, 2025a. doi: 10.48550/ARXIV.2506.05176. URL <https://doi.org/10.48550/arXiv.2506.05176>.

Ziyin Zhang, Chaoyu Chen, Bingchang Liu, Cong Liao, Zi Gong, Hang Yu, Jianguo Li, and Rui Wang. Unifying the perspectives of NLP and software engineering: A survey on language models for code. *Trans. Mach. Learn. Res.*, 2024, 2024b. URL <https://openreview.net/forum?id=hkNnGqZnpa>.

Ziyin Zhang, Zihan Liao, Hang Yu, Peng Di, and Rui Wang. F2LLM technical report: Matching SOTA embedding performance with 6 million open-source data. *CoRR*, abs/2510.02294, 2025b. doi: 10.48550/ARXIV.2510.02294. URL <https://doi.org/10.48550/arXiv.2510.02294>.

Xinping Zhao, Xinshuo Hu, Zifei Shan, Shouzheng Huang, Yao Zhou, Zetian Sun, Zhenyu Liu, Dongfang Li, Xinyuan Wei, Qian Chen, Youcheng Pan, Yang Xiang, Meishan Zhang, Haofen Wang, Jun Yu, Baotian Hu, and Min Zhang. Kalm-embedding-v2: Superior training techniques and data inspire A versatile embedding model. *CoRR*, abs/2506.20923, 2025. doi: 10.48550/ARXIV.2506.20923. URL <https://doi.org/10.48550/arXiv.2506.20923>.

Tianyu Zheng, Ge Zhang, Tianhao Shen, Xueling Liu, Bill Yuchen Lin, Jie Fu, Wenhua Chen, and Xiang Yue. Opencodeinterpreter: Integrating code generation with execution and refinement. In Lun-Wei Ku, Andre Martins, and Vivek Srikanth (eds.), *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pp. 12834–12859. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-ACL.762. URL <https://doi.org/10.18653/v1/2024.findings-acl.762>.