Capturing LLM Cases for RAG Reuse

Lake Yin, David Leake

Luddy School of Informatics, Computing, and Engineering, Indiana University, Bloomington, IN, USA

Abstract

This paper presents initial research aimed at increasing accuracy and consistency of LLM responses by capturing cases from LLM reasoning and reusing them as context for future LLM queries. This approach gives LLMs a simple memory mechanism for lifelong learning from feedback and provides a practical form of retrieval augmented generation (RAG) that does not require a preexisting dataset. We implemented case-based processes with two different levels of feedback, tested them with multiple LLM models, and evaluated their accuracy on problems from the Massive Multitask Language Understanding benchmark. Our experiments suggest that capturing LLM cases and feedback for RAG reuse is a promising approach for improving LLM performance over time.

Keywords

Case-based reasoning, Large Language Models, Memory, Retrieval-Augmented Generation

1. Introduction

As LLMs have gained prominence, interest has grown in enhancing their processing with memory capabilities from case-based reasoning [1, 2]. This paper proposes capturing cases containing queries, responses, and feedback during LLM processing to provide to the LLM as context when solving similar future problems. This process can be seen as a form of retrieval-augmented generation [3] based on LLM problem-solving experience. We hypothesize two potential benefits of such an approach: first, increasing system accuracy, and, second, supporting solution consistency, by biasing the LLM toward solutions similar to those provided in the past. Increased consistency could in turn contribute to solution predictability, supporting trust [4].

This approach exploits the natural learning process of CBR for a form of LLM lifelong learning without retraining the LLM. Consistent with CBR, and in contrast to LLM training, the case learning is inertia-free. In contrast to standard RAG, which requires a preexisting database, this approach enables the LLM to gather its own set of cases describing past solutions. Because it can gather and reuse information about both successes and failures, it can exploit the ability of CBR to anticipate and avoid failures [5]. Even when feedback is not available, information about past solutions could potentially increase the consistency of LLM solutions.

This paper describes an initial algorithm for this approach and evaluates an implementation on three collections of problems drawn from the Massive Multitask Language Understanding (MMLU) dataset [6]. For the tests, LLMs are augmented with a module that captures the problem, solution, and feedback as a case to store. For subsequent similar problems, the most relevant prior case is retrieved and provided as context to the LLM. The provided feedback is simple—simply whether a solution is correct or incorrect—to minimize burden on the user. The evaluation shows that even under this constraint, the tested LLMs are often able to perform better with the case-based component.

When an LLM provides erroneous information, rather than simply capturing and storing that case, the feedback can be provided to the LLM immediately, as an opportunity for additional processing. The paper also tests multi-round feedback situations, in which initial feedback of a problem prompts subsequent LLM processing, which itself receives feedback. Experimental results support that in many cases, even a single round of feedback improves performance, but multiple rounds do not provide consistent benefit.

ICCBR Workshop on Memory-Based Reasoning for Responsible GenAI at ICCBR2026, August 15, 2026, Bremen, Germany

0000-0001-9464-025X (L. Yin); 0000-0002-8666-3416 (D. Leake)

© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2026-09-03

CEUR-WS.org/Vol-4255/lake.pdf

2. Related Work

Integrations of CBR with LLMs have prompted much interest (See Bach et al. [1].) Most relevant to this paper is work on bringing CBR-inspired memory capabilities to LLMs [2], on exploiting cases for RAG [7], and on providing cases to LLMs as a knowledge source for LLM reasoning [8].

Multiple approaches for integrating CBR with LLMs have been proposed in the literature. For example, Feng et al. [9] propose starting from a small number of seed cases and gradually building a case base by integrating expert insight, generating synthetic cases with LLMs, and crowdsourcing feedback. Hatalis et al. [10] propose an algorithm for integrating CBR with LLMs for agentic goal solving. Wilkerson and Leake [8] study implementation of CBR with LLMs in a medical triage classification domain, and Regulagedda et al. [11] test an LLM-based CBR implementation in a math domain, with particular focus on the comparative performance of implicit versus explicit case adaptation. Wiratunga et al. [7] explore the use of case-based reasoning to enhance the RAG step of LLMs for legal cases by retrieving context from past cases using LLMs, which is used to augment the query.

Gan et al. [12] present an approach to visual decision-making in driving scenarios, based on retrieving cases that are then adapted by an LLM. Dannenhauer et al. [13] explore how case-based reasoning can be used to modify a user's prompt iteratively in a code generation context by retrieving past examples of code, generating code, and either storing cases when positive feedback is provided from the user, or reweighing the cases in the event of negative feedback. Guo et al. [14] also apply CBR to generate test scripts for software engineering. Some studies have experimentally shown performance improvement with case learning. For example, Guo et al. [15] apply a CBR cycle for automating a data science workflow, with demonstrated improvement over iterations.

An important characteristic of CBR is the ability to learn from explaining failures [5]. For tasks such as code generation, complex feedback on flaws with generated solutions is automatically provided with error reports, and can be exploited by a CBR system. However, such feedback is not necessarily available in other task domains and may require multiple attempts. This paper explores whether simple and limited feedback is sufficient to improve performance over time in situations where multiple rounds of feedback may be infeasible.

3. The PRIM Case Capture and Reuse Algorithm

This section describes our basic algorithm, which we call Prior Response and Interaction Memory (PRIM). Given a problem, a previous case is retrieved based on the semantic similarity between the current problem and the past problem of the case. Each case includes a previous problem, the LLM's previous response, and feedback to the response. If the feedback is negative, the LLM is prompted to add an explanation for why it was incorrect (cf. research on LLM error recovery [16]). The LLM is instructed to use the previous experience and provided feedback to inform its current situation, an implicit directive towards case adaptation [8].

In the single feedback mode, every question-response-feedback chain is stored as a case, indexed by the original question (see Section 5 for implementation details). If the response was incorrect, an explanation message generated by the LLM will be appended to the chain before storage.

In double feedback mode, PRIM executes an extra step to allow an opportunity to provide a second answer if the first is incorrect, using the prompt shown in Table 1. Responses that are correct on the first attempt are stored as a question-response-feedback chain while responses that are correct on the second attempt are stored as a question-response-feedback-response-feedback chain. In double feedback mode, only cases with a correct answer in the first or second response are stored for retrieval, with the original question as key. The process is shown in Algorithm 1.

As an illustration, one of the problems in the MMLU History dataset refers to a text passage from U.S. President Johnson's veto of the Reconstruction Acts, which he describes as tyrannical. The first question asks which of four twentieth-century political positions is most similar to the passage. When the question is presented to the LLM (for this example, gpt-4.1-nano), it incorrectly selects Joseph

Algorithm 1 PRIM

```
Q ← Problems
C ← {}
mode ∈ {single, double}
for all q_i ∈ Q do
    if |C| > 0 then                                  ▷ Build context h around retrieved case
        c ← RetrieveCase(q_i, C)
        h ← c.append(AddPrompt(q_i))
    else
        h ← [AddPrompt(q_i)]
    end if
    h.append(QueryLLM(h))                  ▷ Query LLM and append response to the context
    if IsCorrect(q_i, GetAnswer(h)) then          ▷ Add feedback depending on the response
        h.append(Affirmative())
        IndexCase(C, GetCase(h, success, single))
        continue
    end if
    h.append(RecoveryPrompt(mode))
    h.append(QueryLLM(h))  ▷ Either make 2nd attempt or add explanation, depending on mode
    if mode == double then
        if IsCorrect(q_i, GetAnswer(h)) then
            h.append(Affirmative())
            IndexCase(C, GetCase(h, success, double))
        end if
    else
        IndexCase(C, GetCase(h, fail, single))
    end if
end for
```

Welch's opposition to Senator McCarthy. When prompted to recover from this error, it instead selects Justice Murphy's dissent in Korematsu v. United States, explaining that it missed protection of individual rights as the core position when considering the dissent. This interaction was stored in a new case. This case was later retrieved as relevant to another question on the same passage from President Johnson's veto, which asked which president wrote the passage. Given the retrieved case, the passage was correctly identified as from Johnson, given the stored prior analysis of the passage's stance on authority and oppression, which the LLM connected to Johnson's personal opinions. The question was not correctly answered by the baseline system, which assumed that the passage referred to Jackson's rhetoric on the Bank of the United States. In this instance, providing the case recovery information appeared to encourage the LLM to focus on power over individuals, not the economy, to rule out the Jackson passage.

4. Experimental Questions

To investigate the benefit of capturing cases for RAG reuse, the experiment addresses two questions:

RQ1 How does case capture and reuse as LLM context affect LLM accuracy, in single-round, limited-information feedback situations?

RQ2 How does an additional round of feedback affect LLM accuracy?

	Single, correct	**Single, incorrect**	**Double, correct**
User	Please answer the following multiple choice problem carefully. Think step by step. [QUESTION]	Please answer the following multiple choice problem carefully. Think step by step. [QUESTION]	Please answer the following multiple choice problem carefully. Think step by step. [QUESTION]
LLM	[ANSWER]	[ANSWER]	[ANSWER]
User	Yes, that is correct.	No, that is incorrect. Can you reason why?	No, that is incorrect. Can you reason why and provide a new answer? Think step by step.
LLM		[EXPLANATION]	[ANSWER]
User			Yes, that is correct.

Table 1
Conversation case templates for (1) LLM first response correct, (2) LLM first response incorrect, in single feedback mode, and (3) LLM second response correct, in double feedback mode.

	Subject order (descending)	**# of problems**
history	high_school_european_history high_school_us_history high_school_world_history	687
science	high_school_biology high_school_chemistry high_school_physics	750
math	high_school_mathematics high_school_statistics	548

Table 2
The order of subjects and total number of problems given for each problem set. Problems within each subject are simply ordered by the database default.

5. Experimental Setup

Evaluation Dataset: Both variants of PRIM were evaluated on the MMLU benchmark, a standard dataset for evaluating LLM accuracy on multiple choice questions [6]. The dataset spans 57 tasks, of which we focused on three subsets collecting similar topics: `history` with 687 questions, grouping high school US history, European history, and world history, `science`, with 750 questions, grouping high school biology, chemistry, and physics, and `math`, with 548 questions, grouping high school mathematics and statistics. In our tests, problems are solved by the LLM one subject category at a time. Within each subject, the subtopics for each topic are solved in the order listed in Table 2. For each subtopic, problems are solved in the default order provided by the MMLU benchmark.
LLMs Tested: To cover a mix of open source and proprietary LLMs, six models were tested: gpt-3.5-turbo, gpt-4.1-nano, Llama-3.1-8B-Instruct, Llama-3.1-70B-Instruct, Qwen2.5-32B-Instruct, and phi-4 [17, 18, 19]. All open source models were downloaded from their official HuggingFace repositories.
Indexing and Retrieval: In order to generate embeddings for all of the problems, the HuggingFace implementation of bge-small-en-v1.5 was used with Sentence-BERT and retrieved using the L2-norm distance with FAISS [20, 21, 22].
The Feedback Model: In order to represent a form of limited feedback, the LLM is only informed that its response is either correct or incorrect, with no added information.
Accuracy Criterion: The LLM is only considered correct if it provided the correct answer in the first response, even in the double feedback condition. The second response in the double feedback condition is not considered during evaluation, but is included in the context used when retrieved later as a case.
Evaluation Baseline: The accuracy of PRIM is compared to a baseline of the accuracy of each LLM on the same prompts, without case retrieval.

gpt-3.5-turbo			
	Baseline	Single	Double
history	78.6	80.3	81.2
science	66.9	64.3	***71.9***
math	61.9	62.8	64.1

gpt-4.1-nano			
	Baseline	Single	Double
history	82.1	**85.0**	83.7
science	83.3	85.6	**88.1**
math	83.0	***87.4***	***90.7***

Llama-3.1-8B-Instruct			
	Baseline	Single	Double
history	76.9	***82.1***	81.8
science	67.5	68.4	70.7
math	58.0	60.9	63.9

Llama-3.1-70B-Instruct			
	Baseline	Single	Double
history	89.1	90.5	90.0
science	86.5	86.5	87.1
math	82.3	82.5	82.3

Qwen2.5-32B-Instruct			
	Baseline	Single	Double
history	88.8	90.4	89.4
science	89.6	89.6	89.9
math	89.8	90.7	91.6

phi-4			
	Baseline	Single	Double
history	88.2	88.9	88.7
science	91.6	91.1	91.2
math	78.8	***90.9***	91.1

Table 3
Accuracy results for tested LLMs for single-feedback, double-feedback, and baseline conditions. Bold in the single feedback mode column indicates statistically significant higher accuracy than the baseline, and bold in the double feedback mode column indicates statistically significant higher accuracy than the single feedback mode ($p < 0.05$). Italicized and bold numbers indicate highly significant results ($p < 0.01$).

6. Results

To establish statistical significance for results, McNemar's test was used to test whether the set of answers produced by the single feedback mode was significantly different from the baseline, and if the set of answers produced by the double feedback mode was significantly different from the single feedback mode, by creating a contingency matrix between the problems solved correctly and incorrectly between each configuration. McNemar's test was chosen because, as a paired test, it accounts for problem order.

Table 3 presents accuracy results for each combination of dataset, feedback condition, and LLM, with statistically significant results (p < .05) shown in bold. Italicized and bold numbers indict highly significant results ($p < 0.01$). These results respond to the first experimental question, **RQ1**, supporting the benefit of capturing and reusing cases as LLM context for certain models and certain problem sets. Figure 1 visualizes results for one of the single-feedback combinations, single feedback with gpt-4.1-nano on the history problem set, illustrating the benefit of ongoing learning. Performance mostly mirrors the baseline for the first few questions before diverging and exceeding the baseline as more relevant cases are stored and retrieved over time.

In response to **RQ2**, in four trials there is a significant difference between the single feedback mode and the baseline, but there are only three trials in which the double feedback mode is significantly better than the single feedback mode. In addition, there are six trials in which the accuracy actually decreases when using the double feedback mode over the single feedback mode. These drops are not statistically significant, so more study is needed. However, the results could suggest that there are only marginal benefits to using multiple rounds of feedback with PRIM and in some cases, can actually be harmful to performance. It is possible that this might be because the second chance to answer provides too much extraneous information, resulting in the case misleading the LLM when it is retrieved. Alternatively, it is possible that the focus on storing only successful cases in the double feedback mode is excluding certain negative cases that are useful as retrieved cases.

On the history problem set with gpt-4.1-nano as the LLM, the double feedback mode underperformed the single feedback mode. Manual analysis shows that out of a total of 687 questions, there were

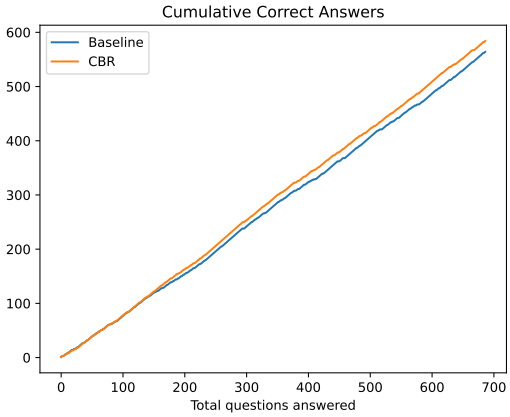

Figure 1: Cumulative number of correct answers for the baseline and single feedback CBR with gpt-4.1-nano on the history problem set.

23 questions that the baseline missed but the CBR system with double feedback answered correctly. Given that in aggregate the double feedback mode only answered 11 more questions correctly than the baseline, this shows that the double feedback mode may cause the LLM to make mistakes not present in the baseline.

7. Limitations

This study provides a simple first demonstration. One limitation is that the system uses a simple retrieval approach, which may limit its ability to retrieve the most relevant cases. The evaluation itself focuses on a few domains, for answering multiple-choice questions, and uses a limited range of LLMs; tests with additional datasets—and datasets with different characteristics and requiring richer answers—and additional LLMs would give a stronger assessment of generalizability.

8. Conclusions and Future Work

This paper proposes augmenting LLMs with a memory component to capture query-response cases with feedback, furnishing a growing case base of contextual information for RAG. It presents the PRIM algorithm for this process, and an initial evaluation on test datasets drawn from the MMLU benchmark. The evaluation suggests that PRIM can improve performance over time even with simple feedback, generating a useful case base of contextual information for RAG, without requiring an external curated dataset. The evaluation also tested providing the feedback immediately to the LLM for a second round of processing, with the unexpected result that multiple rounds of feedback may not improve performance. This is an area for future study.

Broadening the scope to additional tasks and more open-ended responses would be a natural next step, as would testing for agentic tasks with richer potential failure feedback. Additional work is required to understand the best methods for retrieval and indexing LLM success and failure cases. The approach also relies on LLM reasoning about its own failure, so future research could explore different methods for eliciting the most informative explanations, or how to integrate relevant external information into the explanation generation process.

Interestingly, although most of the LLMs tested only show significant improvement with CBR systems on certain problem sets, gpt-4.1-nano shows significant improvements across all domains. Reasons for this are unclear, but it is possible that LLMs could be trained or tuned specifically to reason about mistakes or better use provided feedback. This is another possible avenue for future work. Another broad question is comparison of this case-based approach to fine-tuning the LLM using captured answers

[23]. CBR contrasts in providing inertia-free learning; it would also be interesting to compare accuracy, computational costs, and data requirements.

The experiments in this paper focus solely on accuracy benefits. However, we hypothesize that case capture for RAG reuse can increase response consistency, supporting user trust, even in the absence of feedback. In open-ended tasks with a range of solutions, case capture could also provide a form of personalization, as the responses favored by a user in the past influence those generated by the LLM in the future. Ramifications for consistency and personalization are an interesting area for future research, as is the effects of both on user trust and satisfaction with LLM responses.

Finally, we note the opportunity for an approach in which captured cases for sufficiently similar queries are used to directly solve queries to the LLM when possible, replacing LLM processing with CBR, thus bypassing expensive LLM processing. In that approach, would CBR serve as a *case-based intelligent component*, wrapped around another system and learning from its behavior, as proposed by Riesbeck [24]. This is an interesting potential extension.

Acknowledgments

This research was supported in part by Lilly Endowment, Inc., through its support for the Indiana University Pervasive Technology Institute.

Declaration on Generative AI

The authors have not employed any Generative AI tools for the writing of this paper.

References

[1] K. Bach, R. Bergmann, F. Brand, M. Caro-Martínez, V. Eisenstadt, M. W. Floyd, L. Jayawardena, D. Leake, M. Lenz, L. Malburg, D. H. Ménager, M. Minor, B. Schack, I. Watson, K. Wilkerson, N. Wiratunga, Case-Based Reasoning Meets Large Language Models: A Research Manifesto For Open Challenges and Research Directions, 2025. URL: https://hal.science/hal-05006761.

[2] M. W. Floyd, D. Leake, D. H. Ménager, I. D. Watson, K. Wilkerson, Levels of AI memory - and case-based ways for LLMs to ascend them, in: ICCBR Workshops, 2025, pp. 2–14. URL: https://ceur-ws.org/Vol-3993/paper1.pdf.

[3] W. Fan, Y. Ding, L. Ning, S. Wang, H. Li, D. Yin, T.-S. Chua, Q. Li, A survey on RAG meeting LLMs: Towards retrieval-augmented large language models, in: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '24, Association for Computing Machinery, New York, 2024, p. 6491–6501.

[4] T. Ueno, Y. Sawa, Y. Kim, J. Urakami, H. Oura, K. Seaborn, Trust in human-AI interaction: Scoping out models, measures, and methods, in: CHI Conference on Human Factors in Computing Systems Extended Abstracts, ACM, 2022, p. 1–7.

[5] K. Hammond, Case-Based Planning: Viewing Planning as a Memory Task, Academic Press, San Diego, 1989.

[6] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, J. Steinhardt, Measuring massive multitask language understanding, in: Proceedings of the International Conference on Learning Representations (ICLR), OpenReview.net, 2021.

[7] N. Wiratunga, R. Abeyratne, L. Jayawardena, K. Martin, S. Massie, I. Nkisi-Orji, R. Weerasinghe, A. Liret, B. Fleisch, CBR-RAG: Case-based reasoning for retrieval augmented generation in llms for legal question answering, in: Case-Based Reasoning Research and Development, Springer, Cham, 2024, pp. 445–460.

[8] K. Wilkerson, D. Leake, On implementing case-based reasoning with large language models, in: Case-Based Reasoning Research and Development, ICCBR 2024, Springer, Cham, 2024, pp. 404–417.

[9] K. J. K. Feng, Q. Z. Chen, I. Cheong, K. Xia, A. X. Zhang, Case repositories: Towards case-based reasoning for AI alignment, 2023. URL: https://arxiv.org/abs/2311.10934. arXiv:2311.10934.

[10] K. Hatalis, D. Christou, V. Kondapalli, Review of case-based reasoning for LLM agents: Theoretical foundations, architectural components, and cognitive integration, 2025. URL: https://arxiv.org/abs/2504.06943. arXiv:2504.06943.

[11] R. Regulagedda, N. Krupashankar, D. Leake, Keep adaptation simple: Implicit vs. explicit adaptation for LLM-based CBR, in: Case-Based Reasoning Research and Development, ICCBR 2026, Springer, 2026. In press.

[12] W. Gan, M.-S. Dao, K. Zettsu, Case-based reasoning augmented large language model framework for decision making in realistic safety-critical driving scenarios, Safety Science 201 (2026) 107234.

[13] D. Dannenhauer, Z. Dannenhauer, D. Christou, K. Hatalis, A case-based reasoning approach to dynamic few-shot prompting for code generation, in: ICML 2024 Workshop on LLMs and Cognition, 2024. URL: https://openreview.net/forum?id=Kt9bM32oDY.

[14] S. Guo, H. Liu, X. Chen, Y. Xie, L. Zhang, T. Han, H. Chen, Y. Chang, J. Wang, Optimizing case-based reasoning system for functional test script generation with large language models, in: Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2, KDD '25, Association for Computing Machinery, New York, NY, USA, 2025, p. 4487–4498. URL: https://doi.org/10.1145/3711896.3737254. doi:10.1145/3711896.3737254.

[15] S. Guo, C. Deng, Y. Wen, H. Chen, Y. Chang, J. Wang, DS-agent: automated data science by empowering large language models with case-based reasoning, in: Proceedings of the 41st International Conference on Machine Learning, ICML'24, JMLR.org, 2024.

[16] S. Tan, K. Lin, K. Sen, M. Zaharia, Recovery-bench: Evaluating agentic recovery from mistakes, in: NeurIPS 2025 Workshop on Evaluating the Evolving LLM Lifecycle: Benchmarks, Emergent Abilities, and Scaling, 2025. URL: https://openreview.net/forum?id=8FZRnDgDxq.

[17] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, A. Yang, A. Fan, A. Goyal, A. Hartshorn, A. Yang, A. Mitra, A. Sravankumar, A. Korenev, A. Hinsvark, et al., The Llama 3 herd of models, 2024. URL: https://arxiv.org/abs/2407.21783. arXiv:2407.21783.

[18] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, H. Lin, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Lin, K. Dang, K. Lu, K. Bao, K. Yang, L. Yu, M. Li, M. Xue, P. Zhang, Q. Zhu, R. Men, R. Lin, T. Li, T. Tang, T. Xia, X. Ren, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Wan, Y. Liu, Z. Cui, Z. Zhang, Z. Qiu, Qwen2.5 technical report, 2025. URL: https://arxiv.org/abs/2412.15115. arXiv:2412.15115.

[19] M. Abdin, J. Aneja, H. Behl, S. Bubeck, R. Eldan, S. Gunasekar, M. Harrison, R. J. Hewett, M. Javaheripi, P. Kauffmann, J. R. Lee, Y. T. Lee, Y. Li, W. Liu, C. C. T. Mendes, A. Nguyen, E. Price, G. de Rosa, O. Saarikivi, A. Salim, S. Shah, X. Wang, R. Ward, Y. Wu, D. Yu, C. Zhang, Y. Zhang, Phi-4 technical report, 2024. URL: https://arxiv.org/abs/2412.08905. arXiv:2412.08905.

[20] S. Xiao, Z. Liu, P. Zhang, N. Muennighoff, C-pack: Packaged resources to advance general Chinese embedding, 2023. arXiv:2309.07597.

[21] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using siamese BERT-networks, 2019. URL: https://arxiv.org/abs/1908.10084. arXiv:1908.10084.

[22] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, H. Jégou, The Faiss library, 2025. URL: https://arxiv.org/abs/2401.08281. arXiv:2401.08281.

[23] Y. Zhang, S. Sun, M. Galley, Y. Chen, C. Brockett, X. Gao, J. Gao, J. Liu, B. Dolan, DialoGPT: Large-scale generative pre-training for conversational response generation, 2020. URL: https://arxiv.org/abs/1911.00536. arXiv:1911.00536.

[24] C. Riesbeck, What next? The future of CBR in postmodern AI, in: D. Leake (Ed.), Case-Based Reasoning: Experiences, Lessons, and Future Directions, AAAI Press, Menlo Park, CA, 1996, pp. 371–388.