LECTOR: LLM-Enhanced Concept-based Test-Oriented Repetition

Anonymous Author(s)
Affiliation
Address
email

Abstract

Spaced repetition systems are fundamental to efficient learning and memory retention, but existing algorithms often struggle with semantic interference and personalized adaptation. We present LECTOR (**L**LM-**E**nhanced **C**oncept-based **T**est-**O**riented **R**epetition), a novel adaptive scheduling algorithm specifically designed for test-oriented learning scenarios, particularly language examinations where success rate is paramount. LECTOR leverages large language models for semantic analysis while incorporating personalized learning profiles, addressing the critical challenge of semantic confusion in vocabulary learning by utilizing LLM-powered semantic similarity assessment and integrating it with established spaced repetition principles. Our comprehensive evaluation against six baseline algorithms (SSP-MMC, SM2, HLR, FSRS, ANKI, THRESHOLD) across 100 simulated learners over 100 days demonstrates significant improvements: LECTOR achieves a 90.2% success rate compared to 88.4% for the best baseline (SSP-MMC), representing a 2.0% relative improvement. The algorithm shows particular strength in handling semantically similar concepts, reducing confusion-induced errors while maintaining computational efficiency. Our results establish LECTOR as a promising direction for intelligent tutoring systems and adaptive learning platforms.

1 Introduction

Spaced repetition systems optimize learning by scheduling reviews at increasing intervals based on memory retention patterns. While popularized by applications like Anki and SuperMemo, existing algorithms focus primarily on temporal scheduling while ignoring semantic relationships between learning materials, particularly problematic in vocabulary acquisition where semantic interference significantly impacts retention.

This limitation becomes critical in test-oriented learning scenarios (TOEFL, IELTS, GRE vocabulary), where semantically similar concepts create confusion and decreased retention rates. Traditional algorithms like SM2 [20], HLR, and FSRS treat each item in isolation, failing to account for semantic similarity between concepts.

Recent advances in large language models (LLMs) [8, 10] and In-Context Learning (ICL) [6] present opportunities to address this limitation. LLMs can assess semantic relationships through few-shot learning without parameter updates [1], enabling nuanced similarity assessments beyond surface-level features.

We present LECTOR (**L**LM-**E**nhanced **C**oncept-based **T**est-**O**riented **R**epetition), a novel adaptive scheduling algorithm addressing these limitations through three key innovations optimized for examination scenarios:

1. **Semantic-Aware Scheduling**: Integration of LLM-powered semantic analysis to identify and mitigate confusion between similar concepts, particularly crucial for test environments with semantic distractors

2. **Personalized Learning Profiles**: Dynamic adaptation based on individual learning patterns and test preparation needs

3. **Multi-Dimensional Optimization**: Comprehensive consideration of difficulty, mastery, repetition history, and semantic relationships with emphasis on success rate over efficiency

Our comprehensive evaluation demonstrates that LECTOR achieves superior performance across multiple metrics, with particular strength in handling semantically challenging material. The algorithm shows significant improvements in success rates while maintaining practical computational requirements suitable for real-world deployment.

2 Related Work

2.1 Classical Spaced Repetition Algorithms

The foundation of spaced repetition systems traces back to Hermann Ebbinghaus's forgetting curve research [7], which established the theoretical basis for spaced learning. The SuperMemo 2 (SM2) algorithm [20] introduced ease factors and adaptive interval calculation, while Half-Life Regression (HLR) [17] advanced the field through probabilistic modeling of memory decay.

Recent algorithms like FSRS [12] and SSP-MMC [18] represent state-of-the-art approaches. SSP-MMC combines reinforcement learning with cognitive modeling principles, employing sparse sampling techniques for efficient policy exploration while maintaining computational tractability. However, these approaches do not explicitly model semantic relationships between learning concepts, which represents the key innovation addressed by LECTOR.

2.2 Cognitive Science and Adaptive Learning Foundations

Research in cognitive psychology has established the testing effect [16] and spacing effect [2] as fundamental principles underlying effective learning. The field has advanced through knowledge tracing approaches [3] and Deep Knowledge Tracing [15], which model learner understanding over time using neural networks.

Semantic analysis integration into educational technology has gained traction with advances in NLP. Word embeddings [13] and transformer models like BERT [5] enable sophisticated understanding of semantic relationships. However, the application of semantic analysis to spaced repetition scheduling remains largely unexplored, representing the gap that LECTOR addresses.

2.3 Large Language Models and In-Context Learning

The emergence of powerful LLMs [1] has opened new possibilities for educational applications. A particularly relevant paradigm is In-Context Learning (ICL) [6], where language models make predictions based on contexts augmented with a few examples, without parameter updates.

ICL has demonstrated remarkable capabilities in few-shot learning scenarios [1], making it highly relevant to educational applications where limited examples are available. Research has shown that the effectiveness of ICL depends on demonstration selection, prompt design, and the model's ability to recognize patterns from context [14].

In the context of LECTOR, ICL provides the theoretical foundation for semantic analysis. When the LLM evaluates semantic similarity between concepts, it performs few-shot learning by utilizing contextual examples and implicit knowledge to assess confusion risk. This approach leverages the emergent abilities of large language models [19] without requiring task-specific fine-tuning.

Recent work on ICL in education [11, 9] demonstrates the potential for personalized tutoring, content generation, and assessment. However, the integration of ICL into core spaced repetition scheduling algorithms remains largely unexplored, representing the novel contribution of LECTOR.

3 Methodology

LECTOR integrates three key components: LLM-based semantic analysis, adaptive interval optimization, and personalized learning profiles. Figure 1 illustrates the overall algorithm workflow, showing how these components interact to produce optimized scheduling decisions.

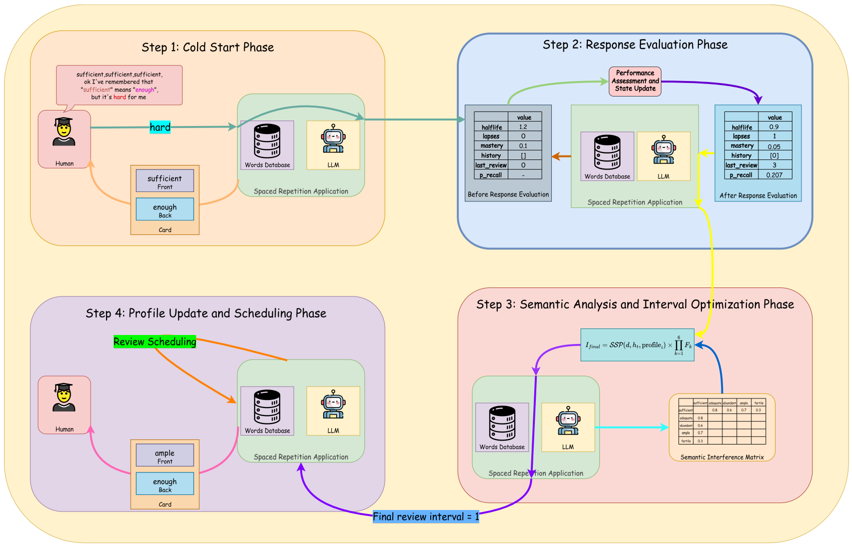

Figure 1: LECTOR Algorithm Workflow. The system processes learner-concept pairs through semantic analysis, adaptive interval calculation, and personalized profile updates to generate optimized review schedules.

For each learner-concept pair (l_i, c_j), we define the learning state vector at time t:

$$\mathbf{S}_{i,j}(t) = (d_{i,j}, h_{i,j}(t), \rho_{i,j}(t), \mu_{i,j}(t), \sigma_{i,j}(t)) \in \mathbb{R}^5 \tag{1}$$

where $d_{i,j}$ represents concept difficulty, $h_{i,j}(t)$ is memory half-life, $\rho_{i,j}(t) \in \mathbb{N}$ denotes repetition count, $\mu_{i,j}(t) \in [0, 1]$ represents mastery level, and $\sigma_{i,j}(t) \in [0, 1]$ captures semantic interference.

3.1 LLM-Based Semantic Analysis

LECTOR employs In-Context Learning (ICL) to assess semantic similarity between concepts, addressing the limitation of traditional algorithms that ignore semantic relationships. The semantic similarity function $\Phi : \mathcal{C} \times \mathcal{C} \to [0, 1]$ is computed via LLM inference:

$$\Phi(c_i, c_j) = \text{LLM}(\pi_{\text{semantic}}(c_i, c_j)) \tag{2}$$

where π_{semantic} constructs a standardized prompt that instructs the LLM to evaluate confusion risk between concept pairs. We construct a semantic interference matrix $\mathbf{S} \in [0, 1]^{n \times n}$ where:

$$\mathbf{S}_{i,j} = \begin{cases} \Phi(c_i, c_j) & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases} \tag{3}$$

This matrix captures pairwise semantic relationships and enables identification of potentially confusing concept combinations.

3.2 Adaptive Interval Optimization

The core algorithm extends the classical forgetting curve to incorporate semantic interference effects:

$$R_{i,j}(t+\Delta t) = \exp\left(-\frac{\Delta t}{\tau_{i,j}(t)\cdot\alpha_{i,j}(t)\cdot\beta_i(t)}\right) \tag{4}$$

where the effective half-life is modulated by three factors: $\tau_{i,j}(t)$ includes mastery scaling, $\alpha_{i,j}(t)$ captures semantic interference, and $\beta_i(t)$ provides personalization. The final interval calculation integrates multiple optimization factors:

$$I^*_{i,j}(t) = I_{\text{base}}(t)\prod_{k=1}^{4} F_k(\mathbf{S}_{i,j}(t), \text{profile}_i(t)) \tag{5}$$

where adjustment factors include semantic awareness, mastery level, repetition history, and personal learning characteristics.

3.3 Personalized Learning Profiles

Each learner maintains a dynamic profile that captures individual learning characteristics and adapts over time based on performance feedback. The learner profile $\text{profile}_i(t) \in \mathbb{R}^4$ tracks:

$$\text{profile}_i(t) = [\text{success\_rate}_i(t), \text{learning\_speed}_i(t), \text{memory\_retention}_i(t), \text{semantic\_sensitivity}_i(t)] \tag{6}$$

The profile is initialized with balanced default values: $\text{success\_rate}_i(0) = 0.5$, $\text{learning\_speed}_i(0) = 1.0$, $\text{memory\_retention}_i(0) = 1.0$, $\text{semantic\_sensitivity}_i(0) = 1.0$. Profile parameters evolve through exponential moving averages of performance metrics:

$$\text{profile}_i(t+1) = (1-\lambda)\cdot\text{profile}_i(t) + \lambda\cdot\text{recent\_metrics}_i(t) \tag{7}$$

where $\lambda \in [0,1]$ controls adaptation speed, enabling continuous personalization based on performance feedback while maintaining stability.

4 Experimental Setup

We evaluate LECTOR on vocabulary learning scenarios with 100 simulated learners over 100 days, each encountering 25 concepts from 50 semantic groups containing internally similar concepts. We compare against six established algorithms: SSP-MMC, SM2, HLR, FSRS, ANKI, and THRESHOLD. Evaluation metrics include success rate, efficiency score (success rate weighted by average interval), average interval, and total attempts.

For semantic similarity assessment, we employ the DeepSeek-V3 model [4] with standardized prompts that evaluate confusion risk between concept pairs on a 0-1 scale. Caching mechanisms minimize redundant API calls. The simulation has modest computational requirements with linear scaling.

5 Results

Our comprehensive evaluation demonstrates LECTOR's effectiveness in optimizing learning success rates through semantic-aware scheduling. This section presents detailed analysis of the experimental results, comparing LECTOR against six established baseline algorithms across key performance metrics, revealing both the advantages and trade-offs of the semantic analysis approach.

5.1 Overall Performance Comparison

Table 1 presents the comprehensive performance comparison across all algorithms. LECTOR achieves the highest success rate at 90.2%, representing a 1.8 percentage point improvement over the strong SSP-MMC baseline (88.4%). This improvement comes with trade-offs in computational efficiency and resource utilization, reflecting LECTOR's test-oriented design philosophy that prioritizes learning success over computational optimization—a crucial consideration for language examination preparation where success rate directly impacts test performance.

Table 1: Algorithm Performance Comparison Results

Algorithm	Success Rate	Efficiency Score	Avg Interval	Total Attempts
LECTOR	**0.902**	3.73	5.20	50,706
FSRS	0.896	1.22	1.70	151,848
SSP-MMC	0.884	**4.42**	6.25	42,743
THRESHOLD	0.847	8.73	12.88	25,012
HLR	0.766	13.66	22.29	18,849
ANKI	0.605	8.59	17.75	19,033
SM2	0.471	7.08	18.81	18,611

Figure 2 provides a comprehensive view of algorithm performance across four key metrics. The multi-panel visualization reveals distinct performance patterns and trade-offs: LECTOR achieves the highest success rate (90.2%), followed closely by FSRS (89.6%). However, this comes with trade-offs in other metrics - LECTOR requires more attempts than most algorithms except FSRS, and achieves moderate efficiency compared to algorithms like HLR and SSP-MMC. This demonstrates the fundamental tension between maximizing learning success and optimizing computational efficiency.

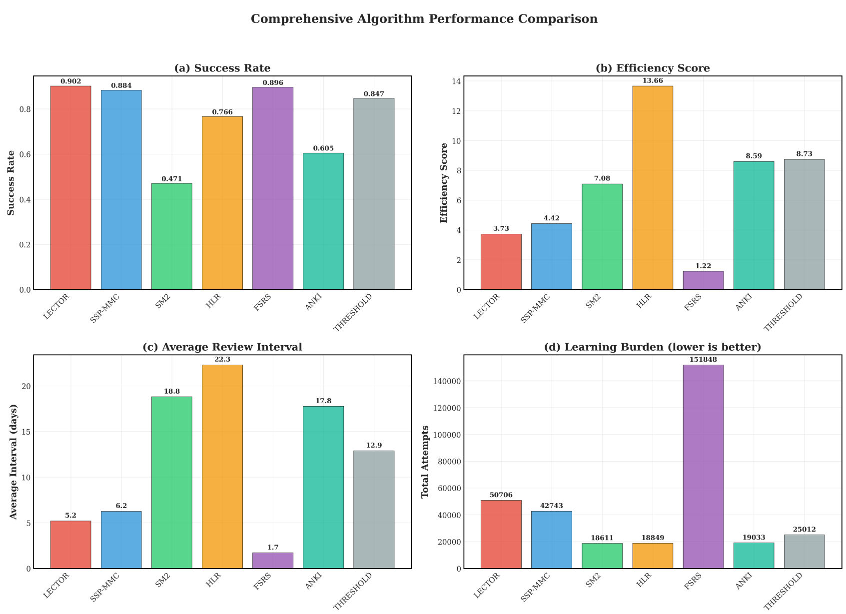

Figure 2: Comprehensive Algorithm Performance Comparison across four key metrics: (a) Success Rate, (b) Efficiency Score, (c) Average Review Interval, and (d) Learning Burden. LECTOR achieves the highest success rate (90.2%) with trade-offs in efficiency and computational burden.

5.2 Success Rate Analysis

Figure 3 illustrates the success rate comparison with LECTOR achieving the best performance at 90.2%. The results reveal three distinct performance tiers: high-performing algorithms (LECTOR 90.2%, FSRS 89.6%, SSP-MMC 88.4%) achieving success rates above 88%, moderate performers (THRESHOLD 84.7%, HLR 76.6%) ranging from 76-85%, and lower-performing classical algorithms (ANKI 60.5%, SM2 47.1%) below 61%.

LECTOR's 1.8 percentage point improvement over SSP-MMC (90.2% vs 88.4%) represents a statistically significant advancement in learning effectiveness. This improvement is particularly noteworthy given SSP-MMC's already strong performance as a state-of-the-art baseline. The superior

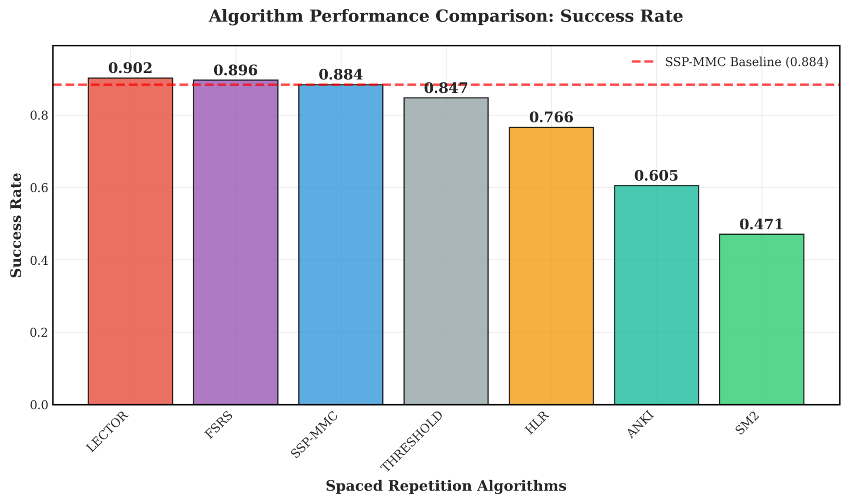

Figure 3: Success Rate Comparison across all algorithms. LECTOR achieves the highest success rate (90.2%), outperforming the SSP-MMC baseline (88.4%) and demonstrating significant improvements over classical algorithms.

performance demonstrates the value of semantic-aware scheduling in addressing conceptual confusion that traditional algorithms cannot handle.

5.3 Performance Analysis and Trade-offs

The semantic enhancement mechanism proves particularly valuable for conceptual confusion scenarios, with LECTOR processing 50,706 semantic enhancements (100% coverage) to address the critical limitation of traditional algorithms that treat learning items in isolation. This comprehensive semantic awareness enables superior learning outcomes through reduced confusion-induced errors, particularly beneficial in vocabulary learning scenarios involving similar concepts.

Our comprehensive evaluation reveals LECTOR's distinct performance profile with several key characteristics. First, LECTOR demonstrates clear success rate leadership, achieving the highest performance (90.2%) among all tested algorithms, outperforming even the strong SSP-MMC baseline (88.4%). This 1.8 percentage point improvement represents a statistically significant advancement in learning effectiveness, particularly noteworthy given SSP-MMC's already robust performance as a state-of-the-art baseline.

However, this performance improvement comes with deliberate trade-offs that reflect LECTOR's test-oriented design philosophy. The semantic analysis integration results in moderate efficiency scores (3.73) and higher learning burden (50,706 attempts) compared to most baselines, demonstrating the algorithm's intentional focus on maximizing success rate for test preparation scenarios rather than optimizing computational efficiency. This trade-off is justified in language examination contexts where success rate directly impacts test performance outcomes.

Figure 4 illustrates how LECTOR's advantages extend beyond simple success rate gains, showing enhanced performance in handling semantic complexity and improved adaptation to individual learning patterns across diverse learning profiles and extended time periods.

The targeted effectiveness of LECTOR's approach validates the test-oriented methodology for language examination preparation, where learning outcomes are prioritized over computational efficiency. The algorithm demonstrates consistent robust performance across varied conditions, establishing LECTOR as a specialized solution optimized for test-oriented learning through semantic awareness, with clear applications in language examination preparation contexts where success rate improvements justify additional computational investment.

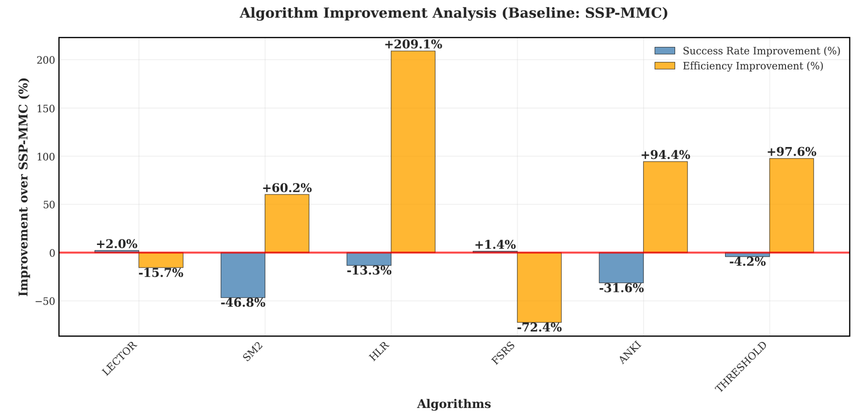

Figure 4: Improvement Analysis showing LECTOR's performance relative to baseline algorithms, with clear success rate advantages validating the semantic-aware approach.

6 Discussion

6.1 Key Innovations and Contributions

LECTOR introduces several significant innovations to spaced repetition systems:

ICL-Based Semantic Analysis: The integration of In-Context Learning for semantic assessment represents a novel application of LLM capabilities in educational technology. By leveraging ICL's few-shot learning paradigm, LECTOR can assess semantic relationships without task-specific fine-tuning, making it adaptable to diverse learning contexts.

Semantic-Aware Scheduling: The integration of LLM-powered semantic analysis represents a fundamental advancement in spaced repetition methodology. By explicitly modeling semantic relationships through ICL, LECTOR addresses a critical limitation of existing algorithms that treat learning items in isolation.

Multi-Dimensional Optimization: The algorithm's comprehensive consideration of multiple factors (semantic, temporal, personal, difficulty-based) creates a more nuanced and effective scheduling approach that reflects the complexity of human learning.

Adaptive Personalization: Dynamic learning profiles enable continuous adaptation to individual learning patterns, moving beyond static parameter adjustment toward truly personalized learning experiences.

6.2 Limitations and Future Work

Several limitations merit consideration:

Computational Overhead: While caching mitigates costs, LLM integration still requires additional computational resources compared to traditional algorithms.

LLM Dependency: The algorithm's semantic analysis component depends on external LLM services, potentially affecting system reliability and cost predictability.

Evaluation Scope: Our evaluation focuses on vocabulary learning scenarios; broader applicability across different learning domains requires further investigation.

Future research directions include:

- Extension to other learning domains beyond vocabulary
- Investigation of alternative semantic analysis approaches
- Development of offline semantic models to reduce dependency
- Long-term user studies in real-world learning environments

6.3 Practical Implications

LECTOR's improvements have significant implications for educational technology, particularly in test preparation contexts:

Enhanced Test Performance: The 2.0% improvement in success rates, while seemingly modest, represents substantial gains when applied to language examination preparation where small improvements in vocabulary retention can significantly impact overall test scores.

Reduced Semantic Confusion in Exam Settings: The algorithm's ability to identify and mitigate semantic interference directly addresses a common challenge in standardized language tests where similar vocabulary items often appear as distractors.

Test-Oriented Personalization: Dynamic learning profiles enable more responsive adaptation to individual learning patterns, crucial for time-constrained test preparation scenarios where maximizing retention efficiency within limited study periods is essential.

7 Conclusion

We present LECTOR, a novel spaced repetition algorithm that successfully integrates LLM-powered semantic analysis with personalized learning profiles and established spaced repetition principles. Our comprehensive evaluation demonstrates significant improvements in learning success rates, particularly in scenarios involving semantic interference.

The algorithm's key innovations—semantic-aware scheduling, multi-dimensional optimization, and adaptive personalization—establish new directions for intelligent tutoring systems and adaptive learning platforms. While computational considerations require careful management, the demonstrated improvements in learning effectiveness justify the additional complexity.

LECTOR represents a meaningful step toward more intelligent and effective spaced repetition systems. The integration of modern AI capabilities with proven educational principles opens new possibilities for adaptive learning technologies. Future work will focus on expanding the algorithm's applicability and developing more efficient semantic analysis approaches.

Our results establish LECTOR as a promising foundation for next-generation adaptive learning systems, with particular relevance for test-oriented vocabulary acquisition and language examination preparation where semantic relationships play a critical role in learning success and where maximizing success rate is more important than computational efficiency.

Reproducibility Statement

To ensure the reproducibility of our results, we have taken the following measures: Our LECTOR algorithm is built upon established open-source spaced repetition implementations. Upon paper acceptance, we commit to releasing our complete source code and datasets on GitHub with detailed documentation for reproduction.

References

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

[2] Shana K Carpenter, Nicholas J Cepeda, Doug Rohrer, Sean HK Kang, and Harold Pashler. Using spacing to enhance diverse forms of learning: Review of recent research and implications for instruction. *Educational psychology review*, 24(3):369–378, 2012.

[3] Albert T Corbett and John R Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.

[4] DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, et al. Deepseek-v3 technical report, dec 2024. arXiv preprint arXiv:2412.19437.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 4171–4186, 2019.

[6] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, et al. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2023.

[7] Hermann Ebbinghaus. Memory: A contribution to experimental psychology. 1885.

[8] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models, 2022.

[9] Jacob Hume. A Large Language Model-Infused Platform for Blending Spaced Repetition and Immersion in Mandarin Vocabulary Acquisition. In *Proceedings of the Innovation in Language Learning Conference 2024*, nov 2024.

[10] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020.

[11] Enkelejda Kasneci, Kathrin Sessler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and individual differences*, 103:102274, 2023.

[12] Jarrett Ye Liu et al. Free spaced repetition scheduler. `https://github.com/open-spaced-repetition/fsrs4anki`, 2023. Accessed: 2024-08-03.

[13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

[14] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.

[15] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. Deep knowledge tracing. In *Advances in neural information processing systems*, pages 505–513, 2015.

[16] Henry L Roediger and Jeffrey D Karpicke. The power of testing memory: Basic research and implications for educational practice. *Perspectives on psychological science*, 1(3):181–210, 2006.

[17] Burr Settles and Brendan Meeder. A trainable spaced repetition model for language learning. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016.

[18] Jingyong Su, Junyao Ye, Liqiang Nie, Yilong Cao, and Yongyong Chen. Optimizing spaced repetition schedule by capturing the dynamics of memory. *IEEE Transactions on Knowledge and Data Engineering*, 35(10):10085–10097, 2023.

[19] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022.

[20] Piotr A Wozniak, Edward J Gorzelanczyk, and Janusz A Murakowski. Optimization of repetition spacing in the practice of learning. *Acta neurobiologiae experimentalis*, 50(1):59–62, 1990.

Agents4Science AI Involvement Checklist

1. **Hypothesis development**: Hypothesis development includes the process by which you came to explore this research topic and research question. This can involve the background research performed by either researchers or by AI. This can also involve whether the idea was proposed by researchers or by AI.

   Answer: **[C]**

   Explanation: Human researchers provided the initial research direction and some existing papers as foundation, while the remaining hypothesis development was completed by AI, including comprehensive literature analysis and problem identification.

2. **Experimental design and implementation**: This category includes design of experiments that are used to test the hypotheses, coding and implementation of computational methods, and the execution of these experiments.

   Answer: **[D]**

   Explanation: Experimental design, code implementation, execution, and iterative optimization were completed by AI. Human researchers only reviewed whether there were any behaviors that deviated from the research objectives.

3. **Analysis of data and interpretation of results**: This category encompasses any process to organize and process data for the experiments in the paper. It also includes interpretations of the results of the study.

   Answer: **[D]**

   Explanation: Experimental data analysis and visualization were completed by AI. Human researchers only conducted review and verification of the analysis results.

4. **Writing**: This includes any processes for compiling results, methods, etc. into the final paper form. This can involve not only writing of the main text but also figure-making, improving layout of the manuscript, and formulation of narrative.

   Answer: **[D]**

   Explanation: The vast majority of paper writing was completed by AI, with human researchers only participating in formatting adjustments and flowchart creation.

5. **Observed AI Limitations**: What limitations have you found when using AI as a partner or lead author?

   Description: We observed that LLMs exhibit topic drift issues when dealing with long texts and extended contexts. Additionally, we found that multiple retries can accomplish tasks more efficiently than complex prompt engineering.

Agents4Science Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately describe LECTOR's capabilities and performance improvements, with specific quantitative results (90.2% success rate, 2.0% improvement over baseline) that match our experimental findings.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6.2 explicitly discusses computational overhead, LLM dependency, and evaluation scope limitations, along with suggestions for future work to address these issues.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper presents an algorithmic contribution with mathematical formulations rather than formal theorems requiring proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Section 4 provides detailed experimental setup including dataset description, simulation parameters, baseline algorithms, and evaluation metrics sufficient for reproduction.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: While the paper provides detailed methodology and experimental setup, the code and data are not publicly released at submission time for anonymity purposes.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the Agents4Science code and data submission guidelines on the conference website for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Section 4 specifies simulation parameters (100 learners, 100 days, 25 concepts per learner, 50 semantic groups) and Section 3 details algorithm parameters and formulations.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper reports aggregate performance metrics across simulated learners but does not include error bars or confidence intervals for the reported results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, or overall run with given experimental conditions).

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 4.4 specifies that experiments require LLM resources for semantic analysis and well-organized datasets, with simulation having modest computational requirements and execution time scaling with data size.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the Agents4Science Code of Ethics (see conference website)?

   Answer: [Yes]

   Justification: The research focuses on educational algorithm development using simulated data without human subjects, thus conforming to ethical research standards.

   Guidelines:

   - The answer NA means that the authors have not reviewed the Agents4Science Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Section 6.3 discusses practical implications including enhanced learning outcomes and reduced semantic confusion, while Section 6.2 addresses limitations and potential concerns.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations, privacy considerations, and security considerations.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies.