Milestone 3: Research Proposal
Backdoors in RAG-Based LLM Systems

Michael Hudson

July 2025

1 Introduction & Motivation

Retrieval-Augmented Generation (RAG) integrates a large language model with an external retrieval module, enabling it to incorporate relevant information from a large corpus of passages during generation. A typical RAG pipeline consists of three components: a knowledge base of passages; a retriever that embeds and ranks passages by semantic similarity to the input query; and a large language model that conditions its output on both the original query and the retrieved context.

This modular design brings practical benefits. Since the knowledge base can be updated independently of the LLM, RAG enables grounded responses with current or domain-specific context without retraining. It has gained traction across industries, including health [1], finance [2] and legal [3], and is often cheaper and faster to deploy than fine-tuning an LLM. However, the same modularity introduces new vulnerabilities. Since the LLM often treats retrieved context as trustworthy, even minor manipulations of the corpus can yield significant behavioural changes. Moreover, corpus poisoning does not require LLM access, only the ability to insert or modify passages, thereby significantly lowering the barrier to entry.

2 Literature Review

Retrieval-Augmented Generation (RAG) pipelines have emerged as a prime target for adversarial attacks, largely due to their modular design and reliance on external corpora. Recent work has sought to exploit this vulnerability through various attack vectors. Zou et al. (2024) [4] propose *PoisonedRAG*, which demonstrates that a single, adversarially crafted passage — generated using HotFlip-style gradient-based learning — can be inserted into various corpora to reliably hijack retrieval for a fixed query. However, the attack's impact is limited by its specificity to particular queries and the detectability of the resulting poisoned documents, which often exhibit unnatural phrasing.

Xue et al. (2024) [5] put forward a more general approach in *BadRAG*. Contrastive Optimisation on a Passage (COP) leverages gradient-based learning to craft passages of a given length that are retrieved only in the presence of a specific trigger word in the query. This represents a broader attack with an additional degree of stealth, but still relies on a fixed, user-selected trigger that is held constant during the optimisation process.

In contrast, Cheng et al. (2024) [6] introduce a model-level attack in *TrojanRAG*, where the retriever is fine-tuned on poisoned query–passage pairs containing certain trigger tokens. This inserts persistent backdoors into the retriever's embedding space, allowing the attack to generalise across multiple triggers and queries. TrojanRAG achieves strong results across different retriever architectures and LLMs, but assumes that the attacker can modify the retriever's parameters.

Together, these works expose several attack surfaces in RAG pipelines but stop short of exploring the inverse problem: trigger optimisation for a fixed passage.

In *AgentPoison*, Chen et al. (2024) [7] present a attack that targets RAG-based LLM agents that follow multi-step reasoning protocols. It jointly optimises both a trigger and the associated adversarial passages using a HotFlip-style gradient-based method. Whilst the attack is effective in influencing agent behaviour, it is designed for RAG-based LLM agents rather than standard RAG-based LLMs. Additionally, its emphasis on multi-step action control leaves open the question of whether joint trigger-passage optimisation can be adapted for retrieval-time attacks in standard RAG systems.

Other notable work includes that of Chen et al. (2025) [10], which influences the opinions expressed by RAG-based LLM systems on sensitive topics by modifying retrieval rankings through surrogate model training. Zhong et al. (2023) [9] demonstrate that inserting a small number of adversarial passages into a corpus can cause retrievers to consistently rank them highly. Wallace et al. (2019) [8] show that fixed triggers can be appended to inputs to cause consistent, targeted failures across various NLP models.

Several techniques facilitate optimisation of discrete language inputs. HotFlip [11] uses gradient-based character substitutions to craft adversarial passages. Attacks such as PoisonedRAG and AgentPoison adopt similar discrete optimisation techniques at the token level. Gumbel-Softmax [12] provides a continuous approximation of categorical sampling, making it possible to apply gradient-based optimisation to discrete variables.

3 Proposed Contributions

(C1) **Ablation of Adversarial Passage Length**. Conduct a detailed ablation study on adversarial passage length, which is fixed to 50 tokens in the original BadRAG framework. By systematically varying passage length and evaluating its effect on retrieval success and stealthiness, uncover key trade-offs that inform the design of more efficient and covert attacks.

(C2) **Trigger Optimisation for Fixed Passages**. Reverse the BadRAG problem by fixing the poisoned passage and optimising the trigger instead. This formulation introduces a complementary attack surface and allows us to explore how trigger length and position influence retrieval effectiveness.

(C3) **Joint Trigger–Passage Optimisation**. Build on (C2) by jointly optimising both the trigger and the poisoned passage, treating them as a coupled pair. This formulation will necessitate the use of alternative optimisation techniques and provide insight into how coordinated query and corpus manipulations can enhance the effectiveness of retrieval-time attacks.

(C4) **Improving Fluency of Poisoned Passages**. Investigate methods to improve the naturalness of poisoned passages generated in (C3), aiming to reduce detectability. Specifically, explore incorporating fluency constraints during optimisation and partially constraining edits to preserve LLM-generated content, balancing attack effectiveness with linguistic plausibility.

4 Potential Impact

This project introduces a novel perspective on corpus-level attacks in RAG systems by systematically exploring the roles of passage length, trigger structure, and joint optimisation techniques. In particular, the use of gradient-guided trigger learning and differentiable optimisation over discrete text expands the methodological toolkit for studying retrieval-time adversarial behaviour.

The findings will lay groundwork for future adversarial research, including better understanding of how and why retrieval-time attacks succeed, and how to optimise them for greater stealth, transferability, or domain adaptation. These insights also contribute to the broader goal of aligning retrievers with intended behaviour in modular LLM pipelines. But beyond improving understanding of attack surfaces, this work can inform the design of more robust retrieval systems. By identifying configurations that maximise stealth and generalisability, the findings can help guide future defences – such as detection heuristics or input sanitisation.

5 Feasibility & Timeline

This project builds on well-established codebases: both *PoisonedRAG* and *BadRAG* have open-source implementations available on OpenReview. Experiments will be run on the Natural Questions (NQ) dataset [13], which is widely used in retrieval-augmented generation research. I also have access to Imperial's GPU cluster, enabling efficient model training, gradient-based optimisation, and large-scale retrieval evaluation. Given this foundation, the project is technically feasible within the proposed timeline.

w/c	Research Activities	Potential Challenges
7 Jul	(C1) Conduct ablation study on passage length. Vary the number of tokens in adversarial passages and evaluate performance using rank/similarity metrics. This isolates the role of passage length in retrieval dynamics.	Longer passages may dominate retrieval by token mass; results may need length-normalised scoring
14 Jul	(C2) Implement trigger optimisation using HotFlip-style gradient guidance. Systematically vary trigger length and insertion position (start, end, random). This reveals structural factors that influence trigger efficacy.	Semantically odd or degenerate triggers may emerge; balancing stealth and impact
21 Jul	(C3) Develop joint optimisation of trigger–passage pairs. Start with HotFlip-style token substitutions, then move to Gumbel-Softmax sampling for smoother convergence. Compare to independent (disjoint) optimisation baselines.	Instability in joint optimisation; results sensitive to learning rate and sampling temperature
28 Jul	(C4) part 1. Incorporate fluency constraints into passage generation. First, add a language model perplexity penalty to the loss function.	Fluency constraints may reduce attack strength; balancing readability and retrieval control
4 Aug	(C4) part 2. Test constrained optimisation where only the first and/or last tokens of an LLM-generated passage are editable. Evaluate both naturalness and attack performance.	As above
11 Aug	Finalise all experiments. Modularise implementations, log all hyperparameters and outcomes. Prepare ablation and comparison plots.	Managing GPU usage across experiments
18 Aug	Write evaluation scripts; generate plots, tables, and visualisations. Start drafting results and discussion for project report.	This may take some time; allowing two weeks
25 Aug	Final code and report clean-up. Check reproducibility and documentation. Archive datasets and model checkpoints.	

Ethical Considerations. This project studies vulnerabilities in RAG pipelines with the goal of improving their robustness. No personal data will be used, and all experiments will be conducted on public datasets. Results will be framed to support responsible disclosure and defence-oriented analysis, in line with established norms in adversarial research.

References

[1] Wang, C., Ong, J., Wang, C., Ong, H., Cheng, R. & Ong, D., 2023. Potential for GPT technology to optimize future clinical decision-making using retrieval-augmented generation. Annals of Biomedical Engineering, pp.1–4.

[2] Loukas, L., Stogiannidis, I., Diamantopoulos, O., Malakasiotis, P. & Vassos, S., 2023. Making LLMs worth every penny: Resource-limited text classification in banking. In: Proceedings of the ACM International Conference on AI in Finance (ICAIF).

[3] Mahari, R.Z., 2021. AutoLaw: Augmented legal reasoning through legal precedent prediction. arXiv preprint arXiv:2106.16034.

[4] Qi, J., Chen, H., Liu, Q., Li, L., Ni, J., Yang, Y. & Yu, P. S. (2024) PoisonedRAG: Poisoning the Retrieval-Augmented Generation via Corpus Distortion. *arXiv*. [Preprint] https://arxiv.org/abs/2402.12168. [Accessed 24th June 2025].

[5] Xue, J., Zheng, M., Hu, Y., Liu, F., Chen, X. & Lou, Q. (2024) BadRAG: Identifying Vulnerabilities in Retrieval Augmented Generation of Large Language Models. *arXiv*. [Preprint] https://arxiv.org/abs/2406.00083. [Accessed 24th June 2025].

[6] Liu, X., Chen, L., Han, K., Cheng, W., Shen, C. & Zhang, J. (2024) TrojanRAG: Clean-Label Backdoor Attacks on Retrieval-Augmented Generation. *arXiv*. [Preprint] https://arxiv.org/abs/2405.13008. [Accessed 24th June 2025].

[7] Liu, W., Liu, C., Chen, H., Liu, Q., Li, L., Yang, Y. & Yu, P. S. (2024) AgentPoison: Poisoning Language Agents with Dictionary Attacks. *arXiv*. [Preprint] https://arxiv.org/abs/2402.07953. [Accessed 24th June 2025].

[8] Wallace, E., Feng, S., Kandpal, N., Gardner, M. & Singh, S. (2019) Universal Adversarial Triggers for Attacking and Analyzing NLP. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China, pp. 2153–2162. https://doi.org/10.18653/v1/D19-1221.

[9] Zhong, Z., Huang, Z., Wettig, A. & Chen, D. (2023) Poisoning Retrieval Corpora by Injecting Adversarial Passages. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Singapore, pp. 13764–13775. https://doi.org/10.18653/v1/2023.emnlp-main.849.

[10] Wang, A., Zhang, Y., Huang, S., Liu, C., Yang, Y. & Yu, P. S. (2024) FlippedRAG: Retrospective Retrieval-Augmented Generation with Poisoned Demonstrations. *arXiv*. [Preprint] https://doi.org/10.48550/arXiv.2501.02968.

[11] Ebrahimi, J., Rao, A., Lowd, D. & Dou, D. (2018) HotFlip: White-Box Adversarial Examples for Text Classification. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Melbourne, Australia, pp. 31–36. https://doi.org/10.18653/v1/P18-2006.

[12] Jang, E., Gu, S. & Poole, B. (2016) *Categorical reparameterization with Gumbel-Softmax*. arXiv preprint arXiv:1611.01144. http://dx.doi.org/10.48550/arXiv.1611.01144.

[13] Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., et al. (2019) Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics.* 7, 453–466. https://doi.org/10.1162/tacl_a_00276.