Path-Aware Defense Against Prompt Injection Attacks in Retrieval-Augmented Large Language Models

Damien L. Perez
School of Information Technology, University of Cincinnati, Cincinnati, OH, USA.
damienperez@uc.edu

Iecac Rhodes
Department of Computer Science, University of Central Florida, Orlando, FL, USA.
isaac356@ucf.edu

Abstract

Retrieval-augmented generation (RAG) architectures have rapidly become a dominant paradigm for grounding large language model outputs in external knowledge, yet they introduce novel attack surfaces through prompt injection that threaten the integrity, confidentiality, and reliability of downstream applications. This paper presents a comprehensive system-level analysis of path-aware defense mechanisms designed to counter adversarial prompt injections in RAG pipelines. Departing from conventional input-filtering approaches, path-aware strategies operate on the provenance and trajectory of information as it moves from retrieval sources through fusion and generation stages, enabling fine-grained attribution, anomaly detection, and targeted intervention. The discussion examines structural trade-offs between defense granularity, latency, scalability, and model performance, and it locates path-aware methods within broader architectural patterns for robust AI infrastructure. The paper further addresses governance frameworks, fairness implications, operational resilience, and long-term sustainability, highlighting how path-level visibility transforms incident response, auditability, and regulatory compliance. By synthesizing insights from distributed systems, adversarial machine learning, and information retrieval, this work articulates a research roadmap for building resilient RAG ecosystems in which safety is not an afterthought but a property of the retrieval-generation pathway itself.

Keywords

retrieval-augmented generation, prompt injection, adversarial robustness, path-aware defense, large language models, information retrieval, system security.

1. Introduction

The integration of retrieval mechanisms into large language models has fundamentally altered the landscape of natural language processing by enabling systems to dynamically access external knowledge repositories during inference. Retrieval-augmented generation architectures, which combine document retrieval with generative transformers, have demonstrated remarkable improvements in factual accuracy, domain adaptability, and knowledge-intensive task performance compared to parametric models alone [1], [2]. These hybrid systems are now deployed across high-stakes domains including medical question answering, legal research, financial analysis, and enterprise knowledge management, where the reliability of generated information is paramount. However, the very components that

confer these advantages also introduce structural vulnerabilities that adversaries can exploit through prompt injection attacks, threatening the trustworthiness of the entire pipeline [6], [7]. Unlike traditional adversarial attacks focused on perturbing input tokens to cause misclassification, prompt injection in RAG systems operates by contaminating the retrieval corpus or manipulating the interaction between retrieved context and the generative model, creating subtle yet dangerous failures that can propagate silently through deployments.

The urgency of addressing prompt injection in RAG systems stems from their socio-technical embedding in decision-support infrastructures where errors can have cascading consequences. As these systems move from research prototypes to production services, the attack surface expands beyond the model weights to include the retrieval index, the query formulation process, the fusion logic that combines retrieved passages with user prompts, and the final generation step. Existing defense strategies largely focus on input sanitization, adversarial training, or post-hoc output filtering, which treat the system as a black box and ignore the rich structural information available in the retrieval-generation pathway. This paper argues for a paradigm shift toward path-aware defense, wherein the full trajectory of information from retrieval source to generated token is instrumented, monitored, and protected. By attributing outputs to specific retrieval paths and detecting deviations from expected provenance patterns, path-aware mechanisms can identify and neutralize injection attacks that would otherwise bypass perimeter defenses. The following sections develop a systemic perspective on this approach, analyzing architectural design choices, governance implications, sustainability challenges, and the broader policy landscape that will shape the adoption of resilient RAG systems.

2. The Emerging Threat Landscape of Prompt Injection in RAG Systems

Prompt injection attacks against retrieval-augmented language models can be categorized along multiple dimensions that reflect the layered architecture of these systems. Direct injection occurs when an adversary inserts malicious instructions into the user-facing prompt in an attempt to override the system prompt or exfiltrate sensitive context, while indirect injection targets the external documents that will be retrieved and integrated into the model's working memory during inference [7]. The latter vector is particularly insidious because it allows attackers to compromise many downstream interactions by poisoning a single upstream knowledge source, effectively turning the retrieval corpus into a persistent threat. For instance, a malicious actor might inject hidden instructions into a webpage that a RAG system indexes, causing the model to ignore safety constraints whenever that page is retrieved for relevant queries. These attacks exploit the trust boundary between the retrieval corpus—often treated as an unexamined ground truth—and the controlled generation environment, creating a structural mismatch that input-only defenses cannot resolve.

The severity of prompt injection in RAG systems is amplified by the interaction between retrieval relevance and adversarial influence. When a retrieval system returns a highly relevant document that also contains injected prompts, the generative model naturally attends strongly to that content, making the injected instructions particularly effective. This phenomenon creates a direct tension between the utility of high-relevance retrieval and the risk of amplifying malicious content, a trade-off that naive filtering approaches fail to balance adequately. Moreover, the composability of RAG pipelines—where multiple retrieval sources, rerankers, and fusion modules are chained together—introduces complex failure modes in which an injection might only manifest its effects under specific combinations of query, retrieved set, and model state. Such stateful attacks are difficult to reproduce and detect in

offline testing, underscoring the need for runtime monitoring that captures the full information path. The landscape is further complicated by the emergence of multi-turn conversational RAG systems, where an injection planted in an earlier turn can persist in the dialogue history and influence later interactions, enabling long-lived compromise vectors that evade single-turn defenses [19].

Cross-domain comparisons with traditional information security illuminate both the novelty and the familiarity of prompt injection challenges. In database management systems, SQL injection attacks similarly exploit the commingling of data and control planes, and decades of defensive practice have demonstrated that parameterized queries—which enforce a strict separation between structure and content—provide robust protection. RAG systems currently lack analogous architectural separation; retrieved content and system instructions are often concatenated into a monolithic prompt string that the language model parses holistically, erasing provenance boundaries. While some proposals advocate prompt hardening through escape sequences or delimiter conventions, these are surface-level mitigations that fail to address the fundamental conflation of control and data. The path-aware perspective instead draws on lessons from provenance tracking in distributed systems and taint analysis in software security, proposing that the lineage of each piece of information should be explicitly represented and validated throughout the pipeline. This framing situates RAG security within a broader tradition of defense-in-depth, where multiple layers of path attestation, anomaly detection, and dynamic intervention cooperate to contain threats before they influence generation outcomes.

3. Foundations of Path-Aware Defense

Path-aware defense for retrieval-augmented generation rests on the principle that the provenance, transformation, and influence of information within a system constitute a rich signal for detecting and mitigating adversarial manipulations. At its core, the approach involves instrumenting the RAG pipeline to maintain a directed attribution graph that records how each retrieved passage contributes to the final generated sequence, including intermediate steps such as reranking, aggregation, and cross-attention fusion. This graph enables the system to answer questions such as “which retrieved segments most influenced this output token?” and “did any retrieved content originate from an untrusted or anomalous source?” By making these dependencies explicit, path-aware mechanisms transform the defense problem from a brittle binary classification of inputs into a continuous monitoring and intervention framework that can respond to threats with surgical precision. The concept draws on the path-level intervention paradigm, where safety mechanisms operate on the causal chains linking retrieval to generation rather than on surface-level features [13].

The conceptual foundations of path-aware defense intersect with several established research traditions. In program analysis, dynamic information flow tracking techniques label data with security metadata as it propagates through a system, enabling the detection of tainted values at sensitive sinks. Analogously, a path-aware RAG system can tag each retrieved document with its source, integrity level, and chain of custody, then propagate these tags through the encoding and cross-attention layers of the generative model. When the model attends to a substantially tainted segment during generation, the defense system can apply graduated responses ranging from down-weighting the influenced logits to blocking the output entirely and triggering a retrieval fallback. This approach avoids the all-or-nothing nature of input gatekeeping and allows legitimate retrieved content to inform outputs while containing the blast radius of any injected material. Furthermore, attribution graphs support explainability

and auditing requirements that are increasingly mandated in regulated industries, as they provide concrete evidence of why a particular output was produced and which sources contributed to it. This transparency is valuable not only for security forensics but also for building user trust and meeting algorithmic accountability obligations under frameworks such as the EU AI Act [10].

Designing effective path attribution for large transformer models poses significant systems challenges that must be addressed without sacrificing inference efficiency. Naively tracking every attention head's contribution to every output token would impose prohibitive memory and computational overhead, necessitating approximate attribution methods that aggregate path information at the passage or document level rather than at the token level. Techniques derived from integrated gradients and attention rollout provide scalable alternatives that can be computed alongside the forward pass with modest overhead when implemented in optimized inference runtimes. The choice of attribution granularity involves a trade-off between detection sensitivity and performance impact: finer-grained tracking can catch subtly injected instructions that hide within otherwise benign passages, while coarser tracking inexpensively identifies bulk anomalies such as entire documents from compromised domains. A productive design pattern is to deploy coarse path monitoring as a continual low-cost background process and escalate to fine-grained attribution only when anomalies are detected, creating a hierarchical defense that adapts its resource consumption to the threat level. This architectural principle mirrors the tiered alerting systems common in network security operations centers, where deep packet inspection is reserved for suspicious flows while most traffic receives lightweight scrutiny.

4. Architectural Integration and System Trade-Offs

Integrating path-aware defenses into production RAG infrastructures requires careful consideration of the end-to-end system architecture, including indexing pipelines, retrieval services, model serving, and application logic. A modular design is essential to avoid tight coupling between the path attribution subsystem and the core generation engine, which must remain independently updateable as model architectures evolve. The preferred integration pattern positions a path monitor as a sidecar process that observes the interaction between the retrieval backend and the language model, consuming attribution metadata and issuing intervention signals without being on the critical path for normal processing. This design enables the defense mechanism to be deployed incrementally, tested in shadow mode alongside existing systems, and rolled back independently if performance or stability issues arise. The sidecar approach also facilitates compliance with organizational security policies that may require third-party auditing components to be isolated from core inference infrastructure.

The latency implications of path-aware monitoring are a primary concern for user-facing applications with strict response time requirements. The additional computation required for attribution tracking and anomaly scoring must be overlapped with generation as much as possible through pipelining and parallelization. Modern transformer serving frameworks already exploit operator fusion and continuous batching, and path metadata can be threaded through these optimizations without fundamentally disrupting the execution model. Where generation is performed autoregressively, path scores can be computed incrementally and used to trigger early termination if injection is detected mid-generation, thereby saving the cost of producing subsequent tokens that would be discarded. This early-stopping capability actually improves system efficiency under attack, partially offsetting the steady-state

monitoring overhead. For bulk offline processing scenarios such as document summarization or report generation, the relative overhead of path tracking is even less significant, and the richer tracing information can be stored for post-hoc analysis and model debugging.

Scalability across multi-tenant and multi-index deployments introduces additional complexity. Large organizations often maintain separate retrieval indices for different departments, knowledge domains, or security tiers, and a RAG system may query multiple indices in parallel before fusing results. Path-aware defenses must correctly attribute information to its originating index and apply per-index trust policies that reflect organizational data governance rules. For example, retrieved content from a public web index should be treated as inherently less trustworthy than content from an internal, curated knowledge base, and the path monitoring system should encode these differential trust levels in its scoring logic. This requirement drives the need for a policy engine that translates high-level organizational security postures into concrete path-threshold configurations, enabling non-expert administrators to manage defense parameters without deep machine learning expertise. The policy engine can also implement context-dependent rules, such as requiring higher provenance assurance for outputs in financial or medical contexts than for general conversational queries. Designing such policy frameworks is as much a socio-technical governance challenge as an engineering one, demanding collaboration between security architects, domain experts, and legal teams.

Comparing path-aware defense to alternative mitigation strategies clarifies its relative advantages and limitations. Input sanitization approaches that strip suspicious patterns from user queries are computationally cheap and easy to deploy but cannot address indirect injection via retrieved documents and are prone to false positives that degrade legitimate queries. Adversarial training methods that fine-tune models on injected examples improve robustness to known attack patterns but struggle to generalize to novel injection strategies and incur substantial training costs at each model update. Constitutional AI techniques that embed safety instructions directly into the system prompt can be overridden by sufficiently forceful injected instructions [20]. Path-aware defense complements rather than replaces these methods: sanitization can serve as a first-pass filter to reduce the load on the path monitor, adversarial training can harden the model's resistance to residual injection that passes path checks, and constitutional prompts can provide an additional safety backstop. The resulting layered defense architecture mirrors successful patterns from cybersecurity, where no single control is expected to be impenetrable, and the combination of diverse, overlapping protections yields system-level resilience.

5. Governance, Fairness, and Operational Resilience

The adoption of path-aware defenses in retrieval-augmented language models carries significant implications for organizational governance and regulatory compliance. As jurisdictions worldwide introduce AI accountability frameworks, the ability to demonstrate that outputs are traceable to specific, vetted sources becomes a critical capability for risk management and legal defensibility. Path attribution graphs serve as auditable artifacts that can be produced during incident investigations, regulatory inspections, or litigation to show the exact provenance of contested outputs. This capability shifts the compliance burden from retrospective guesswork to structured forensic analysis, potentially reducing liability exposure for deployers who can demonstrate diligent monitoring. Furthermore, the granularity of path information aligns with emerging standards for AI system documentation, such as model cards and datasheets for datasets, by extending the concept of provenance from training data

to run-time information flows [3]. Organizations that instrument their RAG pipelines early with path tracking will be better positioned to meet future regulatory requirements without disruptive retrofitting.

Fairness considerations in path-aware defense design require careful attention to avoid introducing new forms of bias or exacerbating existing inequities. The trustworthiness scores assigned to retrieval sources must be calibrated to avoid systematically discounting content from marginalized communities, non-Western sources, or minority languages that may be underrepresented in curated knowledge bases. If the path monitor is configured with conservative thresholds that favor content from established institutional sources, it could suppress valuable perspectives from grassroots organizations, independent researchers, or oral knowledge traditions that are less likely to appear in authoritative corpora. Mitigating this risk requires participatory design processes that involve diverse stakeholder groups in defining trust criteria, as well as technical mechanisms that allow users to inspect and challenge source-level trust decisions. Transparency dashboards that surface which sources were down-weighted and why can empower users to make informed judgments about the completeness and fairness of the system's outputs, turning the path-aware infrastructure from a purely protective mechanism into a platform for algorithmic contestation and accountability.

Operational resilience is enhanced by the detection granularity that path-aware systems provide. In the event of a known corpus poisoning incident, the attribution graph enables precise identification of all generated outputs that were influenced by the compromised documents, facilitating targeted retractions or corrections rather than blanket service shutdowns. This capability is critical in high-availability environments such as healthcare decision support or financial trading, where suspending AI assistance entirely carries its own risks. Path-based incident response also supports differential containment strategies: outputs with minor influence from a tainted source might be flagged with a caveat and allowed to remain visible, while outputs heavily dependent on compromised content can be automatically retracted. Such proportional responses maintain service continuity while managing risk exposure, reflecting a mature operational security posture. The integration of path monitoring with organizational security information and event management systems further enables correlation of AI-level anomalies with broader infrastructure events, such as detecting that a spike in injection attempts coincides with a known phishing campaign targeting content authors whose material feeds the retrieval index.

6. Sustainability, Deployment, and Long-Term Robustness

The long-term sustainability of path-aware defenses depends on their ability to adapt to the co-evolution of retrieval corpora, language model capabilities, and adversarial tactics. Retrieval indices in production environments are dynamic, continuously ingesting new documents and expiring stale ones, which means that the distribution of provenance paths shifts over time. A path monitor configured with static thresholds will gradually lose effectiveness as the statistical properties of the corpus drift, necessitating online adaptation mechanisms that recalibrate anomaly detectors based on recent observed path patterns. This adaptation must be robust to adversarial manipulation of the distribution itself, as sophisticated attackers might attempt to slowly introduce malicious content that shifts the baseline in a favorable direction before launching an injection campaign. Techniques from robust statistics and online anomaly detection provide principled approaches to maintaining detection efficacy under distribution shift, but their integration into RAG path monitors

remains an open engineering challenge that will determine the operational lifespan of these defenses.

Deployment across heterogeneous hardware environments—from cloud data centers to edge devices running quantized models—requires careful abstraction of the path monitoring interface to ensure portability. The attribution computation must be separable from the specifics of the model architecture and inference runtime, enabling a single path-aware policy engine to govern multiple model variants and hardware backends. This goal can be advanced through standardized attribution APIs that expose token-level influence scores in a model-agnostic format, analogous to how hardware performance counters provide a uniform interface to diverse processor architectures. The development of such standards would lower the barrier to entry for path-aware defense adoption and encourage a competitive ecosystem of monitoring tools, similar to the vibrant market that emerged around application performance management. Industry consortia and open-source communities are well-positioned to drive this standardization, ensuring that path attribution does not become a proprietary lock-in mechanism but rather a shared infrastructure for trustworthy AI.

The adversarial arms race between injection techniques and path defenses will inevitably escalate, and system designers must plan for continuous improvement cycles rather than one-time hardening. Attackers will develop methods to camouflage injected instructions as benign stylistic patterns, split malicious payloads across multiple documents to evade single-source attribution, or exploit edge cases in the attribution algorithm itself to cause false negatives. Defending against these adaptive threats requires a combination of regular red-teaming exercises that simulate sophisticated adversaries, automated regression testing of path monitor performance against a curated injection corpus, and architectural provisions for rapid deployment of updated detection signatures without model retraining. The sustainability of the defense ecosystem also depends on the availability of shared threat intelligence, so that an injection technique discovered against one organization's RAG deployment can be rapidly mitigated across the entire sector. Building such intelligence-sharing mechanisms while respecting competitive sensitivities and privacy regulations is a governance challenge that parallels those faced in cybersecurity threat information sharing, and similar institutional models—such as information sharing and analysis centers—may prove effective.

Resource consumption is an underappreciated dimension of sustainability for path-aware defenses at scale. While the per-query computational overhead may be modest, the aggregate energy and hardware costs of running continuous attribution monitoring across millions of daily queries can become environmentally significant. This concern intersects with broader efforts to reduce the carbon footprint of large-scale AI deployments [3]. Optimizing path tracking to minimize redundant computation—for instance, by caching attribution patterns for frequently retrieved documents or sharing attribution substructures across queries that retrieve overlapping passage sets—can substantially reduce the marginal energy cost. Moreover, the early-stopping capability enabled by path monitoring, which aborts generation upon detecting injection, can reduce the total computational work performed under attack, yielding a net energy savings in adversarial scenarios. Lifecycle analysis of path-aware RAG systems should account for these dynamic factors rather than assuming a fixed overhead, and system designers should prioritize algorithmic efficiency alongside detection accuracy when selecting attribution methods for production use.

7. Conclusion

Path-aware defense represents a structural reimagining of security for retrieval-augmented large language models, shifting the emphasis from surface-level input filtering to deep instrumentation of information provenance and influence throughout the generation pipeline. This paper has argued that such an approach is necessitated by the growing sophistication of prompt injection attacks that exploit the trust boundaries, composability, and statefulness of modern RAG architectures. By maintaining explicit attribution graphs that link outputs to their retrieval underpinnings, path-aware systems enable precise threat detection, proportional intervention, and rich forensic auditability that align with emerging regulatory expectations and operational resilience requirements. The systems analysis presented here underscores that effective defense cannot be achieved through any single technique; rather, path monitoring must be embedded within a layered security architecture that combines input sanitization, adversarial training, policy engines, and adaptive response mechanisms.

The path forward involves interdisciplinary collaboration spanning distributed systems, machine learning, security engineering, and technology policy. The development of standardized attribution interfaces, efficient approximation algorithms, and adaptive detection frameworks will determine the technical feasibility of path-aware defenses at global scale. Equally important are the governance innovations that ensure these mechanisms are deployed in ways that respect fairness, transparency, and sustainability. As retrieval-augmented generation continues its trajectory from research novelty to societal infrastructure, the choices made today about how to instrument and protect the retrieval-generation pathway will shape the trustworthiness of AI-mediated information access for years to come. Path-aware defense offers not a finished solution but a promising direction for embedding safety into the very architecture of knowledge-grounded AI systems, making robustness a property of the path rather than a patch upon its surface.

References

1. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems, 33, 9459-9474.

2. Guu, K., Lee, K., Tung, Z., Pasupat, P., & Chang, M. W. (2020). Retrieval augmented language model pre-training. In International Conference on Machine Learning (pp. 3929-3938). PMLR.

3. Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (pp. 610-623).

4. Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., ... & Oprea, A. (2021). Extracting training data from large language models. In 30th USENIX Security Symposium (pp. 2633-2650).

5. Wallace, E., Feng, S., Kandpal, N., Gardner, M., & Singh, S. (2019). Universal adversarial triggers for attacking and analyzing NLP. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (pp. 2153-2162).

6. Perez, E., & Ribeiro, M. T. (2022). Ignore previous prompt: Attack techniques for language models. In NeurIPS ML Safety Workshop.

7. Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., & Fritz, M. (2023). Not what you've signed up for: Compromising real-world llm-integrated applications with indirect

prompt injection. In Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security (pp. 79-90).

8. Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., ... & Fedus, W. (2022). Emergent abilities of large language models. Transactions on Machine Learning Research.

9. Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., ... & Liang, P. (2021). On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258.

10. Dinan, E., Humeau, S., Chintalapudi, B., & Weston, J. (2019). Build it break it fix it for dialogue safety: Robustness from adversarial human attack. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (pp. 4537-4546).

11. Papernot, N., McDaniel, P., Sinha, A., & Wellman, M. P. (2018). SoK: Security and privacy in machine learning. In 2018 IEEE European Symposium on Security and Privacy (EuroS&P) (pp. 399-414).

12. Ram, O., Levine, Y., Dalmedigos, I., Muhlgay, D., Shashua, A., Leyton-Brown, K., & Shoham, Y. (2023). In-context retrieval-augmented language models. Transactions of the Association for Computational Linguistics, 11, 1316-1331.

13. C. Shi, S. Li, W. Lu, W. Wu, C. Wang, Z. Cheng, F. Shen, and T. Chua (2026)TraceRouter: robust safety for large foundation models via path-level intervention.arXiv preprint arXiv:2601.21900.

14. Borgeaud, S., Mensch, A., Hoffmann, J., Cai, T., Rutherford, E., Millican, K., ... & Sifre, L. (2022). Improving language models by retrieving from trillions of tokens. In International Conference on Machine Learning (pp. 2206-2240). PMLR.

15. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

16. Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., ... & Yih, W. T. (2020). Dense passage retrieval for open-domain question answering. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 6769-6781).

17. Thoppilan, R., De Freitas, D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H. T., ... & Le, Q. (2022). LaMDA: Language models for dialog applications. arXiv preprint arXiv:2201.08239.

18. Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., ... & Irving, G. (2019). Fine-tuning language models from human preferences. arXiv preprint arXiv:1909.08593.

19. OpenAI (2023). GPT-4 technical report. arXiv preprint arXiv:2303.08774.

20. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of NAACL-HLT (pp. 4171-4186).

21. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692.