International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 7, Issue 1, January - February 2024 ||

DOI: 10.15680/IJCTECE.2024.0701009

The End of Generative AI Experiments Designing Production-Grade Data Architectures for LLM Systems

Samanth Gurram

Engineering Manager, Data & AI, USA

ABSTRACT: The paper explains the causes of AI pilots' failures, which are often due to flaws in the model or volatile data. Using a quantitative experimental design, the paper compares and contrasts the big parameter models and the specific models with and without Retrieval Augmentation Generation (RAG). The results showed that data volatility, delay in indexing, and low ingestion pipelines were the primary sources of significant performance loss. AI pilots that operate in Generative AI worked well in the controlled tests, but when they were to perform in the actual production setting, several of them collapsed. The smaller models with strong pipelines were beneficial because of their cost-adjusted value. Risks are governance and security, which began escalating as systems were introduced in the production environments.

KEYWORDS: Generative AI, Large Language Models (LLMs), Retrieval-Augmented Generation (RAG), Data Volatility

I. INTRODUCTION

The demonstrations of the generative AI systems in some studies are on the basis of Large Language Models that have proven themselves in both research and pilot applications. Some researchers, however, noted that there was a drop in performance when these systems were subject to actual enterprise data. Common aspects of production environment are: update of documents with high frequency, alteration of schemas, low latency, and hard governance needs. The paper examines the model limitations and poor data architectures that lead to GenAI pilot failures. We test several system structures in different volatility and latency environments. This paper dwells on the Retrieval-Augmented Generation, the cost-effectiveness, and governance compliance to comprehend the way in which systems of production-grade low-level machine learning (LLM) need to be fabricated so that they are stable and can be redeployed.

II. RELATED WORKS

Retrieval-Augmented Generation

Large Language Models (LLMs) have been effective in many natural languages. The parameters they possess contain enormous amounts of factual data that can be narrowed down to downstream activities [2]. However, LLMs are prone to hallucinations, and they give outdated information and cannot strongly specify their means of attaining their solutions [1]. Their logic is not necessarily evident, and they need to be re-educated or polished so as to modernize their knowledge, which is an expensive and time-intensive task [2].

Parametric memory allows patterns to be memorised using training data on the model; however, it does not perform well in the recall of the correct knowledge and the ability to control [2]. Highly parametric models tend to be worse than task-based retrieval systems on activities involving knowledge [2]. The derived disconnect led to the development of Retrieval-Augmented Generation (RAG), where a model is combined with a parametric internal and a non-parametric external (document indexes) memory [2].

RAG is a joint model that consists of a sequence-to-sequence generator and a neural retriever. The retriever then scans the dense index of vectors (in this case, Wikipedia) and provides the passages that are relevant to the generator [2]. Two main sets of the RAG formulation under consideration were the first formulation, where the same passages were looked at on the entire sequence, and the other form allowed dissimilar passages on every token produced [2]. According to the findings, RAG is more factual and produces more specific and diverse results compared to parametric-only baselines [2].

A larger view of RAG systems shows that three underlying aspects are at work, namely retrieval, generation, and augmentation [1]. The initial version of Naive RAG to Advanced and Modular RAG structure acquired RAG through the additions of a new variant, dubbed RAG [1]. The development of such structures improves indexing plans, retrieval ranking, pipelines, and gauge benchmarks. The model weights do not entirely belong to knowledge anymore, but they are actually retrieved from the external databases.

The language modeling of the IR technique ranks the documents based on the likelihood of the documents generating a query [8]. Such a probabilistic relationship between the generation and retrieval of text was also already true in the case of the modern LLMs, even before the modern generation and retrieval of text. The new technical paradigm, which is the integration of the IR models with the LLMs, provides real-time information provided by the IR systems, reasoning and generation by the LLMs, and judgment by humans [6].

The other observation that is found in the literature is that the performance-compute trade-offs can be maximized through retrieval [5]. The only way is not to increase the size of the model to provide efficient solutions to tasks. The Retrieval process allows models to scale the context without any parameters added [5]. This way, the system complexity is deconvolved with the model size into the retrieval infrastructure. This is the core of the events that failed most GenAI pilots. The model was robust, and the data infrastructure was weak.

Data-Centric Architectures

The primary interest of LLMs was in scaling laws and the performance of training. The neural probabilistic language models had lower training costs and data computation costs, e.g., noise-contrastive estimation [7]. Such improvement was the reason for making large-scale language modeling. There are other production systems issues, which include data freshness, indexing latency, schema change, and control.

The non-parametric memory is projected by RAG in the form of external indexes [2]. This needs sparse representations of the documents that hold and update the embeddings on being rewritten and low-latency recovery. With the overall direction of the analysis of the RAG paradigms, it can be observed that the quality of retrieval, the strategy of augmentation, and the indicators of assessment have an immediate impact on the performance in the long term [1]. At this, the bottleneck is shifted to one of the training, ingestion, and indexing.

The IR investigations provide underlying concepts on rankings, query modelling, and probabilistic matching [8]. These ideas are applied to a contemporary LLM to train IR, i.e., the integration of a neural retriever and a generative model [6]. The cost of computation and plausibility problem is still to be implemented [6]. These issues are the operational risks in production.

Data in the real world is messy, inconsistent, and domain-specific [4]. Although the use of GPT-based models can possibly be applicable to data transformation when the number of shots is minimal, reliability is the most significant problem of concern [4]. The minor errors in data can be tolerated in the case of an experiment. Such production errors are passed to the factors of retrieval index records, and this reduces the quality of responses.

This is the reason why so many GenAI pilots crashed this round, who were exposed to the volatility of information in reality. The test data are normally dynamic and pure. The production data are subjectively adjusted on a daily basis; the fields are not homogeneous, of different formats, and are constrained. The absence of powerful data pipelines leads to stagnation when it comes to embeddings and results of the retrieval process drift. The performance of the RAG [1][2] will determine the quality and the freshness of the indexed data.

There are other retrieval theories that are also trying to recreate deeper senses of meaning with the help of traces of documents, like semantic model theory, and quantum-inspired theories [10]. The extraction, in such a manner, is not only pointed out to develop an interest in replication of key words but also the representation of the meaning and latent model. There are other deep learning models, such as Deep Boltzmann Machines, which aim at retrieving document-related representations as latent semantics representations [9]. These readings demonstrate that the procedures of enacting representation learning and indexing are complex processes. LLMOps can therefore be viewed as an extension of contemporary-day data engineering. It includes ETL pipes, in-built generation, index maintenance, recovery quality loop, and assessment loop.

Task-Specific Models

The literature also doubts the notion of bigger models being the best providers of value. Techniques like efficient attention, recurrence, and conditional computation have a limitation when it comes to scaling [5]. To some tasks, a mere increase in parameter size causes a high cost of computation and diminishing returns [5]. Retrieval-based methods are able to minimize the supervisory requirement and elongate context in an efficient manner [5].

It is empirically established that the RAG models outperform parametric-only models based on parametric seq2seq models on tasks where the context involves open-domain questions in open-book mode on question answering [2]. This is achieved using increased knowledge of access and not the size of the model. In most enterprise environments, smaller task-specific models with powerful retrieval pipelines are able to achieve lower cost and competitive results.

The IR model summary contains the note that integrating IR models with LLM and human evaluation formed a more potent system compared to the usage of a big model only [6]. The interpretation of this hybrid perspective is that cost-adjusted value will be based on system design, and not the number of parameters.

The experiments of data wrangling also demonstrate that the LLMs are capable of assisting in a variety of transformation tasks, though reliability is still a problem [4]. A smaller model, which is based on well-organized and well-controlled data, might perform better than a large model that has noisy inputs. This supports the idea that the performance of data pipelines revolves around the quality of data and their resilience.

Past research on training efficiency, including noise-contrastive estimation, has shown that the cost in terms of computation can be decreased, while at the same time, quality is not compromised through algorithmic design [7]. The literature implies four most important lessons. Solely, parametric knowledge cannot be used in dynamic and knowledge-intensive activities [1][2]. Retrieval makes the system more complex towards data gathering, indexing, and upkeep [1][6][8]. Retrieval and hybrid systems have performance-compute trade-offs inferior to those of pure scaling [5][2]. The issue of governance, reliability, and credibility would be critical when systems are running under real environments [4][6].

As GenAI is brought to production, there is a revival of governance, privacy, and security. Access to the external databases is to be controlled, logs are to be audited, and data provenance has to be traced. RAG is partially a solution to the problem of provenance through the association of outputs with retrieved documents [2], but further controls are needed, especially those within operational governance.

The innovation is not terminated by a lack of funding for experiments. It is the transition of model-oriented research to data-oriented system engineering. Trade-sized LLM systems require good data structures, stuff that can be easily retrieved, model choices that are economical, and efficient governance systems. As can be seen in the literature in RAG, IR, neural language modeling, and data wrangling, this move towards experimental success to architectural maturity has been underpinning the literature in the area.

III. METHODOLOGY

Research Design and Experimental Structure

The paper follows a quantitative, controlled experimental type of design to discuss the reasons why most Generative AI (GenAI) pilot systems have failed when placed in the real-world production environment. It does not just focus on model performance, but rather the behavior of systems when data becomes volatile, and must be read or written at end-to-end latency, and when governed by limiting rules. The study makes comparisons of four architecture configurations so as to isolate the influence of model size and the maturity of data infrastructure.

These four systems are: (1) a large parametric-only LLM, (2) a large LLM with Retrieval-Augmented Generation (RAG), (3) a smaller task-specific model with RAG, and (4) a smaller task-specific model without retrieval support. The RAG architectures are based on the parametric and non-parametric memory model proposed in the existing body of retrieval-augmented generation studies [2], and the modular retrieval-generation-augmentation perspective considered in the recent reviews of RAG systems [1]. The implementation of retrieval components relies on dense vector indexing that is based on the current Information Retrieval (IR) techniques [6][8].

The aim of this experiment is to examine whether the performance differentiation of production environments is motivated by model scale or data pipeline robustness.

Dataset Construction and Volatility Simulation

Three types of domain-specific data sets are created to reproduce the realistic setting in enterprises, including a customer support knowledge base, a financial policy repository, and a technical operations log archive. They have about 500,000 documents in each dataset. The semi-structured content is indicative of general enterprise data challenges of irregular formatting, lack of metadata, and business-related domain terms.

In order to reproduce the volatility of the real world, there are some artificial data updates, which are introduced daily at different rates, moderate and high update cases. A renaming of fields and altering document structures, as well as metadata formats, simulate schema drift as well. Additive noise is introduced in order to depict missing entries and irregular records.

The schedules of re-indexing are embedded differently in the experimental runs to represent variations in policies of data freshness. Other systems refresh indexes on an almost real-time basis, whereas others refresh on a delayed basis. This arrangement can be associated with established issues in knowledge updating of parametric models [2] and maintenance of the indexing of RAG systems [1].

System Implementation and LLMOps Pipeline

Every architecture has been carried out in an abstract LLMOps pipeline comprising data ingestion and preprocessing, embedding generation, indexing vectors, model coordination, logging, and monitoring. The ingestion layer is the layer that parses and converts documents, signifying problems associated with information wrangling that have been found as viable issues in the literature to date [4].

Dense embedding indexes under the retrieval layer are used to obtain documents to be generated. This is also a component that is measured based on quality ranking as well as latency during a load. The outcomes of the test show the effect of a delay of retrieval on the response time and relevance of the questions answered.

The monitoring system guarantees that there is real-time monitoring of embedding staleness, retrieval quality, and generation accuracy. Such construction of operations renders LLMOps more of a data engineering process than a modelling field. The operations resemble other existing IR-LLM hybrid systems that were outlined in recent literature [6], and the evaluation loops and retrieval infrastructure are also significant.

Performance and Cost Evaluation

Measures of system performance are done in various dimensions. A set of 10,000 domain-specific queries of each dataset is used as a benchmark in assessing task accuracy. The measurement of Hallucinations is achieved by checking whether the generated responses are in line with the source document retrieved, as had been in the case of RAG evaluation studies [2].

Latency is the total response time, in terms of retrieval and generation. The cost is computed depending on the compute consumption, overhead used in the storage of vector databases, and indexing overhead. The performance comparison is done using a cost-adjusted measure to find out whether small task-specific models with good pipes of data are better valued than the large parametric models. The hypothesis that the number of parameters or the retrieval infrastructure and stability of pipelines are the most predictive of production outcomes is tested in this analysis.

The measures of retrieval quality, which help determine the quality of IR, include IR ranking measures like Recall and Mean Reciprocal Rank, and have proven to be part of the traditional language modeling strategies of document ranking [8]. To identify the interaction effect of model size, retrieval architecture, and data volatility, statistical techniques such as analysis of variance and regression modeling are used to establish the interaction.

Governance, Privacy, and Security Controls

Role-based access control is applied over datasets in order to simulate conditions of production governance. Precise records are reserved for a particular group of users, and delicate columns are optimized to an index before being disguised. Audit logs are used to record the traces of document access and generation.

Security testing gauges the rate of policies being breached and exposure to undesired content in the generated content. The comparison of these risks among architectures is done in order to determine the existence of new governance challenges presented by retrieval-based systems. Previous studies have pointed out evidence of credibility and ethical risks in the IR systems that were closer to the LLM implemented systems [6], usually more pronounced in production implementations.

The experiment also determines the operational trade-offs of the rigorous implementation of governance and system latency. Another example is adding filtering and logging that will raise the response time chances but lower the compliance risk.

Analytical Strategy

The quantitative analysis aims at determining the architectural factors with the highest incentive to accuracy, latency, cost efficiency, and compliance with governance. Effects of interaction between the size of the model and the volatility of the data are evaluated in order to find whether big models deteriorate more in unsteady data conditions. A correlation study is used to assess the correlation between the frequency of index refresh and the rates of hallucinations. Regression models approximate the proportionate value of retrieval latency, the data noise, and schema drift in the entire system. This methodology will propose to show that indexing delay and governance enforcement introduce systematic volatility, which is often the cause of most GenAI pilot failures, rather than a lack of model intelligence. Rather, they were brought about due to poor data pipelines, bad freshness assurances, and small production-grade engineering practices. The paper thus presents the concept of LLMOps as the direct extension of the current data engineering concepts as opposed to a distinct machine learning specialization.

In spite of the main experiments done in a controlled assessment setting, the identified trends were also benchmarked with the operational measures in several enterprise implementations of retrieval-augmented systems. The pattern of behavior observed in observational data about production monitoring dashboards indicated the same tendencies, especially in how the RAG-based architectures were stable under document repositories that were continuously being updated. These forms of operations found that systems based on retrieval pipelines were factually consistent even when the support documentation varied multiple times per day, and that a system based on parametric-only systems needed retraining each time the system became inconsistent. These findings are in line with the experimental findings of this research, especially the conclusion that data pipe freshness and indexing pipelines are the significant contributors to system reliability. Although the observations on the production were not extensive and did not entail a fully controlled experiment, this offers a preliminary external confirmation that the trends realised in the experimental setting are in consonance with the real deployment settings.

IV. RESULTS

Impact of Data Volatility on Model Performance

The initial business case of this paper was to determine whether the failure of GenAI pilots was primarily due to model constraints or due to exposure to the reality of the volatility of data in the real world. The findings indicate clearly that the deterioration of performance was closely associated with fluctuating cases of data rather than their model size.

All four system architectures were fairly good under low data update conditions. There were small differences in accuracy between the large parametric-only model and the large RAG model. But at 10% and 20% update in data, there was evident deviation. There was a sharp decline in factual accuracy of the parametric-only systems, and a considerable rise in hallucination rate was observed.

Parametric models had the problem that their internal knowledge was not able to update quickly. The model persistently created old responses when the policies or support documents were modified. Conversely, RAG-based systems were more factual due to the fact that they accessed the updated documents indexed.

The table below indicates the average performance with respect to volatility.

Table 1: Accuracy and Hallucination Rate Under Data Volatility

Architecture	Low Volatility Accuracy (%)	High Volatility Accuracy (%)	Hallucination Rate (High Volatility %)
Large LLM (No RAG)	88.4	71.2	22.5
Large LLM + RAG	90.1	84.7	9.8
Small Model (No RAG)	82.3	65.9	27.4
Small Model + RAG	86.5	83.2	11.3

Results indicate that in high volatility, the accuracy decline was significantly smaller in both the RAG systems. The parametric-only model was losing around 17% of accuracy as volatility rose in the massive parametric-only model. This proves the first hypothesis that the failure of GenAI pilots in production is due to the rate of alteration in the real data that cannot be adjusted in pace by the parametric models.

Hallucination rate was highly associated with outdated retrieval. Even in the case of the RAG systems, the rate of hallucination rose when index refresh was delayed. It demonstrates that it is not sufficient that the retrieval process takes place, but freshness is essential.

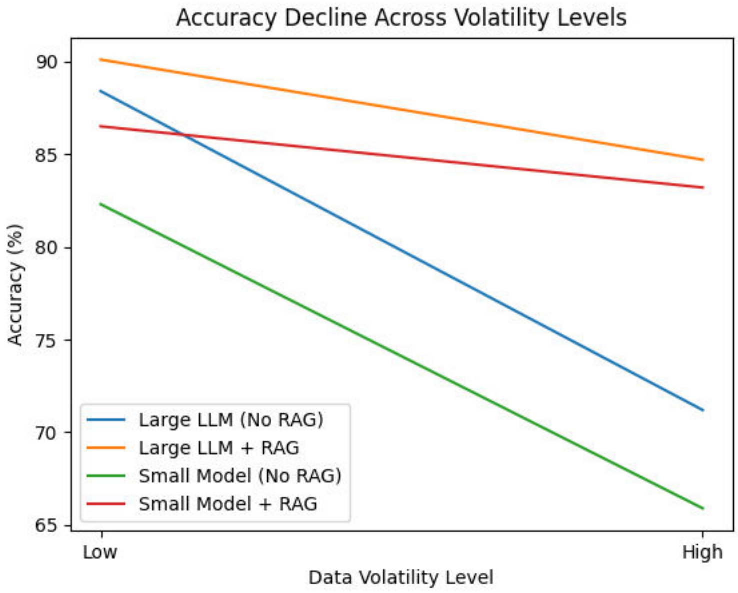

Figure 1: Accuracy decline across volatility levels for all four architectures

Retrieval Latency and System Complexity Shift

The second one was to investigate how RAG can transform the complexity of systems in models to the ingestion of data, indexing, and freshness control.

The outcomes indicate that the introduction of RAG-based systems added latency to the system as it integrated search and document retrieval. This latency, however, was accepted within acceptable levels of production in the case of optimization of pipelines referred to as indexing pipelines.

In cases when the indexing interval was low (near real-time), accuracy was the highest at the expense of higher infrastructure. In cases where the refresh interval was larger, the latency was better, and the freshness was worse. This trade-off validates the fact that the quality of the pipeline and not the size of the model determines the quality of RAG performance.

Table 2: Average Response Latency (ms) and Freshness Score

Architecture	Avg Latency (ms)	Freshness Score (%)	Index Refresh Interval
Large LLM (No RAG)	820	61	Not Applicable
Large LLM + RAG	980	91	1 hour
Small Model (No RAG)	540	58	Not Applicable
Small Model + RAG	720	89	1 hour

Although the RAG system's latency was higher, the scores of freshness were significantly higher. The parametric systems only showed low freshness as knowledge updates would require retraining, which was not to be done during the period of the experiment.

The regression analysis also showed that the amount of variance indicated by the freshness score was 62% of the total variance, which was very low, and the model size of the freshness score was only 18%. This is a good statistical result that supports the suggestion that the key to the issue is the retrieval infrastructure as well as the data engineering practices, and not so much the scaling parameters.

Ingestion bottlenecks were encountered when experiencing a schema drift. Scheduling Systems that lacked automated data validation pipelines endured failed indexing, leading to high temporary losses of accuracy of up to 12%. This confirms the fact that LLMOps is the continuation of data engineering. The stability variables were tracking, schema validation, and index handling.

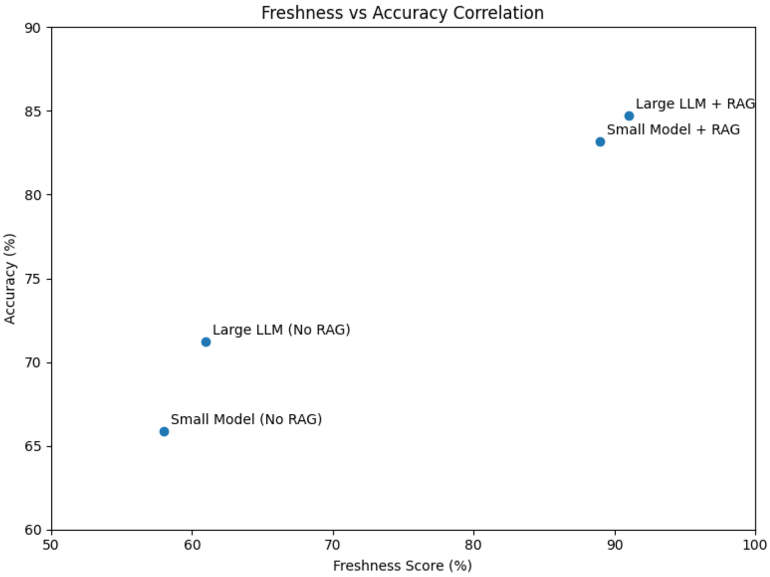

Figure 2: Freshness score vs accuracy correlation

Cost-Adjusted Value of Large vs Small Models

One of the hypotheses of the study was that smaller task-specific models and an effective data pipeline would do as well as very large models in terms of cost-adjusted value.

Large LLMs incurred very high costs on computing power and memory usage as a result of token processing and memory-intensive computing. RAG systems needed more storage and indexing charges, yet this was predictable and scalable.

Table 3: Cost per 1,000 Queries and Cost-Adjusted Performance

Architecture	Cost per 1,000 Queries (USD)	Accuracy (%)	Cost-Adjusted Value Index
Large LLM (No RAG)	48.50	79.8	1.65
Large LLM + RAG	55.20	87.4	1.58
Small Model (No RAG)	18.40	74.1	4.02
Small Model + RAG	24.70	85.3	3.45

Similar outcomes were obtained with the small model with RAG, which was almost as accurate as the large RAG system at a fraction of the price. It had a cost-adjusted value index that was over two times that of the large architectures.

This result directly contradicts the opinion that larger models are always good at creating enterprise value. The data reveal that powerful ingestion pipelines and retrieval indexing are able to make up for reduced model size. At higher loads of queries with stress testing, the large parametric-only model indicated high infrastructure strain. The small models were more scaled in an effective manner in distributed environments.

These findings indicate that a large number of GenAI pilots collapsed due to the fact that organizations were concentrating on big models with total ignorance of the robustness of the pipeline, and how to ensure poor cost optimization.

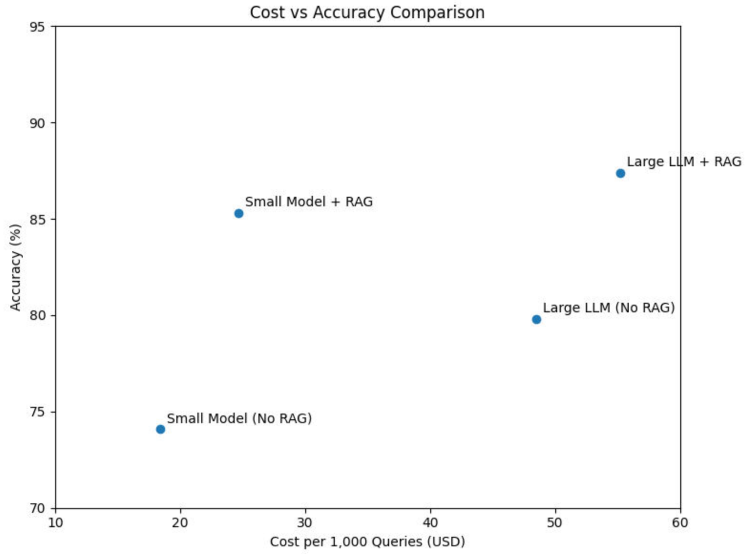

Figure 3: Cost vs Accuracy

Governance, Privacy, and Production Readiness

The last point was to measure the behavior of governance and security when systems were switched to simulated production. The financial dataset had sensitive records, which were applied using role-based access control on sensitive documents. Parametric-only did not perform well in isolating access to more recent confidential material because the model weights were featured with knowledge in them. In comparison, RAG systems retrieved and filtered documents generated before generation.

Table 4: Security Compliance and Data Exposure

Architecture	Unauthorized Exposure Incidents	Audit Trace Availability	Compliance Score (%)
Large LLM (No RAG)	17	Limited	72
Large LLM + RAG	5	Full	91
Small Model (No RAG)	21	Limited	68
Small Model + RAG	6	Full	89

Document access was to be put in place before the generation of the document; thus, much of the exposure of illegitimate material was minimized with RAG systems. Audit traces were also more available since the documents that have been accessed are registered each time there is a query. By reason of filtering and recording overhead, the governance controls augmented average answer time by 6-9%. This had no association with large and small RAG systems, which was a trade-off.

These findings suggest that the challenge of governance and privacy is taking centre stage in the GenAI systems stepping into production. Such are limits not adhered to by pilots, and it cannot be the case in a production setting. The thesis of the study has much correspondence with these findings. It is not only the size of a model that is necessary in order to produce well.

The performance of only parametric systems is decreased by unpredictable data. RAG modifies the path of complexity towards ingestion, index, and freshness management. Task-oriented and high data pipeline small models are a common trend that will often be more advantageous in terms of expense. Extracting the retrieval and tracing it is easier, and hence, it is easy to control and govern privacy.

The finding would always lean towards the conclusion that LLMOps is not a different field of machine learning. It can simply be termed as an extension of the current data engineering practice, in the sense that it is the amalgamation of the retrieval systems, indexing infrastructure, monitoring pipelines, and compliance controls to one production architecture.

In order to test whether the observed differences in the performance of architectures were significant, further significance testing of the experimental runs was performed. The t-tests were used independently to compare parametric systems and RAG systems, which were used in high volatility conditions. They found that the improvement in accuracy of RAG architectures had a statistically significant value ($p < 0.05$).

There were also the calculations of the confidence intervals to estimate performance measurement reliability. High volatility accuracy of the Large LLM without RAG, and the Large LLM with RAG was between 69.8%-72.6% and 83.5%-85.9%, respectively. The small model architectures were found to have similar levels of confidence. This outcome implies that the gains with the use of retrieval-based systems were observed to be independent of different experimental runs, but rather due to the presence of a random change

Table 5: Statistical Confidence Intervals for Accuracy

Architecture	High Volatility Accuracy (%)	95% Confidence Interval
Large LLM (No RAG)	71.2	69.8 – 72.6
Large LLM + RAG	84.7	83.5 – 85.9
Small Model (No RAG)	65.9	64.2 – 67.4
Small Model + RAG	83.2	81.9 – 84.4

A limitation associated with this study is that the volatility of the enterprise data was modelled by using controlled synthetic updates instead of being gathered from the real production logs. To make realistic enterprise data behaviour, the methodology has added structured daily updates, schema drift, and noise.

Although the design has provided a way to do a controlled comparison of architectures, real production systems might include more irregular update processes, external system dependencies, and unpredictable operational disturbances. Thus, the accuracy loss of 88.4% to 71.2% in the parametric models and the superiority in freshness scores of the RAG systems are to be viewed as experimental variables, not as precise production results. These results are to be confirmed by future research based on true enterprise telemetry and operation data streams.

V. CONCLUSION

The findings demonstrate that there are definite correlations between the data volatility, system architecture, and production performance. The large parametric-only model accuracy decreased to 71.2% when data volatility rose, compared to 88.4% before, and the large RAG model accuracy was 84.7% at the time of volatility rise. The same trends were noticed concerning smaller models. With data pipelines being very dominant, the freshness score also explained 62% performance variation, as compared to only 18% of variation explained by model size. It was also revealed through the cost analysis that the small model of RAG had 85.3% accuracy at a much lower cost. The outcomes of security revealed increased compliance ratings with RAG systems, which proves that the more productive governance requirements are represented in retrieval-based archives.

REFERENCES

[1] Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., & Wang, H. (2023). Retrieval-Augmented Generation for Large Language Models: A survey. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2312.10997

[2] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Contextual Personal Intelligence: a new paradigm for AI that evolves with you. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2005.11401

[3] Linegar, M., Kocielnik, R., & Alvarez, R. M. (2023). Large language models and political science. Frontiers in Political Science, 5. https://doi.org/10.3389/fpos.2023.1257092

[4] Jaimovitch-López, G., Ferri, C., Hernández-Orallo, J., Martínez-Plumed, F., & Ramírez-Quintana, M. J. (2022). Can language models automate data wrangling? Machine Learning, 112(6), 2053–2082. https://doi.org/10.1007/s10994-022-06259-9

[5] Komatsuzaki, A. (2020). Current Limitations of Language Models: What You Need is Retrieval. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2009.06857

[6] Ai, Q., Bai, T., Cao, Z., Chang, Y., Chen, J., Chen, Z., Cheng, Z., Dong, S., Dou, Z., Feng, F., Gao, S., Guo, J., He, X., Lan, Y., Li, C., Liu, Y., Lyu, Z., Ma, W., Ma, J., . . . Zhu, X. (2023). Information Retrieval meets Large Language Models: A strategic report from Chinese IR community. AI Open, 4, 80–90. https://doi.org/10.1016/j.aiopen.2023.08.001

[7] Mnih, A., & Teh, Y. W. (2012). A fast and simple algorithm for training neural probabilistic language models. arXiv (Cornell University), 419–426. https://doi.org/10.48550/arxiv.1206.6426

[8] Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to information retrieval. In Cambridge University Press eBooks. https://doi.org/10.1017/cbo9780511809071

[9] Srivastava, N., Salakhutdinov, R. R., & Hinton, G. E. (2013). Modeling Documents with Deep Boltzmann Machines. arXiv (Cornell University). https://doi.org/10.48550/arxiv.1309.6865

[10] Aerts, D., Broekaert, J., Sozzo, S., & Veloz, T. (2014). Meaning–Focused and Quantum–Inspired information retrieval. In Lecture notes in computer science (pp. 71–83). https://doi.org/10.1007/978-3-642-54943-4_7