POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Impact of Document Quality on Retrieval-Augmented LLMs

Laurea Magistrale in Computer Science Engineering - Ingegneria Informatica

Authors: Arianna Gottardi, Elisa Gallo

Advisor: Prof. Cinzia Cappiello

Co-advisor: Camilla Sancricca

Academic year: 2024-2025

1. Introduction

Large Language Models (LLMs) have revolutionized how we interact with textual information, demonstrating extraordinary capabilities in complex tasks such as question answering, summarization, and reasoning. However, the Retrieval-Augmented Generation (RAG) paradigm represents a significant evolution, as it allows models to dynamically access external knowledge bases to improve the accuracy and relevance of responses. This approach reduces limitations such as out-of-date information and hallucinations but introduces a crucial dependency: the quality of retrieved documents directly influences the reliability of generated responses. This dependency reflects a fundamental principle of data analysis: data quality is key to accurate and reliable decision-making [2]. Just as organizations rely on high-quality data for decisions, RAG systems depend on high-quality documents for reliable responses. However, while traditional quality dimensions are well defined for structured data, they are only partially applicable to unstructured text documents. Despite recent research analyzing text quality in specific contexts (e.g., education[5], healthcare[1]), few studies have investigated how document quality affects LLM performance in production environments[4]—where knowledge bases inevitably contain errors, redundancies, or contradictions that make even state-of-the-art neural models brittle[3]. Understanding the impact of such imperfections on RAG system performance is essential to ensure reliability, robustness, and trust in AI-based systems. This research addresses this gap by developing a Quality Model to objectively measure document quality and quantify its impact on LLM performance in RAG-based Question Answering systems, providing both theoretical foundations and practical guidelines for improving system reliability.

2. Methodology

This research addresses two complementary objectives: (1) defining a multi-dimensional Data Quality model for textual documents, and (2) systematically evaluating state-of-the-art LLMs sensitivity to controlled quality degradations across these dimensions.

DQ Model for documents The framework characterizes document quality through six dimensions, each measured using specific metrics: *Grammar Correctness* quantifies spelling and morphological errors through Word Error Rate

Figure 1: LLM Evaluation Pipeline

(WER) and Character Error Rate (CER): edit-distance metrics that measure substitutions, insertions, and deletions at word and character level with respect to a reference text.

Syntactic Correctness evaluates sentence-level structural integrity using the same WER/CER metrics but applied to assess word order conventions, agreement rules, and syntactic organization rather than lexical correctness.

Beyond grammatical and syntactic correctness, document quality depends on *Readability*, which represents how easily the content can be understood. It measures cognitive accessibility through three complementary formulas: Flesch Reading Ease Score (FRES: 0-100 scale, higher = easier), Flesch-Kincaid Grade Level (FKGL: years of education required), and Gunning Fog Index (GFI: emphasizing polysyllabic word complexity).

Factuality & Information Density measures the amount of relevant information and factual content through three mechanisms: (1) entity-based density (number of named entities divided by the total word count), (2) factual consistency, and (3) semantic relevance (semantic similarity to capture how well the content meets the information need).

Sentence Flow evaluates narrative progression through dual assessment: (1) Order Preservation via Kendall's Tau (quantifies the number of sentences in a different position compared to a reference), and (2) Semantic Flow (consecutive sentence similarity capturing discourse continuity).

Coherence quantifies local semantic consistency through semantic similarity between adjacent sentences, measuring whether consecutive discourse segments maintain topical relatedness independently of structural ordering.

LLM Evaluation Pipeline After defining the dimensions of a "good document", we built a four-step pipeline to investigate LLM robustness to inputs degradation along these dimensions.

1- Error injection: Starting with clean documents, controlled degradations are injected in line with the six dimensions of the DQ model. *Grammar* perturbations included minor spelling and typographical errors, while *Syntax* errors involved structural inconsistencies and word arrangement issues. *Readability* was reduced through increased lexical complexity or unnecessary textual elaboration, and *Factuality & Information density* was altered by introducing inaccuracies or reducing informational content. *Coherence* was disrupted through changes in logical or semantic connections between ideas, while *Sentence Flow* was modified by altering the sequential organization of content.

2- Prompting LLMs: The documents (clean and degraded) were used in Question Answering experiments, forcing the models to rely solely on the context provided without drawing on parametric knowledge.

3- Response Evaluation: Once the model outputs were collected, six metrics were used to analyze performance. Each metric addresses the following questions:

- **Accuracy:** How precise and on-topic is the response?
- **Completeness:** Is all the information that should be there present?
- **Key Concept Coverage:** Is there a key concept needed to answer the question?
- **Conciseness:** How concise or verbose is the response?
- **NumReplicateErrors:** How much noise did the model replicate from the dirty context?
- **NumAdditions:** How many entities or numbers did the model introduce (thus using basic knowledge)?

These metrics can be grouped by their objective: the first three are semantic metrics used to evaluate the qualitative aspects of the answer. Conciseness is a structural metric, while NumReplicateErrors and NumAddition are quantitative, error-oriented metrics.

4- Analysis of Results: The retrieved data was analyzed from three different perspectives: by DQ Dimension (identifying the most critical errors), Cross-Metric (correlations between evaluation metrics), and Input-Output (relationship between input degradation and response quality, identifying reliability thresholds).

3. Implementation

We now move from the conceptual and methodological level to the practical one, describing the

implementation of the two core components: the DQ Model and the Validation Pipeline.

Data Quality Model Implementation The metrics of the Data Quality Model were implemented using standard NLP tools. For the lexical quality dimensions (*Grammar*, *Syntax*, and *Readability*), existing text evaluation libraries were employed, as they already provided the required metrics. For the semantic quality dimensions (*Coherence*, *Sentence Flow*, and *Factuality and Information Density*), additional modules were developed to estimate factual consistency through Natural Language Inference (NLI) and semantic similarity, computed using transformer-based embeddings, with SBERT[1] used as the primary model.

LLM Evaluation Pipeline Implementation The implementation of the pipeline began with keyword extraction to identify which words, if modified, could cause the model difficulty in answering questions. Rather than random selection, we focused on words expressing core facts, hypothesizing this would yield more meaningful targets. To do so, GPT-4 is used to extract factual triples in the form [Subject, Predicate, Object], and finally filtered to keep only nouns, proper nouns, verbs, adjectives, and named entities, up to a maximum of 45% of manipulable words. Once the keywords to be manipulated were identified, the documents were systematically degraded. Only manipulations that were both humanly feasible and significantly affected model performance were retained, leading to the exclusion of some quality dimensions during preliminary evaluation. The final 10 manipulations tested are:

GR	Letter swaps, keyboard typos, character removal, random typos
SYNT	Verb tense errors, word shuffling
READ	Substitution with complex synonym, insertion of definitions, double negation
INFDENS	Content dilution with filler text

After generating the corrupted contexts, LLMs were queried via API using strict prompts. Their outputs were evaluated with the six custom metrics, previously introduced, and the same procedure on clean context provided the baseline. The following provides the implementation idea of these metrics. **Accuracy** was measured as the median of semantic and lexical similarity, with a dynamic penalty for error replication. **Completeness**, on the other hand, was computed as the median of (1) the cosine similarity between the sentences in the response and the corresponding phrases in the reference answer, and (2) a recall score of factual entities. **Key Concept Coverage (KCC)** was computed through a three-stage process: (1) generating an "essential" response capturing the key concepts using google/flan-t5-large[2]; (2) extracting the relevant tokens from this essential response; and (3) calculating the percentage of these tokens that appear exactly in the model's final answer. Additionally, **Conciseness** was calculated as the ratio between the length of the response and that of the reference. Finally, two metrics tracked error propagation: **NumReplicateErrors** counted the number of injected errors literally replicated in responses, while **NumAdditions** identified new factual entities added that were not present in the original context. These metrics provided the quantitative foundation for the final analytical phase, which employed three complementary approaches to examine the data: Dimension analysis, performed by calculating medians and percentage differences relative to the baseline; Cross-metric analysis was performed using correlation-based methods to quantify relationships between the evaluation metrics, while input-output analysis employed statistical and computational techniques to map how variations in context quality affected response reliability across different performance ranges.

4. Experimental Validation

Preliminary Validation: BoolQ[3] (16,000 True/False questions with Wikipedia passages) was used for initial pipeline validation through exact-match scoring. Three limitations emerged: (1) high robustness, showing minimal degradation and thus a limited observable effect of manipulations; (2) chance-level ambiguity, since binary answers allow 50% guessing;

[1] https://huggingface.co/sentence-transformers

[2] https://huggingface.co/google/flan-t5-large

[3] https://github.com/google-research-datasets/boolean-questions

(3) poor granularity, preventing nuanced quality evaluation. These issues motivated the switch to CLAPNQ.

Full-Scale Validation: CLAPNQ[4] (4,946 questions with cohesive long-form answers from Wikipedia) overcomes BoolQ limitations by providing open-ended responses suitable for fine-grained semantic evaluation. We selected 1,954 answerable training questions to compare GPT-4o-mini[5] and Gemini 2.5 Flash[6].

Experimental protocol: For each passage, the experimental protocol consisted of: (1) baseline evaluation with clean context; (2) systematic manipulation across 10 types, affecting 50% and 100% of the targeted keywords or sentences; (3) four runs per configuration; (4) context-constrained prompting of both the clean and manipulated versions; Figure 2 provides a clear representation of the approach.

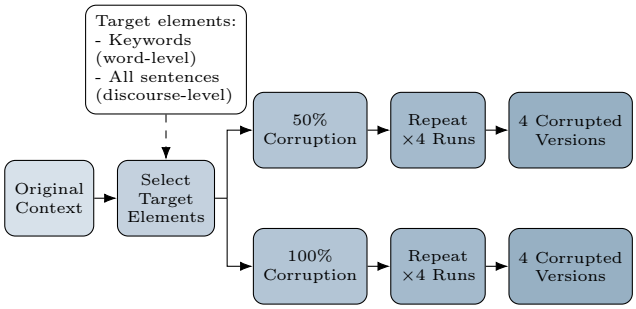

Figure 2: Manipulation Protocol Representation

Experimental scale: 1,954 passages × (80 manipulated + 1 clean) resulted in 158,274 evaluations per model. Median context length was 169 words (7 sentences). Corruption affected 15% of text at 50% intensity and 30% at 100%. Identical manipulations across models enabled direct comparison.

5. Results

By Dimensions The introduction of controlled degradations revealed different error management strategies (Fig. 3, 4) .

Grammar: GPT is more vulnerable to manipulations that make words unrecognizable (e.g., `RAN_TYPO, REMOVE`). Its primary failure strategy is omission: it discards concepts it cannot decode, causing a collapse of the KCC.

[4] `https://github.com/primeqa/clapnq`
[5] `https://openai.com/it-IT/index/gpt-4/`
[6] `https://deepmind.google/models/gemini/`

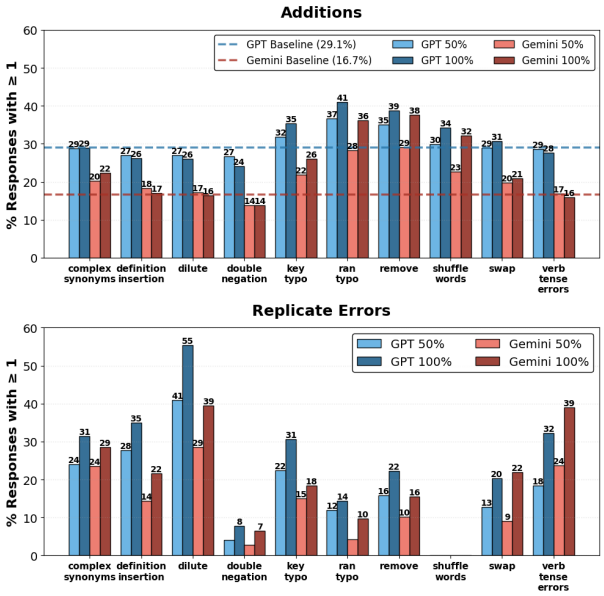

Figure 3: Percentage of responses with non-zero additions and error replications

Gemini is more vulnerable to manipulations that create plausible pseudo-words (e.g., `SWAP`). Its literal copying strategy (evidenced by a collapse in semantic similarity but an increase in lexical similarity) leads to binary failures: it either handles the error perfectly or replicates it catastrophically.

Syntax: Both models showed a shared and "silent" vulnerability: they replicate verb conjugation errors in 30-40% of cases, even when Accuracy and KCC remain high. When faced with `SHUFFLE_WORDS`, Gemini shows a "compensatory explosion," introducing significantly more additional information (hallucinations) compared to its baseline than GPT.

Readability: Both models revealed a dual vulnerability: semantic complexity (`COMPLEX_SYNONYMS` and `DEFINITION_INSERTION`) degrades accuracy, while logical complexity (`DOUBLE_NEGATION`) degrades completeness. When faced with logical complexity, GPT attempts a reformulation (sacrificing details), while Gemini exhibits unstable and "all-or-nothing" behavior, sometimes failing to capture the key concept.

Information Density: The models failed to filter contextual "noise" (dilution) in different ways. GPT showed a "replication-dominated" failure, passively copying noise (over 50% of NumReplicateErrors) and degrading Accuracy. In contrast, for Gemini, manipulation improved its Completeness and KCC scores, as it incor-

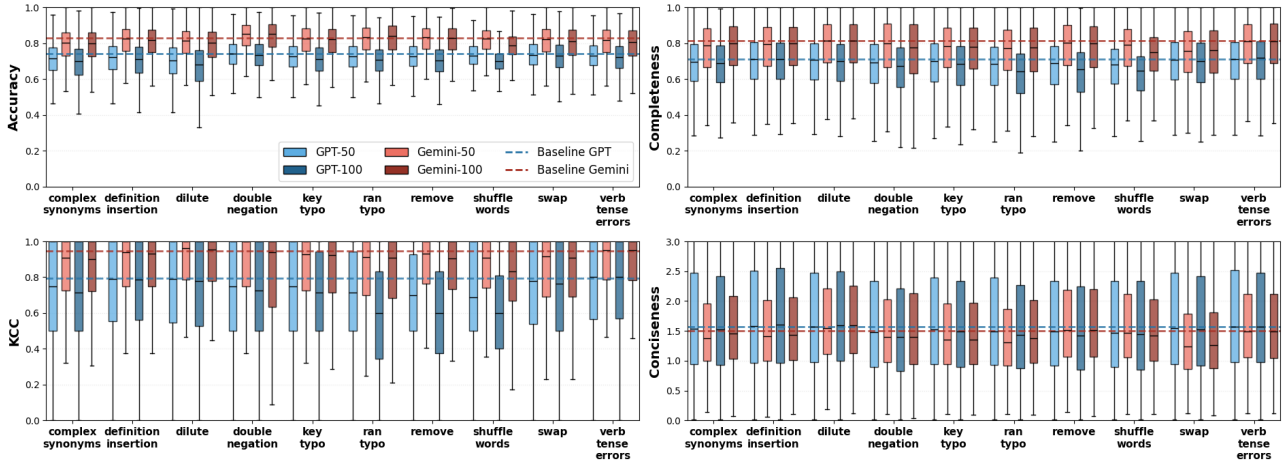

Figure 4: Median values of semantic scores and conciseness

porated information into its responses that was contextually related to the initial context.

Cross-Metric After evaluating each quality dimension individually, we now take a higher-level view to examine how the evaluation metrics interact across different conditions. In the following analysis, we focus on the most revealing relationships for both models. GPT exhibits strong metric correlations: Accuracy-Completeness-Conciseness form coherent 3D regression plane, revealing adaptive trade-offs (clean input → concise answers; ambiguous input → expanded responses; severe degradation → simplified outputs). Both Gemini and GPT

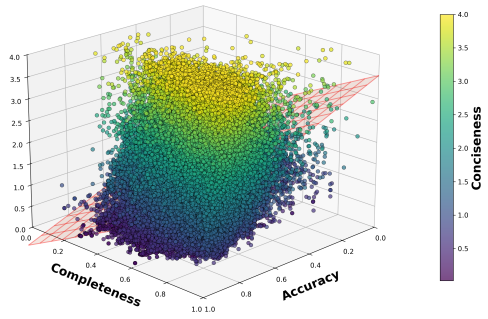

Figure 5: GPT: 3D regression plane

show a negative correlation between Accuracy and quantitative metrics, meaning replicas and additions reduce overall topic precision. Regarding Completeness, GPT has a positive correlation with artificially added content, while Gemini shows a negative correlation, meaning the added information does not cover the required concepts. In GPT, Conciseness positively correlates with added words, whereas Gemini shows a largely flat relationship. These patterns confirm the architectural divergence of the two models: GPT as a dynamic and compensatory system; Gemini shows stable performance but limited adaptability, switching between context reproduction and internal knowledge recall.

Data Quality Guidelines Having examined all experimental results in details, we then move to the final step: drawing practical conclusions. This stage translates the statistical evidence into actionable guidance for real-world RAG workflows. This analysis is guided by two questions. The first is:

(1) *What level of input noise still guarantees reliable model responses?* We derive quality thresholds from the DQ Model to identify acceptable degradation levels. While general trends are clear—GPT degrades gradually, Gemini behaves non-linearly—precise failure points are often unpredictable. Some dimensions show unclear behavior, and metrics can diverge. Table 1 summarizes clear thresholds (green), ambiguous patterns (yellow), and unpredictable failures (red).

Dim.		Semantic	Additions	Replicas
GR	GPT	WER<0.2 CER<0.08	WER<0.3 CER<0.08	WER<0.25 CER<0.05
	Gemini	Stable	Unpred.	WER<0.25 CER<0.05
SYNT	GPT	WER<0.3 CER<0.15	WER<0.3 CER<0.25	WER<0.15 CER<0.05
	Gemini	WER<0.4 CER<0.2	Unpred.	WER<0.15 WER<0.05
READ	GPT	Unpred.	Additions stable at FRES∼40	FRES>30 FKGL/GFI<20
	Gemini	Unpred.	Additions stable at FRES∼40	FRES>30 FKGL/GFI<20
INFDENS	GPT	Rel.>0.5	Additions stable at Rel.∼0.5	Unpred.
	Gemini	Stable	Additions stable at Rel.∼0.5	Unpred.

Table 1: Thresholds for stable output performance.

(2) *Given a target application and contextual*

conditions, which model should be used—GPT-4o-mini or Gemini 2.5 Flash? We propose a decision framework linking task requirements to each model's strengths (Table 2). GPT offers stable, predictable behavior and gradual degradation, ideal when strict context adherence is needed. Gemini excels at extracting main content but is less able to reformulate and more prone to using information outside the context. There is no single best model; the choice depends on error tolerance, input quality, and available validation resources.

Objective	Model
Creativity and Reformulation	GPT
Deep Reasoning	GPT
Grounded Responses (context-based)	GPT
Concise and Rephrased Answers	GPT
Stability and Predictability	GPT
Robust fact Extraction	Gemini
Surface-level summarization	Gemini
Internal Knowledge Use (assuming tolerance for potential inaccuracies)	Gemini

Table 2: Best-performing LLMs for each objective

6. Conclusions and Future Works

In this thesis, we developed and tested a framework to assess how robust LLMs are to errors or quality issues in the documents they process.
The research delivered four primary contributions: (1) *Multi-dimensional DQ Model* characterizing textual quality ; (2) *Realistic degradation methodology* implementing 10 human-feasible manipulations with observable impact, tested across 1,954 CLAPNQ passages at 50%/100% intensity; (3) *Evaluation framework* combining semantic quality (Accuracy, Completeness, KCC), structural appropriateness (Conciseness), and error behavior (NumAddition, NumReplicateErrors) to reveal not only *when* but *how* models fail; (4) *Practical guidelines* specifying input quality thresholds and model selection criteria for production RAG deployment.
Key findings reveal dimension-specific vulnerabilities and model-specific behavioral profiles, enabling us to extract practical guidelines.

Limitations and Future Directions During our work, we faced some limitations: (1) Gemini often draws on prior knowledge despite instructions, complicating assessment of manipulations effects; few-shot prompting could be a possible solution. (2) Short documents (≈170 words) may underestimate manipulation effects; longer texts and multi-document scenarios deserve investigation. (3) Semantic similarity metrics can penalize concise correct answers and reward plausible errors, highlighting the need for LLM-based factuality checks or hybrid human validation. Future directions include testing irrelevant or unanswerable contexts, evaluating documents with multiple errors across different manipulations, extending analysis to other LLMs (Claude, Llama, Mistral), and developing more robust factuality metrics. Exploring these areas can clarify how document quality impacts RAG systems and LLM performance, guiding model choice and emphasizing quality over quantity of input data.

References

[1] C. H. Basch, J. Mohlman, G. C. Hillyer, et al. Public health communication in time of crisis: Readability of on-line covid-19 information. *Disaster Medicine and Public Health Preparedness*, 14(5):635–637, 2020.

[2] C. Batini and M. Scannapieca. *Data Quality: Concepts, Methodologies and Techniques*. Data-Centric Systems and Applications. Springer, Berlin New York, 2006.

[3] Yonatan Belinkov and Yonatan Bisk. Synthetic and natural noise both break neural machine translation, 2018.

[4] Giovanni Monea, Maxime Peyrard, Martin Josifoski, Vishrav Chaudhary, Jason Eisner, Emre Kiciman, Hamid Palangi, Barun Patra, and Robert West. A glitch in the matrix? locating and detecting language model grounding with fakepedia. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, August 2024.

[5] S. Vajjala and I. Lučić. Onestopenglish corpus: A new corpus for automatic readability assessment and text simplification. In *Proceedings of the 13th Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*, pages 297–304, New Orleans, LA, 2018. Association for Computational Linguistics.