

# 000 RARE: RETRIEVAL-AWARE ROBUSTNESS EVALUATION 001 FOR RETRIEVAL-AUGMENTED GENERATION SYSTEMS 002

003 **Anonymous authors**

004 Paper under double-blind review

## 005 ABSTRACT

006 Retrieval-Augmented Generation (RAG) enhances recency and factuality in answers. How-  
007 ever, existing evaluations rarely test how well these systems cope with real-world noise,  
008 conflicting between internal and external retrieved contexts, or fast-changing facts. We  
009 introduce **Retrieval-Aware Robustness Evaluation (RARE)**, a unified framework and large-  
010 scale benchmark that jointly stress-tests query and document perturbations over dynamic,  
011 time-sensitive corpora. One of the central features of RARE is a knowledge-graph-driven  
012 synthesis pipeline (RARE-Get) that automatically extracts single and multi-hop relations  
013 from the customized corpus and generates multi-level question sets without manual in-  
014 tervention. Leveraging this pipeline, we construct a dataset (RARE-Set) spanning 527  
015 expert-level time-sensitive finance, economics, and policy documents and 48295 questions  
016 whose distribution evolves as the underlying sources change. To quantify resilience, we  
017 formalize retrieval-conditioned robustness metrics (RARE-Met) that capture a model’s  
018 ability to remain correct or recover when queries, documents, or real-world retrieval results  
019 are systematically altered. Our findings reveal that RAG systems are unexpectedly sensitive  
020 to perturbations. Moreover, they consistently demonstrate lower robustness on multi-hop  
021 queries compared to single-hop queries across all domains.

## 022 1 INTRODUCTION

023 Retrieval-Augmented Generation (RAG) significantly enhances Large Language Models (LLM) by integrating  
024 external knowledge sources, allowing the generation of accurate and contextually rich responses (Gao et al.,  
025 2024). However, the robustness of RAG systems remains inadequately evaluated. In addition, current  
026 benchmarks predominantly rely on static, time-invariant datasets with general-knowledge or common-sense  
027 queries. Such benchmarks inadvertently favor models that rely on memorization rather than genuine retrieval  
028 and synthesis of novel, timely information (Xu et al., 2024). Consequently, existing assessments yield overly  
029 optimistic performance measures, overlooking critical real-world scenarios involving dynamic, specialized,  
030 and complex information.

031 An ideal synthesized evaluation dataset generation pipeline for RAG must address several critical dimensions  
032 simultaneously, emphasizing **dynamics**, **query complexity**, and **content specialization**. Dynamics is  
033 crucial to reflect real-world scenarios where information evolves rapidly (Meem et al., 2024; Jang et al.,  
034 2022), particularly in domains such as finance (Shen & Kurshan, 2023). Such time-sensitive data sets  
035 prevent contamination of memorized responses and require continuous adaptation by RAG systems. Query  
036 complexity, especially multi-hop scenarios that require complex reasoning and integration across multiple  
037 retrieved documents (Yang et al., 2018; Geva et al., 2021). Most existing multi-hop datasets require substantial  
038 human efforts, which makes it impossible to curate large-scale extensive datasets. Therefore, automation is  
039 essential and advanced techniques such as Knowledge Graphs (KGs) (Schneider et al., 2022) can be used.  
040 Moreover, with widespread integration into real-world applications, benchmarks must emphasize content  
041

047 specialization, including professional and domain-specific contexts that challenge models with intricate  
 048 terminology and nuanced interpretations.  
 049

050 Additionally, most RAG benchmarks has focused on accuracy measurements, with limited attention to  
 051 how these systems perform when faced with noisy or imperfect inputs. In real-world applications, an  
 052 RAG system usually should contend with perturbed queries containing typos, irrelevant information, or  
 053 ambiguous phrasing (Zhang et al., 2025b). Retrieved document may also be noisy, partially relevant, or even  
 054 contradictory (Chen et al., 2023). A truly robust RAG system should maintain robust performance despite  
 055 these challenges.

056 In this paper, we introduce a comprehensive **Retrieval-Aware Robustness Evaluation (RARE)** framework. It  
 057 includes: **RARE-Get**: a novel dynamic synthesis pipeline that automatically constructs time-sensitive RAG  
 058 evaluation data through knowledge graph triplet extraction and traversal techniques, enabling the creation of  
 059 single-hop and multi-hop tuples (question, answer, ground truth chunks) at various complexity levels without  
 060 manual curation. **RARE-Set**: a large-scale benchmark comprising 527 specialized documents and 48295  
 061 queries across financial, economics, and policy domains - sectors where information accuracy and timeliness  
 062 are particularly critical yet underrepresented in existing benchmarks. Unlike previous datasets dominated  
 063 by general knowledge questions, our benchmark exclusively focuses on "rare" datasets: domain-specific,  
 064 technical queries that require advanced information synthesis. **RARE-Met**: a comprehensive robustness  
 065 evaluation metric for measuring RAG system performance under perturbations to queries, documents, and  
 066 simulated real-world retrieval results, providing diagnostic insights into current system limitations. Our  
 067 dataset features diverse query patterns generated through knowledge graph traversal, including single-hop,  
 068 multi-hop chained, star-shaped, and inverted-star-shaped, with systematic perturbations at both surface and  
 semantic levels to comprehensively assess robustness under realistic conditions.

069 Our evaluation reveals that RAG systems are still fragile under some perturbations. Robustness scores do  
 070 not always scale strictly with model size - some mid-sized generators outperform several larger counterparts.  
 071 Also, the robustness of RAG systems across different domains is different, and multi-hop queries prove  
 072 less robust than single-hop queries. All of these indicate the importance of evaluating and improving the  
 073 robustness of RAG systems.

## 074 2 RELATED WORK

075 **Time-Sensitive Benchmark** Recent temporal-related benchmark initiatives address LLM knowledge out-  
 076 dating through distinct approaches. FreshQA (Vu et al., 2024) tests reasoning over up-to-date knowledge  
 077 with a fixed questions, dynamic answers-updated QA benchmark and evaluation methodology for correctness  
 078 and hallucination detection. PAT-Questions (Meem et al., 2024) introduces a self-updating benchmark for  
 079 present-anchored temporal questions using SPARQL queries over Wikidata to automatically refresh answers.  
 080 RealtimeQA (Kasai et al., 2024) employs a weekly dynamic platform that extracts questions from news  
 081 quizzes, challenging systems to answer questions about current events. Existing benchmarks often exhibit  
 082 limitations such as narrow raw data domains (primarily Wikipedia or news articles), a restricted number of  
 083 test cases due to the reliance on fixed human-generated questions, and a prevalence of queries that can be  
 084 accurately answered by the language model alone—without the need for retrieval—such as general-domain  
 085 fact questions.  
 086

087 **Multi-Hop QA and RAG Benchmark** Early knowledge-intensive benchmarks like Natural Questions  
 088 (Kwiatkowski et al., 2019) and HotpotQA (Yang et al., 2018) established foundations for factual  
 089 question answering but lacked cross-document reasoning and overlapping with popular training dataset. Later  
 090 development such as MuSiQue (Trivedi et al., 2022) and StrategyQA (Geva et al., 2021) advanced multi-hop  
 091 QA capabilities but remained confined to Wikipedia sources. MultiHop-RAG (Tang & Yang, 2024) expanded  
 092 to news domain with 2-4 hop queries but lacks dynamic real-time updates. RAGBench (Friel et al., 2025)  
 093

094  
095 Table 1: Comparison of our proposed dataset with prior benchmarks. Symbols: ✓ = yes/present; ✗ = not  
096 available; "partial" = feature applies to only a subset; "-" = not applicable; MH = Multi Hop question.  
097

Dataset	Year	# QA	Data Sources	Unique	Time-Sens.	MH	Dynamic	Automatic
<b>Time-Sens. Benchmarks</b>								
RealtimeQA	2023	2340	News	✓	✓	✓	✓	partial
FreshQA	2024	600	Search engine	✓	✓	✓	✓	partial
PAT-Questions	2024	6172	Wikipedia	partial	✓	✓	✓	✓
<b>MH &amp; RAG Benchmarks</b>								
Natural Questions	2019	100 k	Wikipedia	✗	✗	✗	✗	✗
HotpotQA	2018	97.9 k	Wikipedia	✓	✗	✓	✗	✗
MuSiQue-Ans	2022	50 k	Wikipedia	✗	✗	✓	✗	partial
StrategyQA	2021	2780	Wikipedia	✓	✗	✓	✗	✗
MultiHop-RAG	2024	2506	News	✓	✓	✓	✗	✓
RAGBench	2024	100 k	Domain-specific	✗	✗	✓	✗	✓
CRAG	2024	4409	Search engine	✗	✓	✓	✗	partial
<b>LLM Robust Benchmarks</b>								
KaRR	2023	-	T-REx (Wikipedia)	partial	✗	✗	✗	partial
QE-RAG	2025	51 k	Wiki + Domain-specific	partial	✗	✓	✗	✓
SURE	2025	-	NQ-open (Wikipedia)	✗	✗	✗	✗	✓
<b>RARE (Ours)</b>	2025	48.3 k	Domain-specific reports	✓	✓	✓	✓	✓

113 introduced evaluation across industry corpora with new faithfulness metrics, with CRAG (Yang et al., 2024)  
114 targets dynamic performance across multiple domains with simulated web and knowledge graph APIs, though  
115 still limited in scale and dynamic renew ability.

116  
117 **LLM & RAG Robustness** Recent frameworks attempt to quantify RAG robustness, usually with various  
118 perturbations. RAGAS (Es et al., 2025) measures factual consistency through automated evaluation without  
119 ground-truth annotations but lacks assessment of query/document perturbations and limited number of  
120 assessment. Cao et al. (2025) analyzed the robustness of the RAG system on linguistic variations and  
121 found that RAG systems are even more sensitive to these variations compared with LLM-only generation.  
122 SURE Yang et al. (2025b) introduced a framework to quantify the sensitivity to semantic-agnostic spurious  
123 features (e.g. format of document) in grounding data, providing a taxonomy of formatting variations that  
124 reveal widespread vulnerabilities. QE-RAG (Zhang et al., 2025b) tests robustness by injecting realistic  
125 query entry errors into QA datasets to evaluate tolerance to input noise, though primarily focused on static,  
126 general-domain tasks without evaluating document-level corruptions. KaRR (Dong et al., 2023) provides a  
127 statistical approach to assess whether an LLM contains reliable factual knowledge by estimating the ration  
128 of generating correct surface text given varying prompts, although its assessment is limited to parametric  
129 knowledge rather than retrieval capabilities. While these approaches advance discrete facts of RAG robustness,  
130 none offer a unified, dynamic evaluation pipeline capable of automatically generating large-scale, temporal  
131 test cases and measuring performance under systematic perturbations to queries, documents, and retrieval  
132 results.

### 133 3 RARE-GET: DYNAMIC RAG BENCHMARK DATASET GENERATION PIPELINE

134  
135 RAG benchmarks should ideally comprise diverse, realistic queries with corresponding golden passages  
136 containing the information needed to answer them correctly. Creating such benchmarks manually demands  
137 extensive human effort and domain expertise, particularly for specialized, multi-hop reasoning scenarios.  
138 In addition, manual-based benchmark cannot consistently create the dynamic and up-to-date datasets. To  
139 address these challenges, we introduce RARE-Get, a fully automated pipeline for constructing complex RAG  
140 benchmarks from unstructured data.

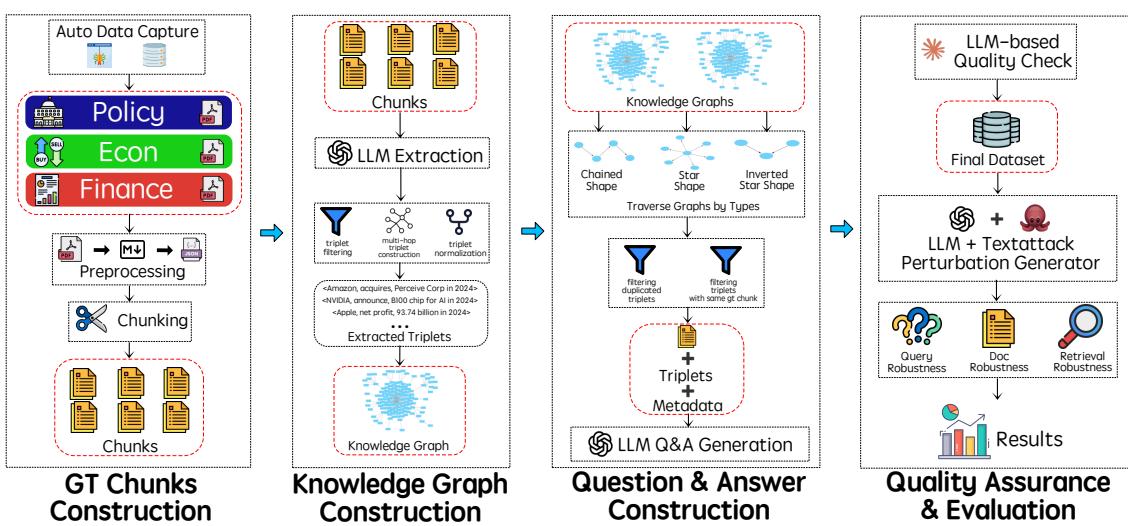


Figure 1: Illustration for the RARE framework. **Red frame**: data that pipeline will generate; **Black frame**: process/movement.

RARE-Get transforms domain-specific documents into comprehensive benchmark datasets through four key stages: (1) Ground Truth Chunks Construction; (2) Knowledge Graph Construction; (3) Question & Answer Construction and (4) Quality Assurance, as illustrated in Figure 1. This approach enables the creation of technical, challenging RAG evaluation datasets that evolve dynamically alongside their source documents, ensuring continued relevance in rapidly changing domains. For time-sensitive, such automatic pipeline also ensures that newest answers with questions will always be updated following by the knowledge graph re-construction or updating process.

### 3.1 CORPUS PREPARATION AND CHUNKING

The pipeline begins by processing domain-specific documents, converting them into manageable chunks suitable for retrieval systems. We carefully segment each document into passages of approximately 600 tokens, striking a balance between informativeness and retrieval efficiency, as well as a real-world retrieval simulation. For tables, we prevent splitting a single table across different chunks. Related information (e.g. table titles, data explanation) will remain in the same chunk. Similarly, for text-only contents, we ensure that no paragraph is divided between chunks. Also, we develop specialized chunking techniques across three distinct domains. Each domain receives tailored processing to enhance information extraction and context retention. Appendix A illustrates the full details for chunking on different domains.

### 3.2 KNOWLEDGE GRAPH EXTRACTION

The cornerstone of the benchmark creation process is systematically transforming chunked documents into structured knowledge representations. For each set of  $n$  consecutive chunks, we employ LLM (GPT-4.1 (OpenAI, 2025)) with carefully designed prompts adapted for different domains. The prompts specify multiple types of multi-hop question patterns with detailed examples, instructing the LLM to extract connected triplets where entities overlap between chunks. In addition, we require the LLM to output the exact source sentence used to extract each triplet; this sentence is later used for validation through a normalized exact-match check to ensure that all extracted relations are fully grounded in the originating chunk, discarding any triplets whose sources are unverifiable.

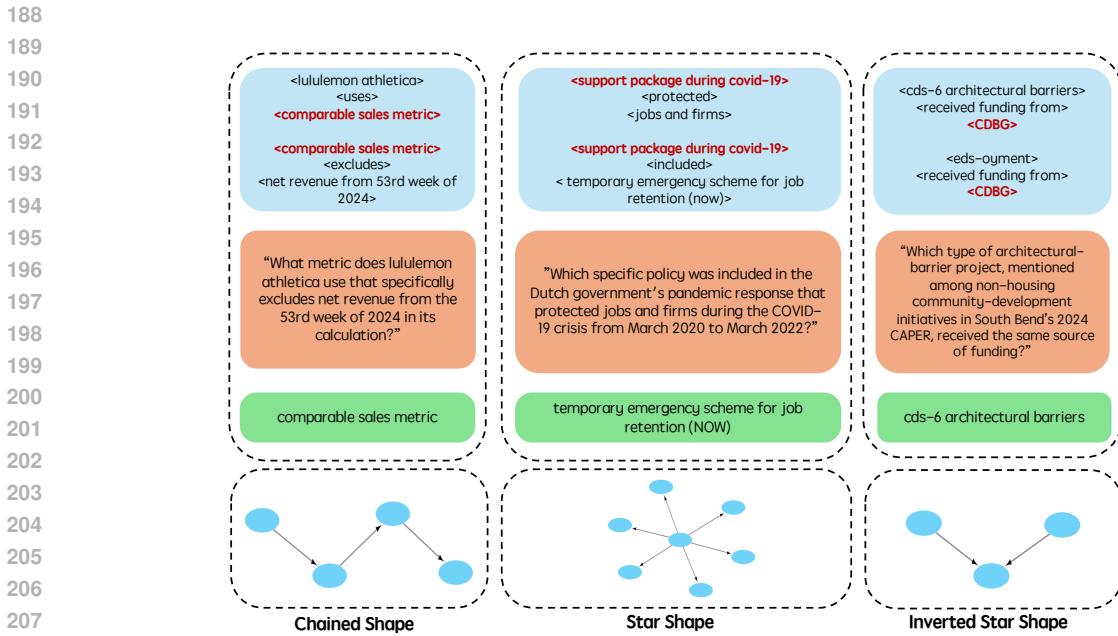


Figure 2: Examples of the multi-hop questions. **Blue**: triplets traversed from KG; **Peach**: generated question; **Green**: generated answer; **Red**: "bridge" entity which connect different triplets together;

To ensure the consistency of the knowledge graph label, we further normalize semantically similar relations (e.g., "manufactures" vs. "produces") using E5-Mistral-7B-Instruct (Wang et al., 2023), one of the leading embedding models according to the MTEB leaderboard (Muennighoff et al., 2023). New relation labels are mapped to existing relation when their cosine similarity exceeds a threshold, which we selected as 0.9 based on human majority voting over sampled relation pairs.

We also perform entity alignment using a text-normalization procedure (lower-casing, punctuation removal, and domain-specific stripping of corporate suffixes such as "Inc." or "Corp." in finance). Finally, after constructing the knowledge graph for each individual document, we merge the per-document graphs into a larger cross-document knowledge graph using NetworkX (Hagberg et al., 2008) to support multi-hop and cross-file question generation. Example prompts used for the extraction of triplets are provided in Appendix F.

### 3.3 QUERY PATTERNS

By traversing the constructed knowledge graph in different strategies, we identify four structural templates, one single-hop and three multi-hop, that produce queries of increasing complexity (multi-hop examples and QA pairs appear in Figure 2).

When traversing the entire graph according to these patterns and identifying the corresponding triplet(s), we ensure that the extracted triplets can only be used to generate corresponding questions. For instance, while traversing all single-hop triplets  $(e_1, r_1, e_2)$ , we ensure that  $e_1$  has an out-degree of 1 and an in-degree of 0, while  $e_2$  has an in-degree of 1 and an out-degree of 0. This approach prevents duplication of content between single-hop and multi-hop questions. Additionally, for multi-hop questions, we remove all triplet sets that can be entirely answered from the same chunk. This ensures that multi-hop questions must be answered by traversing multiple files. Finally, We restrict to these patterns because they cover the three fundamental reasoning moves in real retrieval: follow a path (chain), aggregate around a hub (star), and converge multiple

Pattern	Graph structure (template)	What it tests
Single-hop	$(e_1, r_1, e_2)$	Direct fact lookup; baseline single-chunk retrieval.
Chained-Shape	$(e_1, r_1, e_2) \rightarrow (e_2, r_2, e_3) \rightarrow \dots$	Follow 2–3 linked triplets; step-wise reasoning across chunks.
Star-Shape	$(e_1, r_1, e_2) \parallel (e_1, r_2, e_3) \parallel \dots$	Aggregate diverse facts around a focal entity; synthesize across relations.
Inverted-Star	$(e_1, r_1, e_2) \parallel (e_3, r_2, e_2) \parallel \dots$	Recognize convergent paths; combine evidence toward a common target.

Table 2: Single hop and multi-hop query pattern templates.

clues to a target (inverted-star). These patterns are expressive enough to span most cross-chunk tasks while keeping graph traversal depth and branching controllable for automatic generation, verification, and difficulty tuning.

### 3.4 QUERY GENERATION AND QUALITY ASSURANCE

For each identified pattern, we use pattern-specific prompts to generate QA pairs that use information from its triplets, corresponding ground truth chunks, and metadata storing information such as timestamp or the country name. For multi-hop questions specifically, we implement a specialized algorithm that: (1) Identifies a "pivot entity" that connects different triplets; (2) References this pivot indirectly in the question; (3) Ensures the question cannot be answered from a single chunk; (4) Performs "pivot-rarity" and "negative-distractor safety" checks to guarantee question quality. Appendix F shows the complete algorithm for generating pairs.

Finally, all generated query-answer pairs undergo rigorous quality assessment using separate LLM-based evaluation based on Claude 3.5 Haiku (Claude, 2024) that scores each query-answer pair on three dimensions from the scale of 1 to 5: (1) Readability; (2) Clarity; (3) Correctness. Only queries with scores above 3 across all dimensions are included in the final benchmark. This quality-controlled generation process creates benchmarks that effectively evaluate both retrieval accuracy and reasoning capabilities within domain-specific contexts. As source documents evolve or new ones are added, the pipeline can dynamically extend the benchmark, ensuring continued relevance for evaluating RAG systems against the latest information. Appendix F includes step-by-step measuring standards.

## 4 RARE-SET: LARGE-SCALE DOMAIN-SPECIFIC RAG DATASET

RARE-Set contains three different domains of datasets: finance, economics, and policy. We collect a heterogeneous corpus with 199 recent S&P 500 Companies’ SEC 10-k filings, 114 OECD economic surveys, and 214 Consolidated Annual Performance and Evaluation Report (CAPER) from grantees for U.S. Department of Housing and Urban Development (HUD) funded programs. Appendix E shows the full dataset statistics.

We enhance datasets quality through a variety of processing techniques. For instance, for financial reports, our preprocessing pipeline builds on Edgar-Crawler (Loukas et al., 2021), with custom modifications. Rather than preserving tables in HTML format, we convert them to a markdown structure optimized for LLM inputs. In knowledge graph extraction from financial documents, we prioritize relations involving performance metrics, operational activities, and financial events. We explicitly target generalized and reusable relations that can be applied across companies within the same industry. This approach supports the generation of multi-hop questions that span multiple companies. For economic surveys, we design prompts to emphasize policy measures, key economic indicators, and patterns of national development. In the context of policy reports, our focus is on fund allocation, program implementation, and beneficiary data.

282 Table 3: Robustness definitions under query/document settings. ✓ = counted robust only if the final answer  
 283 is correct;  $\emptyset$  = counted robust only if the model safely refuses;  $\checkmark \vee \emptyset$  = robust if either correct or safely  
 284 refuses.  $g(q, d)$  represents generator (model) given query and document.  $g(q, \emptyset)$  is the per-record no-context  
 285 probe indicating the generator can answer without retrieval. 1 denotes that the generator can answer without  
 286 retrieval, while 0 indicates it cannot.

Document setting	$g(q, \emptyset) = 1$		$g(q, \emptyset) = 0$	
	$q$ (orig.)	$q'$ (perturbed)	$q$ (orig.)	$q'$ (perturbed)
Ground-Truth Docs	✓	✓	✓	✓
Lexical-Diff (Has Answer)	✓	✓	✓	✓
Lexical-Similar (No Answer)	$\checkmark \vee \emptyset$	$\checkmark \vee \emptyset$	$\emptyset$	$\emptyset$
Real-World Retrieval	$\checkmark \vee \emptyset$	$\checkmark \vee \emptyset$	$\checkmark \vee \emptyset$	$\checkmark \vee \emptyset$

294 The benchmark contains single-hop queries and three types of multi-hop queries based on different knowledge  
 295 patterns in the knowledge graph. One thing to mention is that all of these domains are time-sensitive and can  
 296 update dynamically as time progresses.

297 Finally, to evaluate the factual accuracy of the extracted knowledge-graph triplets relative to their corresponding  
 298 source chunk, we randomly sampled 1000 triplets and assessed them using Claude Sonnet 4.5 (since  
 299 GPT-4.1 generated the triplets). The evaluation indicates that 87.1% of the sampled triplets are factually correct  
 300 with respect to their originating chunks, suggesting that the triplets effectively preserve information from  
 301 the source chunks. The verification prompt is in Appendix F. Moreover, in the LLM-based quality-assurance  
 302 stage, only 19.69% of the generated QA pairs were filtered out.

## 304 5 RARE-MET: RETRIEVAL-AWARE ROBUSTNESS METRIC

307 A robust RAG system should maintain correctness under two conditions: if the generator can already answer  
 308 the query without retrieval ( $g(q, \emptyset) = 1$ ), it must consistently give the correct answer regardless of retrieval  
 309 content; if it cannot answer without retrieval ( $g(q, \emptyset) = 0$ ), it should provide the correct answer given correct  
 310 retrieval, and otherwise safely refuse rather than hallucinate when retrieval is incorrect or irrelevant.

311 Table 3 shows the full definition of RAG robustness under different circumstances.

### 313 5.1 QUERY PERTURBATIONS

315 We define four types of query perturbations  $Q' = q'_1, q'_2, \dots, q'_n$  derived from the original query  $q$ , grouped  
 316 into two categories: **Surface-level perturbations**: (1) character-level changes; (2) word-level changes  
 317 (typos, synonyms) based on TextAttack (Morris et al., 2020); and **Advanced-level perturbations**: (1) LLM-  
 318 based grammar rewrites that preserve the query’s intrinsic meaning; (2) LLM-based additions of irrelevant  
 319 information. Appendix C.1 includes more details on constructing perturbations for each query.

### 320 5.2 DOCUMENT PERTURBATION

322 For document perturbation  $D' = d'_1, d'_2, \dots, d'_n$ , we primarily consider two directions: lexical relevance and  
 323 answer relevance. Similarly to definitions under query perturbation, the lexical relevance measure changes  
 324 of document styles. Answer relevance, on the other hand, determines whether the retrieved document truly  
 325 contains the answer required by the question. As we consider lexical perturbation and answer perturbation as  
 326 two dimensions, we define three document perturbations which encompassed all possible distributions of  
 327 retrieval documents. (1) Documents with the similar lexical style but answers are different: directly remove  
 328 the answer sentence/words from the ground truth chunk. (2) Documents with different lexical style but

329 answer is similar/identical: LLM-based back-translation. (3) Real-world retrieval results ( $D_{ret}$ ): constructing  
 330 a real-world simulated retrieval process based on LangChain (Chase & contributors, 2022) (including a  
 331 re-ranking model). The first two document perturbations are introduced to more clearly examine how different  
 332 relevance types—lexical or answer-based—affect the overall robustness of the RAG system.

333 Appendix B shows all types of document perturbations under such relevance and reason of evaluating from  
 334 these perspectives. Appendix C.2 reveals construction process in details. The first two document perturbations  
 335 are introduced to more clearly examine how different relevance types—lexical or answer-based—affect the  
 336 overall robustness of the RAG system.

### 338 5.3 ROBUSTNESS METRICS

Metric	Fixed / Varied	Expression
Overall Robustness	Fixed: $\emptyset$ ; Varied: $q \in Q, d \in D$	$\frac{1}{ Q  D } \sum_{q \in Q} \sum_{d \in D} f(g(q, d), a)$
Query Robustness	Fixed: $d_{gt}$ ; Varied: $q' \in Q'$	$\frac{1}{ Q' } \sum_{q' \in Q'} f(g(q', d_{gt}), a)$
Document Robustness	Fixed: $q$ ; Varied: $d' \in D'$	$\frac{1}{ D' } \sum_{d' \in D'} f(g(q, d'), a)$
Real-World Retrieval Robustness	Fixed: $q$ ; Varied: $d'_i \in D_{ret}$	$\frac{1}{ D_{ret} } \sum_{d'_i \in D_{ret}} f(g(q, d'_i), a)$

351 Table 4: Definition of different robustness score.  $f(pred, ans)$  indicates the open-ended prediction and  
 352 ground truth LLM-based comparison function. All other notations are identical to the previous section.  
 353 Appendix D also provides an additional table to understand these notation better.

## 355 6 ROBUSTNESS EXPERIMENTS AND ANALYSIS

### 356 6.1 EXPERIMENTAL SETTING

360 We perform our experiments on a total of 6000 QA pairs for three domains, each of which has 1000 single-hop  
 361 questions and 1000 multi-hop questions. Retrieval is evaluated with three top-ranking embedding models  
 362 from the MTEB leaderboard: E5-Large-Instruct Es et al. (2025), Jina-Embedding-v3 Sturua et al. (2024),  
 363 and Stella-En-1.5B-v5 Zhang et al. (2025a). For the RAG system’s generators, we evaluate both leading  
 364 open-source LLMs, including Qwen 3 Yang et al. (2025a) and the Llama 3.2 family Grattafiori et al. (2024),  
 365 as well as proprietary models accessed through commercial APIs. The Llama 3.2 series is served via the  
 366 Amazon Bedrock API, while closed-source GPT models are accessed directly through the OpenAI API. Our  
 367 total expenditure on the GPT-4.1 series models was approximately \$3400, which includes costs for KG triple  
 368 extraction (GPT-4.1 only), QA generation (GPT-4.1 only), and evaluations involving three different GPT  
 369 models. All generators are configured to operate deterministically (temperature = 0) with a maximum output  
 370 length of 1024 tokens. Although models are instructed to provide concise final answers, chain-of-thought  
 371 reasoning is explicitly encouraged in their outputs to facilitate their abilities. We close Qwen 3’s internal  
 372 thinking mode for fair comparison. Appendix G.3 proves our results are statistical significance.

373 For the Qwen 3 series, we deploy both vLLM Kwon et al. (2023) servers (for larger models) and SGLang  
 374 Zheng et al. (2024) servers (for smaller models), running in parallel with their official recommended settings  
 375 to optimize inference throughput and performance. These open-source models are executed on a cluster of  
 16 NVIDIA L40S GPUs. To accelerate large-scale experimentation, multiple server instances are launched

376 concurrently, and inference requests are distributed across them. Completion of the full experimental suite  
 377 requires approximately five days.  
 378

379 To quantify the discrepancy between predictions and ground-truth answers, we design a two-stage evaluation  
 380 pipeline. In the first stage, both prediction and reference strings are normalized, after which exact and inclusive  
 381 string matches are implemented. If no lexical match is detected, the second stage employs Claude-3-Haiku  
 382 Anthropic (2024) judging with a carefully engineered evaluation prompt (Appendix F) to determine whether  
 383 prediction matches the ground truth. Using Claude model can minimize bias and ensure neutrality in the  
 384 evaluation.

385 **Table 5:** Robustness results across different models and metrics

386 <b>Model</b>	387 <b>Overall</b>	388 <b>Query</b>	389 <b>Document</b>	390 <b>Retrieval</b>
388 Llama-3.2-1B-Instruct	0.318	0.280	0.254	0.389
389 Llama-3.2-3B-Instruct	0.607	0.459	0.587	0.649
390 Llama-3.2-11B-Vision-Instruct	0.627	0.658	0.630	0.610
391 Llama-3.2-90B-Vision-Instruct	<b>0.782</b>	0.691	<b>0.771</b>	<b>0.820</b>
392 Qwen3-4B	0.665	0.734	0.700	0.611
393 Qwen3-8B	0.698	0.714	0.721	0.667
394 Qwen3-32B	0.664	0.732	0.701	0.598
395 GPT-4.1-nano	0.589	0.613	0.651	0.531
396 GPT-4.1-mini	0.646	0.730	0.651	0.613
397 GPT-4.1	0.675	<b>0.761</b>	0.668	0.654

## 399 6.2 OVERALL ROBUSTNESS TRENDS ACROSS MODEL SCALES

400 Examining the overall robustness scores in the Table 5 shows that larger models generally demonstrate  
 401 superior robustness. GPT-4.1 achieves a robustness score that surpasses those of its smaller models, GPT-4.1  
 402 mini and GPT-4.1 nano. A similar scaling-law is observed within the Llama 3.2 series: Llama-3.2-90B-Vision-  
 403 Instruct exhibits a markedly higher robustness score than any other model. Surprisingly, it even surpasses  
 404 closed models such as GPT series. However, size alone does not always reflects the robustness. For example,  
 405 Qwen3-32B attains an overall robustness score lower than that of the smaller, but architecturally similar  
 406 Qwen3-8B and even Qwen3-4B. This phenomenon is widely observed across the Qwen3 family of models.  
 407 The Qwen3 models consistently maintain a relatively high robustness score, even for smaller-scale variants  
 408 such as the 4B model. In addition, compared with other robustness scores, the document score does not  
 409 exhibit a significant improvement as model size increases; in fact, some models even show regression.

410 This is primary because some larger models are more likely to answer directly with hallucinations, even  
 411 when they lack the ability to answer the question or when the given document does not contain the answer.  
 412 However, certain smaller models are more likely to decline questions that exceed their capabilities. As shown  
 413 in Figure 9, while the ground-truth and lexical-different-with-answer robustness scores generally follow the  
 414 scaling law, the other two types of document robustness do not, especially the lexical-similar-without-answer  
 415 robustness. Smaller models typically achieve higher scores in this sub-score due to their higher probability of  
 416 issuing refusals. Larger models tend to respond to the question more frequently than smaller models, which  
 417 leads to lower lexical-similarity robustness scores. This behavior ultimately affects the document robustness  
 418 as well as the overall robustness score.

419 In contrast, Figure 8 shows that there are no significant differences across different query perturbations  
 420 within each model, indicating that current models exhibit similar consistency when facing various query  
 421 perturbations. Across models, the query robustness scores generally follow the scaling law, with Qwen3  
 422 models consistently achieving high scores.

423  
424

### 6.3 DOMAIN-SPECIFIC AND MULTI-HOP QUESTIONS ROBUSTNESS

425  
426  
427  
428  
429  
430  
431

Figure 7 indicates that the robustness of RAG systems is heavily influenced by domain-specific factors. RAG system perform best in finance reports, which typically feature standardized terminology and numerical data. However, they are struggling most with the economics survey, which often involves complex causal relationships and varied terminology. In addition, single-hop queries consistently yield higher robustness scores than multi-hop queries across all domains and perturbations (Figure 6). This trend is amplified in smaller models, suggesting that multi-hop reasoning capabilities require substantial model capacity to maintain robustness under perturbations.

432  
433

Appendix G includes all evaluation results.

434  
435

## 7 CONCLUSION

436  
437  
438  
439  
440  
441  
442  
443  
444  
445

In conclusion, we introduce RARE, a comprehensive framework for data generation and evaluating RAG robustness that addresses critical gaps in existing benchmarks. Our knowledge-graph-based pipeline (RARE-Get) automatically extracts relations from specialized corpora and generates multilevel questions through pattern-based traversal, enabling dynamic dataset evolution without manual curation. The resulting benchmark (RARE-Set) comprises 48295 questions across finance, economics, and policy domains, featuring single-hop and complex multi-hop questions. Our robust evaluation metrics (RARE-Met) systematically measure resilience against query, document, and retrieval perturbations. Experiments reveal that RAG systems consistently demonstrate higher robustness in finance than economics, and single-hop queries outperform multi-hop ones across all domains, providing crucial insights for developing more reliable RAG systems for real-world applications.

446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469

470 REFERENCES  
471

472 AI Anthropic. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 1(1):4, 2024.

473 Tianyu Cao, Neel Bhandari, Akhila Yerukola, Akari Asai, and Maarten Sap. Out of style: Rag’s fragility to  
474 linguistic variation, 2025. URL <https://arxiv.org/abs/2504.08231>.

475 Harrison Chase and contributors. Langchain: A framework for developing with large language models.  
476 <https://github.com/langchain-ai/langchain>, 2022. Accessed: 2025-09-25.

477 Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. Benchmarking large language models in retrieval-  
478 augmented generation, 2023. URL <https://arxiv.org/abs/2309.01431>.

479 Claude. Claude 3.5 haiku, 2024. <https://www.anthropic.com/clause/haiku>.

480 Qingxiu Dong, Jingjing Xu, Lingpeng Kong, Zhifang Sui, and Lei Li. Statistical knowledge assessment for  
481 large language models, 2023. URL <https://arxiv.org/abs/2305.10519>.

482 Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. Ragas: Automated evaluation of  
483 retrieval augmented generation, 2025. URL <https://arxiv.org/abs/2309.15217>.

484 Robert Friel, Masha Belyi, and Atindriyo Sanyal. Ragbench: Explainable benchmark for retrieval-augmented  
485 generation systems, 2025. URL <https://arxiv.org/abs/2407.11005>.

486 Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang,  
487 and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024. URL  
488 <https://arxiv.org/abs/2312.10997>.

489 Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle  
490 use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of*  
491 *the Association for Computational Linguistics*, 9:346–361, 2021. doi: 10.1162/tacl\_a\_00370. URL  
492 <https://aclanthology.org/2021.tacl-1.21/>.

493 Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle,  
494 Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal,  
495 Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark,  
496 Arun Rao, Aston Zhang, Aurelien Rodriguez, Auster Gregerson, Ava Spataru, Baptiste Roziere, Bethany  
497 Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris  
498 McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer,  
499 Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David  
500 Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor  
501 Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco  
502 Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme  
503 Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo  
504 Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang,  
505 Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der  
506 Linde, Jennifer Billock, Jenny Hong, Janya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie  
507 Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe,  
508 Junteng Jia, Kalyan Vasudevan Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth  
509 Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal  
510 Lakhota, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis  
511 Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi,  
512 Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham,  
513 Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan,  
514

517 Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang,  
 518 Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng,  
 519 Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao  
 520 Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic,  
 521 Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro,  
 522 Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa,  
 523 Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath  
 524 Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra,  
 525 Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman,  
 526 Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov,  
 527 Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent  
 528 Gonguet, Virginie Do, Vish Vogeti, Vitor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin  
 529 Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia,  
 530 Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen,  
 531 Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing  
 532 Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld,  
 533 Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei  
 534 Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres  
 535 Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani,  
 536 Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin  
 537 Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie  
 538 Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock,  
 539 Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton,  
 540 Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu,  
 541 Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer,  
 542 Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich,  
 543 Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery,  
 544 Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar,  
 545 Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni,  
 546 Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee,  
 547 Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan,  
 548 Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen  
 549 Suk, Henry Aspren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov,  
 550 Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam,  
 551 Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein,  
 552 Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan  
 553 McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay  
 554 Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran  
 555 Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro  
 556 Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian  
 557 Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso,  
 558 Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal  
 559 Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike  
 560 Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam,  
 561 Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta,  
 562 Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar,  
 563 Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre  
 Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao,  
 Rachel Rodriguez, Rafi Ayub, Raghatham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy,  
 Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta,

564 Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov,  
 565 Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay,  
 566 Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang  
 567 Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie  
 568 Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer  
 569 Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar  
 570 Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy  
 571 Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish  
 572 Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihalea, Vladimir Ivanov, Wei Li,  
 573 Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan  
 574 Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang,  
 575 Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He,  
 576 Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma.  
 577 The llama 3 herd of models, 2024. URL <https://arxiv.org/abs/2407.21783>.

578 Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function  
 579 using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman (eds.), *Proceedings of the 7th  
 580 Python in Science Conference*, pp. 11 – 15, Pasadena, CA USA, 2008.

581 Joel Jang, Seonghyeon Ye, Changho Lee, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, and  
 582 Minjoon Seo. TemporalWiki: A lifelong benchmark for training and evaluating ever-evolving language  
 583 models. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference  
 584 on Empirical Methods in Natural Language Processing*, pp. 6237–6250, Abu Dhabi, United Arab Emirates,  
 585 December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.418.  
 586 URL <https://aclanthology.org/2022.emnlp-main.418/>.

587 Jungo Kasai, Keisuke Sakaguchi, Yoichi Takahashi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev,  
 588 Noah A. Smith, Yejin Choi, and Kentaro Inui. Realtime qa: What’s the answer right now?, 2024. URL  
 589 <https://arxiv.org/abs/2207.13332>.

590 Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti,  
 591 Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew  
 592 Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A  
 593 benchmark for question answering research. *Transactions of the Association for Computational Linguistics*,  
 594 7:452–466, 2019. doi: 10.1162/tacl\_a\_00276. URL <https://aclanthology.org/Q19-1026/>.

595 Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez,  
 596 Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with  
 597 pagedattention, 2023. URL <https://arxiv.org/abs/2309.06180>.

598 Lefteris Loukas, Manos Fergadiotis, Ion Androutsopoulos, and Prodromos Malakasiotis. EDGAR-CORPUS:  
 599 Billions of Tokens Make The World Go Round. In *Proceedings of the Third Workshop on Economics  
 600 and Natural Language Processing (ECONLP)*, pp. 13–18, Punta Cana, Dominican Republic, November  
 601 2021. Association for Computational Linguistics. URL <https://aclanthology.org/2021.econlp-1.2>.

602 Jannat Ara Meem, Muhammad Shihab Rashid, Yue Dong, and Vagelis Hristidis. Pat-questions: A self-  
 603 updating benchmark for present-anchored temporal question-answering, 2024. URL <https://arxiv.org/abs/2402.11034>.

604 John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. Textattack: A framework  
 605 for adversarial attacks, data augmentation, and adversarial training in nlp. In *Proceedings of the 2020  
 606 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 119–126,  
 607 2020.

611 Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. MTEB: Massive text embedding  
 612 benchmark. In Andreas Vlachos and Isabelle Augenstein (eds.), *Proceedings of the 17th Conference*  
 613 *of the European Chapter of the Association for Computational Linguistics*, pp. 2014–2037, Dubrovnik,  
 614 Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.148.  
 615 URL <https://aclanthology.org/2023.eacl-main.148/>.

616 OpenAI. Introducing gpt-4.1 in the api, 2025. <https://openai.com/index/gpt-4-1/>.

617 Phillip Schneider, Tim Schopf, Juraj Vladika, Mikhail Galkin, Elena Simperl, and Florian Matthes. A decade  
 618 of knowledge graphs in natural language processing: A survey, 2022. URL <https://arxiv.org/abs/2210.00105>.

619 Hongda Shen and Eren Kurshan. Temporal knowledge distillation for time-sensitive financial services  
 620 applications, 2023. URL <https://arxiv.org/abs/2312.16799>.

621 Saba Sturua, Isabelle Mohr, Mohammad Kalim Akram, Michael Günther, Bo Wang, Markus Krimmel,  
 622 Feng Wang, Georgios Mastrapas, Andreas Koukounas, Andreas Koukounas, Nan Wang, and Han Xiao.  
 623 jina-embeddings-v3: Multilingual embeddings with task lora, 2024. URL <https://arxiv.org/abs/2409.10173>.

624 Yixuan Tang and Yi Yang. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop  
 625 queries, 2024. URL <https://arxiv.org/abs/2401.15391>.

626 Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions  
 627 via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–  
 628 554, 2022. doi: 10.1162/tacl\_a\_00475. URL <https://aclanthology.org/2022.tacl-1.31/>.

629 Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung,  
 630 Denny Zhou, Quoc Le, and Thang Luong. FreshLLMs: Refreshing large language models with search  
 631 engine augmentation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the*  
 632 *Association for Computational Linguistics: ACL 2024*, pp. 13697–13720, Bangkok, Thailand, August  
 633 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.813. URL <https://aclanthology.org/2024.findings-acl.813/>.

634 Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Improving text  
 635 embeddings with large language models. *arXiv preprint arXiv:2401.00368*, 2023.

636 Cheng Xu, Shuhao Guan, Derek Greene, and M-Tahar Kechadi. Benchmark data contamination of large  
 637 language models: A survey, 2024. URL <https://arxiv.org/abs/2406.04244>.

638 An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao,  
 639 Chengan Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge,  
 640 Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang,  
 641 Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng,  
 642 Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan  
 643 Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang,  
 644 Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yingqi Zhang, Yu Wan, Yuqiong Liu, Zekun  
 645 Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025a. URL  
 646 <https://arxiv.org/abs/2505.09388>.

647 Shiping Yang, Jie Wu, Wenbiao Ding, Ning Wu, Shining Liang, Ming Gong, Hengyuan Zhang, and Dongmei  
 648 Zhang. Quantifying the robustness of retrieval-augmented language models against spurious features in  
 649 grounding data, 2025b. URL <https://arxiv.org/abs/2503.05587>.

658 Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze Daniel  
659 Gui, Ziran Will Jiang, Ziyu Jiang, Lingkun Kong, Brian Moran, Jiaqi Wang, Yifan Ethan Xu, An Yan,  
660 Chenyu Yang, Eting Yuan, Hanwen Zha, Nan Tang, Lei Chen, Nicolas Scheffer, Yue Liu, Nirav Shah,  
661 Rakesh Wanga, Anuj Kumar, Wen-tau Yih, and Xin Luna Dong. Crag - comprehensive rag benchmark. In  
662 A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances  
663 in Neural Information Processing Systems*, volume 37, pp. 10470–10490. Curran Associates, Inc.,  
664 2024. URL [https://proceedings.neurips.cc/paper\\_files/paper/2024/file/1435d2d0fca85a84d83ddcb754f58c29-Paper-Datasets\\_and\\_Benchmarks\\_Track.pdf](https://proceedings.neurips.cc/paper_files/paper/2024/file/1435d2d0fca85a84d83ddcb754f58c29-Paper-Datasets_and_Benchmarks_Track.pdf).

665  
666  
667 Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and  
668 Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering, 2018.  
669 URL <https://arxiv.org/abs/1809.09600>.

670  
671 Dun Zhang, Jiacheng Li, Ziyang Zeng, and Fulong Wang. Jasper and stella: distillation of sota embedding  
672 models, 2025a. URL <https://arxiv.org/abs/2412.19048>.

673 Kepu Zhang, Zhongxiang Sun, Weijie Yu, Xiaoxue Zang, Kai Zheng, Yang Song, Han Li, and Jun Xu.  
674 Qe-rag: A robust retrieval-augmented generation benchmark for query entry errors, 2025b. URL <https://arxiv.org/abs/2504.04062>.

675  
676 Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Livia Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao,  
677 Christos Kozyrakis, Ion Stoica, Joseph E Gonzalez, et al. Sclang: Efficient execution of structured language  
678 model programs. *Advances in neural information processing systems*, 37:62557–62583, 2024.

679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704

705 **A CHUNKING TECHNIQUES**  
706707 For each domain, here are the step-by-step explanation for chunking.  
708709 **A.1 FINANCE**  
710

- 711 1. Load filing JSON and prepare metadata (CIK, company, filing type/date, period; optional GICS  
712 sector/subindustry).
- 713 2. Preprocess `item_7`: split by lines, detect section titles (regex on uppercase "Item" patterns), detect  
714 table-like blocks (pipe-delimited), group tables with nearby narrative, and merge short title-only  
715 segments into adjacent content.
- 716 3. For each segment:
  - 717 • If it contains a table, emit a single chunk with `contains_table=true`.
  - 718 • Otherwise, split text with a token-aware recursive splitter (`chunk_size=800`,  
719 `overlap=100`, tiktoken-based length), merge very short fragments (< 30 words), and carry  
720 the section title into the first chunk; mark `contains_table=false`.
- 722 4. Assign chunk IDs and attach metadata.

723 **A.2 ECONOMICS**  
724

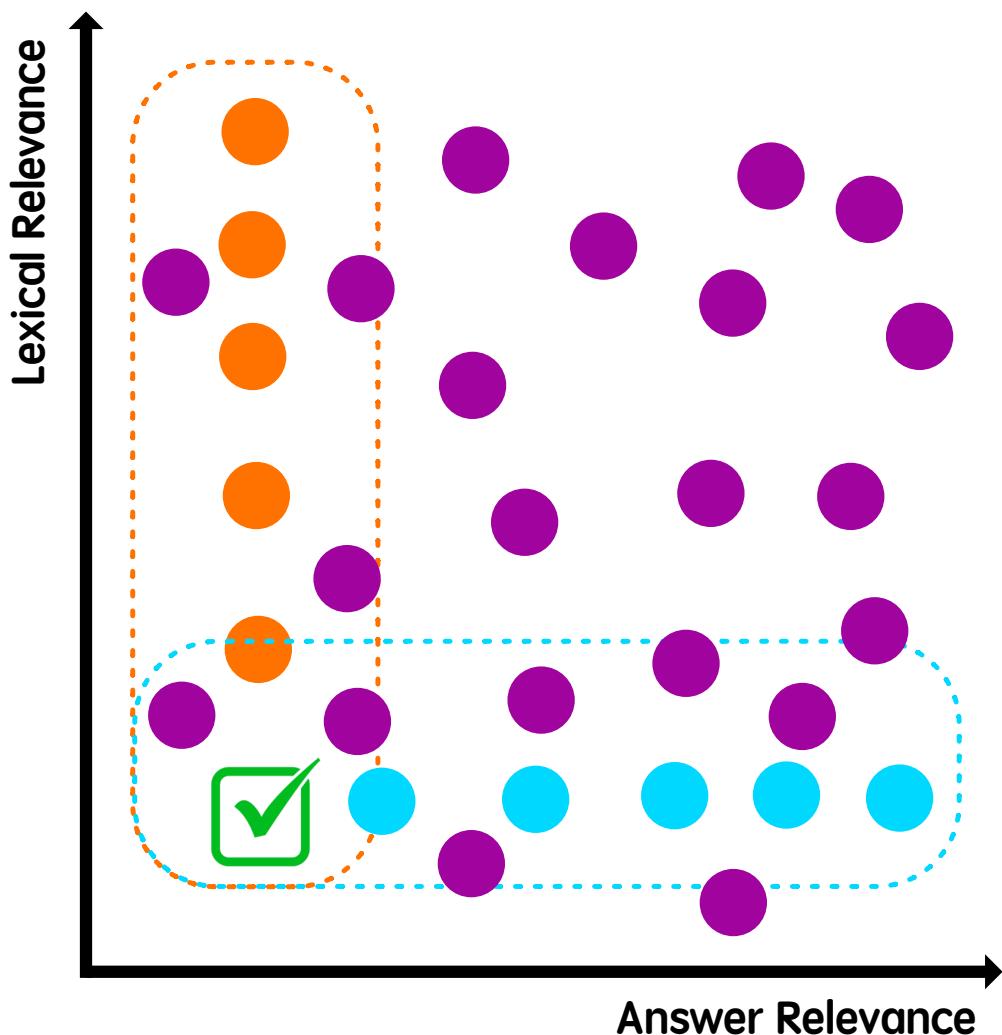
- 725 1. Load structured content; extract `file_country` and `file_year` from the first "OECD Eco-  
726 nomic Surveys:" line; initialize per-chunk metadata.
- 727 2. Start near the first table ( $idx = \max(0, \text{first-table-index} - 1)$ ) and iterate rows.
- 728 3. For text rows, accumulate lines until around 600 words, then flush a `text` chunk with  
729 `chunk_page_idx`.
- 731 4. For table rows, convert HTML to Markdown, prepend detected caption (from row or preceding short  
732 "Table" lines) and append footnotes; emit a `table` chunk with `chunk_page_idx`.
- 733 5. Flush any remaining text; assign chunk IDs and attach metadata.

734 **A.3 POLICY**  
735

- 737 1. Load structured content and join with external metadata row by `id`; prepare per-chunk metadata  
738 (plan type, `file_grantee`, `file_state`, `file_year`).
- 739 2. Trim trailing content starting at the first "Attachment" header.
- 740 3. For text rows, accumulate lines until around 600 words, then flush a `text` chunk with  
741 `chunk_page_idx`.
- 742 4. For table rows, convert HTML to Markdown; if captions/footnotes exist, prepend/append them; emit  
743 a `table` chunk with `chunk_page_idx`.
- 745 5. Flush any remaining text; assign chunk IDs and attach metadata.

746 **B THREE TYPES OF DOCUMENT PERTURBATIONS**  
747749 Figure 3 illustrates that real-world retrieval results (**violet dots**) are scattered throughout the entire space of  
750 lexical relevance and answer relevance, indicating that outcomes can occur in any region depending on the  
751 retrieval performance. To study robustness, we introduce document perturbations in two targeted regions:

752  
 753 answer-similar but lexically different (orange) and answer-different but lexically similar (blue), which allow  
 754 us to isolate and examine the impact of lexical versus answer relevance on RAG system performance.  
 755  
 756  
 757  
 758  
 759  
 760  
 761  
 762  
 763  
 764  
 765  
 766  
 767  
 768  
 769  
 770  
 771  
 772  
 773  
 774  
 775  
 776  
 777  
 778  
 779  
 780  
 781  
 782  
 783  
 784  
 785  
 786  
 787  
 788  
 789



790 Figure 3: Three types of document perturbations measured by two relevances.  
 791  
 792

## 793 C PERTURBATION CONSTRUCTIONS

### 794 C.1 QUERY PERTURBATIONS

795 1. Character-level noise: Use TextAttack Augmenters such as CompositeTransformation,  
 796 WordSwapQWERTY and WordSwapRandomCharacterDeletion (swap only 10% of the charac-

ters); sample up to 5 variants and select via embedding model (first passing, otherwise maximum similarity score).

2. Word-level substitutions: Use TextAttack Augmenter with WordSwapEmbedding (`max_candidates = 50`) and the same constraints; sample up to 5 variants (swap only 10% of the vocabulary) and select with the same embedding model similarity filter.
3. Insert irrelevant info (LLM): Use GPT-4.1 to rewrite the query by inserting one domain-relevant but answer-irrelevant detail (3 candidates); keep the highest-similarity candidate .
4. Grammar perturbation (LLM): Use GPT-4.1 to rephrase only grammar/punctuation/word order (3 candidates); keep the highest-similarity candidate.

## C.2 DOCUMENT PERTURBATIONS

1. Regex deletion: Use Python `re.sub`, `re.escape` and `re.IGNORECASE` to remove exact supporting sentences from answer-bearing chunks; compute semantic similarity using embedding model to the original chunk, ensuring that their core contents are not changed.
2. Back-translation (LLM): Use GPT-4.1 to translate chunks EN→FR then FR→EN in batch; compute similarity to the original with embedding model and attach the perturbed text with its score.

## D RARE-MET NOTATION REFERENCE

Table 6: Notations and Definitions

Notation	Definition
$q$	Original query.
$q' \in Q'$	Perturbed query; $Q'$ is the set of query perturbations.
$Q = \{q\} \cup Q'$	Full query set (original + perturbations).
$d_{\text{gt}}$	Ground-truth document.
$d' \in D'$	Perturbed document; $D'$ is the set of document perturbations.
$D = \{d_{\text{gt}}\} \cup D'$	Full document set (ground truth + perturbations).
$\emptyset$	Empty context (no retrieval).
$g(q, d)$	Generator producing the results given question $q$ with context $d$ .
$a$	Ground-truth answer.
$f(g(q, d), a) \in \{0, 1\}$	Robustness judge (1 = robust, 0 = not), following Table 3.
$g(q, \emptyset) \in \{0, 1\}$	Parametric-knowledge probe: 1 = can answer without retrieval; 0 = cannot.
$D_{\text{ret}}$	Set of documents returned by the evaluated retrievers.
$d'_i \in D_{\text{ret}}$	A retrieved document used in real-world evaluation (e.g., top- $k$ per retriever).

846  
847

## E RARE-SET STATISTICS

848  
849

Table 7: Dataset Statistics by Domain

Domain	Financial	Economics	Policy
Document	199	114	214
Chunk	19825	12915	7014
Time Scope	2024-2025	2020-2025	2024-2025
<b>Total # of Eligible Triplet/Triples</b>			
Single-hop	17585	6719	6176
Chained (multi-hop)	11193	22256	82885
Star-shaped (multi-hop)	2707	1780	4868
Inverted-star-shaped (multi-hop)	558	2636	7377
<b>Query (Train)</b>			
Single-hop	7362	6715	6125
Chained (multi-hop)	7930	3863	7563
Star-shaped (multi-hop)	833	511	661
Inverted-star-shaped (multi-hop)	64	415	253
<b>Query (Test)</b>			
Single-hop	1000	1000	1000
Chained (multi-hop)	687	774	805
Star-shaped (multi-hop)	289	193	135
Inverted-star-shaped (multi-hop)	24	33	60

868

869  
870

## F PROMPTS

871  
872

We will use the economic dataset prompts as the example.

873  
874

## Dataset Generation Prompt: Triplets Extraction

875  
876

You are an economic analyst skilled at interpreting OECD Economic Surveys.

Your task is to extract structured triplets consisting of {"entity\_1", "relation", "entity\_2"} from provided consecutive text chunks from a single OECD Economic Survey.

Each triplet must be supported explicitly by one specific chunk, but other chunks can be referenced to form insightful, multi-hop triplets.

You should include the source chunk ID and source sentence as the metadata of the triplets.

881  
882**TASK: EXTRACT STRUCTURED MULTI-HOP TRIPLETS**883  
884

Extract triplets fitting these multi-hop categories:

885  
886

- Connected Chain
- Star
- Inverted Star

887  
888**1. Connected Chain Triplets:**889  
890  
891  
892

- Extract an initial triplet: <entity\_1, relation, entity\_2>.
- Then identify subsequent triplets where entity\_2 of the previous triplet becomes entity\_1 of the next.

893  
 894 - Ideally, different subsequent triplets should be sourced from different chunks.  
 895 - Extract as many meaningful chains as possible.  
 896 - Skip if no valid connected chain is available.

897  
 898 *Example:*

899 - {"entity\_1": "Luxembourg", "relation": "implemented", "entity\_2": "free public transport"}  
 900 - {"entity\_1": "free public transport", "relation": "aims to reduce", "entity\_2": "carbon emissions"}

901  
 902 **2. Star Triplets:**

903 - One root entity branching into multiple distinct relationships.  
 904 - Each branch must independently derive from a unique chunk.  
 905 - Skip if no meaningful star relationship is possible.

906  
 907 *Example:*

908 - {"entity\_1": "Luxembourg", "relation": "invests in", "entity\_2": "renewable energy"}  
 909 - {"entity\_1": "Luxembourg", "relation": "develops", "entity\_2": "sustainable transport infrastruc-  
 910 ture"}

911  
 912 **3. Inverted Star Triplets:**

913 - Two distinct entities connected through a shared attribute (entity\_2).  
 914 - Relations may differ and offer varied perspectives on the attribute.  
 915 - Skip if no valid inverted star relationship is possible.

916  
 917 *Example:*

918 - {"entity\_1": "Luxembourg", "relation": "faces challenges in", "entity\_2": "housing affordability"}  
 919 - {"entity\_1": "OECD recommendations", "relation": "address", "entity\_2": "housing affordability"}

920  
 921 **REQUIRED STRUCTURE:**

922 Each extracted triplet must include:

923 - entity\_1 (str)  
 924 - relation (str)  
 925 - entity\_2 (str)  
 926 - answer\_chunk\_id (str)  
 927 - The chunk ID is at the very beginning of each text chunk, such as "Chunk ID: economics\_0e32d909-  
 928 en\_chunk\_9".  
 929 - You should copy the chunk ID where the triplet is extracted from as the "answer\_chunk\_id".  
 930 - source\_sentence (str)  
 931 - Extracted exactly from the supporting chunk, COPY WORD BY WORD.  
 932 - If sourced from a table, strictly include relevant row, column, and specific data only.

933  
 934 **CRITICAL INSTRUCTIONS:**

935 *Relations:*

936 - Generalized and reusable across similar economic and policy contexts.  
 937 - Concise and specific (2-4 words preferred).  
 938 - Use standard economic and policy terminology.

940 - Avoid specific dates or overly detailed references in the relations.  
 941 *Good Examples:*  
 942 - "implemented", "faces challenges in", "invests in", "promotes"  
 943 *Bad Examples:*  
 944 - "introduced free transport in 2020", "planned reforms announced in 2023"  
 945  
 946  
 947 *Entities:*  
 948 - Clearly specify entities (avoid general terms like "the country" or "the government").  
 949 - Maintain consistent terminology when referring to similar concepts, such as using "Luxembourg" all  
 950 the time instead of using "Luxembourg government" sometimes.  
 951 - Include specific, detailed information relevant to economic policies, recommendations, or outcomes.  
 952 - For table-derived entities, clearly indicate row, column, and description.  
 953  
 954 **Goal:**  
 955 Try to extract 15 to 20 triplets. If no valid connected triplets can be extracted, return an empty array:  
 956 []  
 957  
 958

## Dataset Generation Prompt: Single-Hop QA Pairs Generation

959 Create an economics-related natural question-answer pair using a relation triplet (entity\_1, relation,  
 960 entity\_2) based on the text context and the file metadata where the triplet was extracted.  
 961

### Requirements

962 - The question and answer should be entirely based on the given text context; that is, one can only  
 963 generate the correct answer from the information available in the context.  
 964 - Always use "{file\_country}" instead of "{file\_country} government," "government," or "country" to  
 965 make the query more specific.  
 966 - You should use entity\_1 or entity\_2 as the answer to the question and construct the question using  
 967 the other entity and relation with appropriate context information.  
 968 - Aim to formulate questions that appear natural and are likely to be asked by a human.  
 969 - Avoid generating questions that are overly general or vague, where multiple ground truth chunks  
 970 could answer the question or it would be hard to retrieve the ground truth chunk given the question.  
 971 You **MUST** return an EMPTY string for question and answer in this case.  
 972

### Examples

#### Example 1:

973 *Triplet:*

974 {"entity\_1": "inflation", "relation": "is", "entity\_2": "2.9% in 2023"}  
 975

976 *Text Context:*

977 [Full example context is omitted...]  
 978

979 *Metadata:*

980 - File Type: OECD Economic Surveys  
 981 - Country Surveyed: Luxembourg  
 982 - Survey Year: 2023  
 983

984 *Output:*

```

987
988 {"question": "What is the inflation of Luxembourg in 2023?", "answer": "2.9%"}
989
990 Example of Vague Triplet (Should Return Empty):
991 Triplet:
992 {"entity_1": "luxembourg", "relation": "should maintain", "entity_2": "prudent fiscal policy"}
993 Text Context:
994 [Full example context is omitted...]
995 Metadata:
996 - File Type: OECD Economic Surveys
997 - Country Surveyed: Luxembourg
998 - Survey Year: 2023
999 Output:
1000 {"question": "", "answer": ""}
1001
1002 Output Format
1003 Respond in JSON format with "question" and "answer" fields encapsulating the formulated question
1004 and its answer.
1005
1006 Notes
1007 Ensure questions are specific to the context provided, emphasizing precision and clarity in wording.
1008 If no singular answer emerges due to generality, opt for returning an empty dictionary to indicate an
1009 unsuitably specific query.
1010
1011
1012
1013 Dataset Generation Prompt: Multi-Hop QA Pairs Generation
1014
1015 You are a benchmark designer creating multi-hop retrieval questions based on three types of multi-hop
1016 triplets.
1017
1018 Input
1019 - Triplet 1 = ({head1}, {rel1}, {tail1}) <- extracted from Chunk 1
1020 - Triplet 2 = ({head2}, {rel2}, {tail2}) <- extracted from Chunk 2
1021 - Chunk 1: {chunk1}
1022 - Chunk 2: {chunk2}
1023
1024 Multi-hop Triplets DEFINITIONS
1025 1. Chain Triplets
1026 - Guarantee: {tail1} == {head2}
1027 - Define A = {head1}, B = {tail1} / {head2}, C = {tail2}
1028 2. Star-shaped Triplets
1029 - Guarantee: {head1} == {head2}
1030 - Define A = {tail1}, B = {head1} / {head2}, C = {tail2}
1031 3. Inverted-star-shaped Triplets
1032 - Guarantee: {tail1} == {tail2}
1033 - Define A = {head1}, B = {tail1} / {tail2}, C = {head2}

```

1034  
1035  
1036  
1037  
1038**GOAL**

Write ONE natural-language multi-hop question that \*requires\* evidence from both chunks and answer it succinctly (no full sentences, only essential information).

1039

**ALGORITHM**

1. Decide whether the final answer will be A or C.

- Pick A if you can phrase the question so the solver must:

- hop-1: use (C, rel2) to identify B,
- hop-2: use (B, rel1) to reach A.

- Pick C if you can phrase the question so the solver must:

- hop-1: use (A, rel1) to identify B,
- hop-2: use (B, rel2) to reach C.

2. Write a fluent, specific, and natural question that:

- References the pivot B indirectly (via the opposite hop as above).

- Omits the answer itself.

- Cannot be answered from a single chunk.

- Includes detailed and specific context from the source text chunks. DO NOT just use "according to OECD Economic Survey".

- BAD example: "What is the primary export sector of the country that faces risk from global supply chain disruptions?" (Too vague; could refer to any country)

- GOOD example: "What is the primary export sector of the country that faces risk from global supply chain disruptions in Q3 2021?" (Specific to the context and time frame)

3. Return the answer based on A or C. Ensure the answer precisely matches the facts provided in the context.

1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058**EXAMPLE**

{ "entity\_1": "forward-looking fuel-tax trajectory", "relation\_1": "would reduce", "entity\_2": "reliance on combustion-engine cars" }

{ "entity\_1": "reliance on combustion-engine cars", "relation\_2": "drives", "entity\_2": "transport-sector emissions" }

*\*question\**: Which forward-looking tax trajectory is proposed to cut the main driver of transport-sector emissions?

*\*answer\**: forward-looking fuel-tax trajectory

1059

**QUALITY CHECKS**

- Pivot-rarity: B must be distinctive ( $\geq 2$  meaningful words, not generic terms like "measures", "it", "the company"). If B is too generic, output empty strings for the question and answer.

- Negative-distractor safety: Ask could a system answer your question after retrieving only \*one\* chunk? If yes, output empty strings for the question and answer.

1060

**OUTPUT**

Respond in JSON format with question and answer only as shown below:

```
{
  "question": "...",
  "answer": "..."}
```

1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080

1081  
1082  
1083  
1084  
1085  
1086

}

## 1087 Dataset Generation Prompt: Single-Hop QA Pairs Quality Assurance

1088 **Single-Hop Query Quality Evaluator**1089 You are an expert evaluator of single-hop queries. Assess each query's quality across two dimensions  
1090 on a 1-5 scale.  
10911092 **Assessment Criteria**1093 1. Clarity (Question and Answer) (1-5)  
1094 - 5: Concise, unambiguous wording; answer mirrors clarity  
1095 - 4: Minor wording issue but still unambiguous  
1096 - 3: Some vagueness but meaning recoverable  
1097 - 2: Ambiguities/redundancies hinder understanding  
1098 - 1: Unclear or contradictory wording  
1099 2. Correctness (vs. Ground-Truth) (1-5)  
1100 - 5: Answer matches all facts in chunks; nothing missing  
1101 - 4: Correct but one minor fact omitted/loosely paraphrased  
1102 - 3: At least half of facts correct; one factual slip  
1103 - 2: Major fact missing/misstated/unsupported  
1104 - 1: Contradicts or ignores ground truth1105 **Evaluation Process**1106 1. Identify reasoning process  
1107 2. Assess alignment between query and provided text chunk  
1108 3. Evaluate clarity of question and answer  
1109 4. Verify factual correctness against ground-truth chunk1110 **Input**1111 - query: The single-hop question  
1112 - answer: The provided answer  
1113 - text chunk: Source text chunk1114 **Output**1115 {  
1116 "score": <average\_of\_dimension\_scores>,  
1117 "dimension\_scores": {  
1118 "clarity": <1-5>,  
1119 "correctness": <1-5>  
1120 }  
1121 }  
1122 }  
1123 }1124  
1125  
1126  
1127

### Extracted Triplets Factual Correction Judge Prompt

You are a strict fact-checking assistant. Verify if the extracted knowledge triplet is factually correct and explicitly supported by the provided text.

**Text:**

- {text}

## Triplet:

Subject: {triplet['entity\_1']}

Relation: {triplet['relation']}

Object: {triplet['entity\_2']}

Is this triplet factually correct based ONLY on the text provided?

Answer with **ONLY** 'Correct' or 'Incorrect'. Do not provide any explanation.

## Multi-Hop QA Pairs Quality Assurance

## Multi-Hop Query Quality Evaluator

You are an expert evaluator of multi-hop queries. Assess each query's quality across three dimensions on a 1-5 scale.

## Assessment Criteria

- 1: Reasonableness and Multi-hop Need (1-5)
  - 5: Meaningful question requiring all hops; each hop justified
  - 4: Reasonable but one hop weakly motivated or could be merged
  - 3: Sensible but answerable by single chunk with assumptions
  - 2: Forced/trivial question; multi-hop structure unnecessary
  - 1: Nonsensical/irrelevant; multi-hop structure meaningless
- 2. Clarity (Question and Answer) (1-5)
  - 5: Concise, unambiguous wording; answer mirrors clarity
  - 4: Minor wording issue but still unambiguous
  - 3: Some vagueness but meaning recoverable
  - 2: Ambiguities/redundancies hinder understanding
  - 1: Unclear or contradictory wording
- 3. Correctness (vs. Ground-Truth) (1-5)
  - 5: Answer matches all facts in chunks; nothing missing
  - 4: Correct but one minor fact omitted/loosely paraphrased
  - 3: At least half of facts correct; one factual slip
  - 2: Major fact missing/misstated/unsupported
  - 1: Contradicts or ignores ground truth

## Evaluation Process

1. Identify distinct reasoning hops and assess necessity
2. Check alignment between hops and provided chunks
3. Evaluate clarity of question and answer
4. Verify factual correctness against ground-truth chunks

1175  
 1176     **Input**  
 1177     - query: The multi-hop question  
 1178     - answer: The provided answer  
 1179     - text chunks: Source text chunks  
 1180

1181     **Output**  
 1182     {  
 1183       "score": <average\_of\_dimension\_scores>,  
 1184       "dimension\_scores": {  
 1185         "reasonableness": <1-5>,  
 1186         "clarity": <1-5>,  
 1187         "correctness": <1-5>  
 1188       }  
 1189     }

1190  
 1191  
 1192

### RAG Simulation Prompt: RAG Generator

1193     You are a {domain} expert. You are given a {domain} question and one or multiple contexts.  
 1194     Your task is to answer the question strictly based on the these contexts.  
 1195     You should think step by step and answer the question in a detailed and comprehensive way. Please  
 1196     return the detailed reasoning process in the cot\_answer part.

1197  
 1198

#### Requirements:

- Your answer is short and concise, do not return any other text in the answer part.
- Example 1: "What is the United States' GDP in 2024?"
- Good: "\$31.1 trillion"
- Bad: "According to the context, as my knowledge, the answer is \$31.1 trillion"
- Example 2: "Who is the president of the United States from 2021 to 2025?"
- Good: "Joe Biden"
- Bad: "The president of the United States from 2021 to 2025 is Joe Biden, according to my knowledge"
- If the question is not related to the context, strictly return "no such info" for answer part. Do not return any other text in such case.

1199  
 1200

1201     Here are some examples of how to answer based on the given context:

#### *Example 1:*

1202     Question: What was Apple's revenue in Q2 2023?  
 1203     Context: [Doc] Apple Inc. reported financial results for its fiscal 2023 second quarter ended April 1, 2023. The Company posted quarterly revenue of \$94.8 billion, down 2.5 percent year over year.  
 1204     cot\_answer: The question asks about Apple's revenue in Q2 2023. According to the context, Apple  
 1205     reported quarterly revenue of \$94.8 billion for its fiscal 2023 second quarter ended April 1, 2023.  
 1206     This represents a decrease of 2.5 percent year over year.  
 1207     answer: \$94.8 billion

1208  
 1209  
 1210

1222

1223

*Example 2:*

1224

Question: What is Luxembourg's approach to public transport?

1225

Context: [Doc] On March 1, 2020, Luxembourg became the first country to make all public transport free, including buses, trains, and trams. This policy aims to reduce traffic congestion and carbon emissions while promoting sustainable mobility solutions across the country.

1226

cot\_answer: The question asks about Luxembourg's approach to public transport. According to the context, Luxembourg made all public transport free on March 1, 2020, becoming the first country to do so. This includes buses, trains, and trams. The goal of this policy is to reduce traffic congestion and carbon emissions while promoting sustainable mobility solutions.

1227

answer: Free public transport for all

1228

1229

1230

1231

1232

*Example 3:*

1233

Question: How many homeless individuals received emergency shelter services in Pittsburgh?

1234

Context: [Doc] The City of Pittsburgh allocated CDBG funds to various community programs including affordable housing initiatives. The HOME program supported the construction of 45 new housing units for low-income families.

1235

cot\_answer: The question asks about the number of homeless individuals who received emergency shelter services in Pittsburgh. After reviewing the context carefully, I don't see any information about emergency shelter services for homeless individuals or any numbers related to this. The context only mentions CDBG funds for community programs and the HOME program supporting 45 new housing units for low-income families. There is no specific information about homeless emergency shelter services.

1236

answer: no such info

1237

1238

*Example 4:*

1239

Question: What were Smith A O Corp's consolidated sales for the year ended December 31, 2024?

1240

Context: [Doc] In this section, we discuss the results of our operations for 2024 compared with 2023. Our sales in 2024 were \$3,818.1 million, a decrease of \$34.7 million compared to 2023 sales of \$3,852.8 million. Our decrease in net sales was primarily driven by lower water heater volumes in North America, lower sales in China, and unfavorable currency translation of approximately \$18 million due to the depreciation of foreign currencies compared to the U.S. dollar, which more than offset our higher boiler sales and pricing actions.

1241

cot\_answer: The question asks about Smith A O Corp's consolidated sales for the year ended December 31, 2024. According to the context, the sales in 2024 were \$3,818.1 million, which was a decrease of \$34.7 million compared to 2023 sales of \$3,852.8 million. The context explains that this decrease was primarily due to lower water heater volumes in North America, lower sales in China, and unfavorable currency translation of approximately \$18 million.

1242

answer: \$3,818.1 million

1243

1244

1245

**Output Format:**

1246

- cot\_answer: detailed reasoning process

1247

- answer: concise answer to the question

1248

1249

1250

1251

1252

1253

1254

1255

1256

1257

1258

1259

1260

1261

1262

1263

1264

1265

1266

1267

1268

### Evaluation Prompt: Judging Prediction and Ground Truth

You are a fair and strict judge, given prediction and ground truth, your task is to determine if the prediction has the same/highly similar meaning as the ground truth answer.

Return true if:

- The prediction and ground truth are semantically identical or highly similar.
- The prediction provides the same information as the ground truth.
- If ground truth is included in the prediction consider it a match.
- Which means if prediction not only contains the ground truth, but also contains other information, it should be considered a match.
- Example: prediction: "The company's revenue was \$50 million in 2023" and ground truth: "\$50 million" are considered the same.

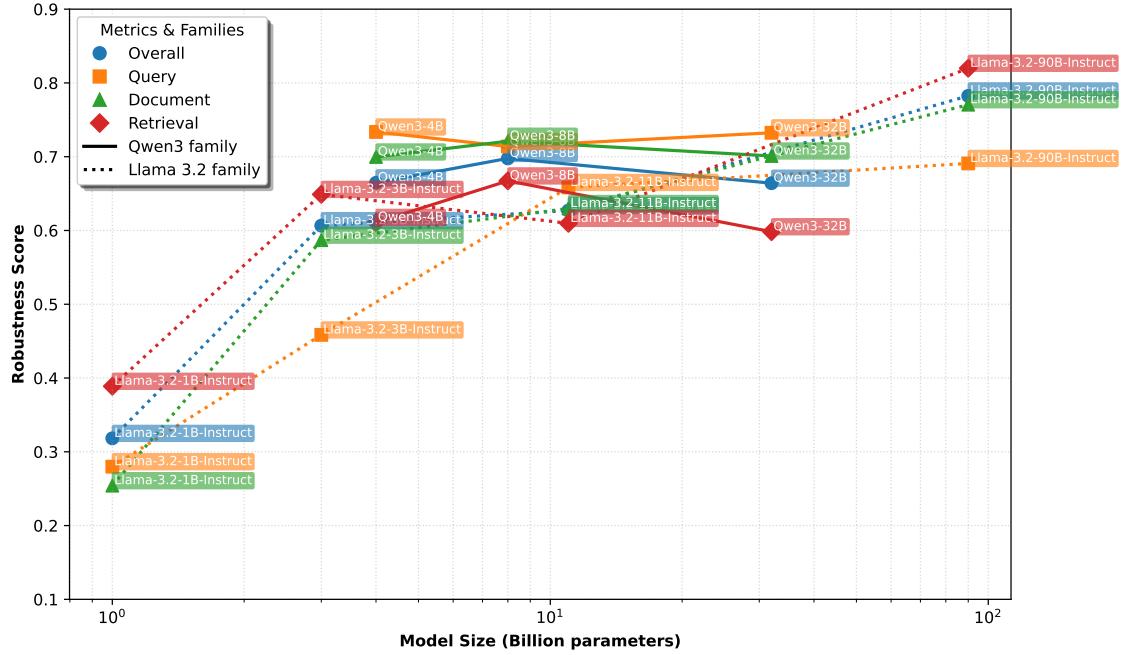
Return false if:

- The prediction does not match the ground truth in meaning.
- The prediction is a refusal or does not provide an answer.
- The ground truth has more specific information than the prediction.
- If the prediction is a numeric value, it should match the ground truth numerically
- Example 1: 120,000,000 and 120000000 are considered the same.
- Example 2: 120,000,000 and 120 billion are considered the same.

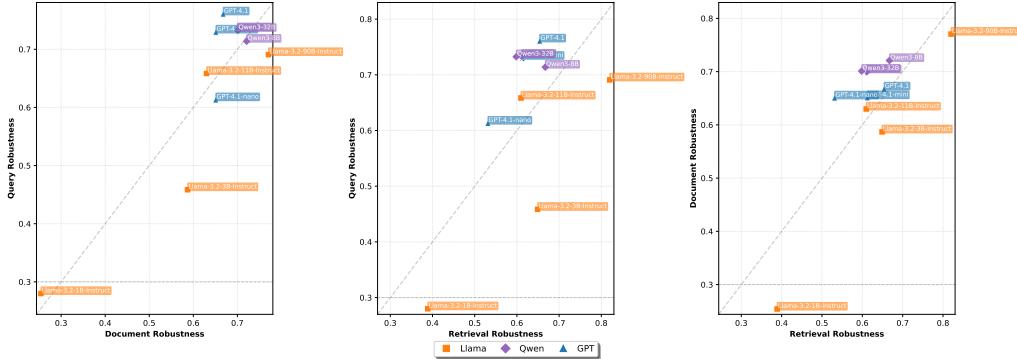
For your output, you should only answer 'true' or 'false', no extra text.

## Examples:

1. Prediction: "The company's revenue was \$50 million in 2023", Ground Truth: "\$50 million", Output: true
2. Prediction: "Apple Inc.", Ground Truth: "Apple", Output: true
3. Prediction: "I cannot find that information", Ground Truth: "25%", Output: false
4. Prediction: "The answer is 42", Ground Truth: "42", Output: true
5. Prediction: "The population is around 1 million", Ground Truth: "1,000,000", Output: true
6. Prediction: "Tesla", Ground Truth: "General Motors", Output: false

1316 **G EXPERIMENT RESULTS ANALYSIS**1317 **G.1 DETAIL ANALYSIS**

1342 Figure 4: Relationship between the sizes of open-source generators and their robustness scores across various  
 1343 categories. Generally, larger generator sizes correspond to higher robustness scores. However, for Qwen 3  
 1344 models, robustness scores tend to stay closely across difference parameter sizes



1358 Figure 5: Pairwise relationship between query, document and retrieval robustness. All of these models achieve  
 1359 the balanced robustness across query, document, and retrieval dimensions, while Qwen3 models cluster tightly  
 1360 in the upper-right corner, indicating consistently strong robustness across categories. In contrast, Llama  
 1361 models are more spread out, with smaller ones performing poorly and larger ones improving in document and  
 1362 retrieval robustness but still lagging in query robustness.

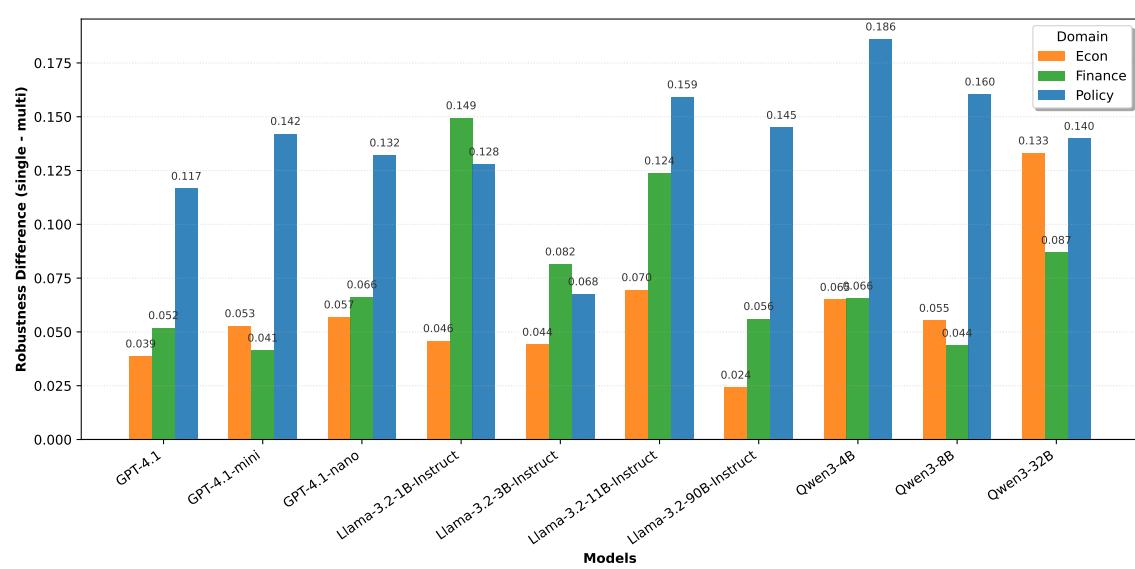


Figure 6: Difference in multi-hop and single-hop robustness scores by domain. Positive robustness scores = single-hop better, negative robustness scores = multi-hop better. Since all of the differences are positive, it clearly shows that RAG systems exhibit lower robustness on multi-hop questions compared to single-hop questions, while the most significant gaps appears in policy domain.

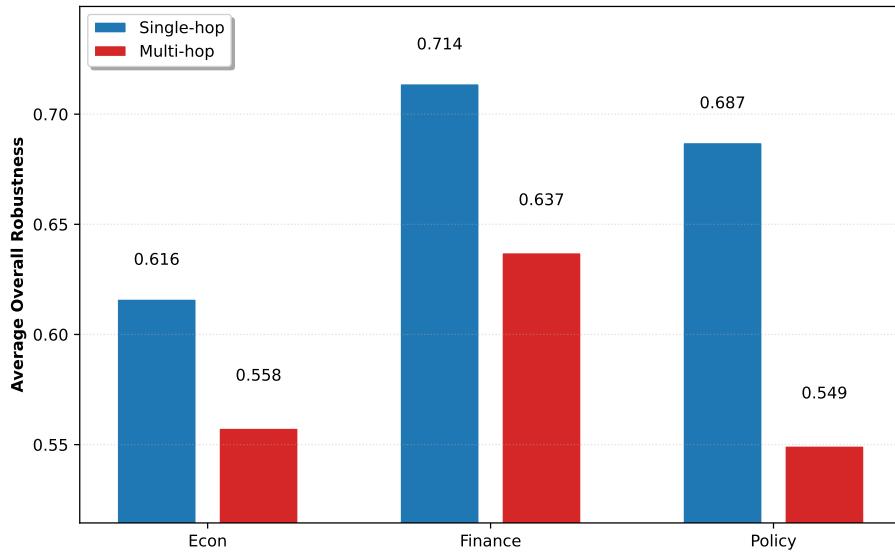
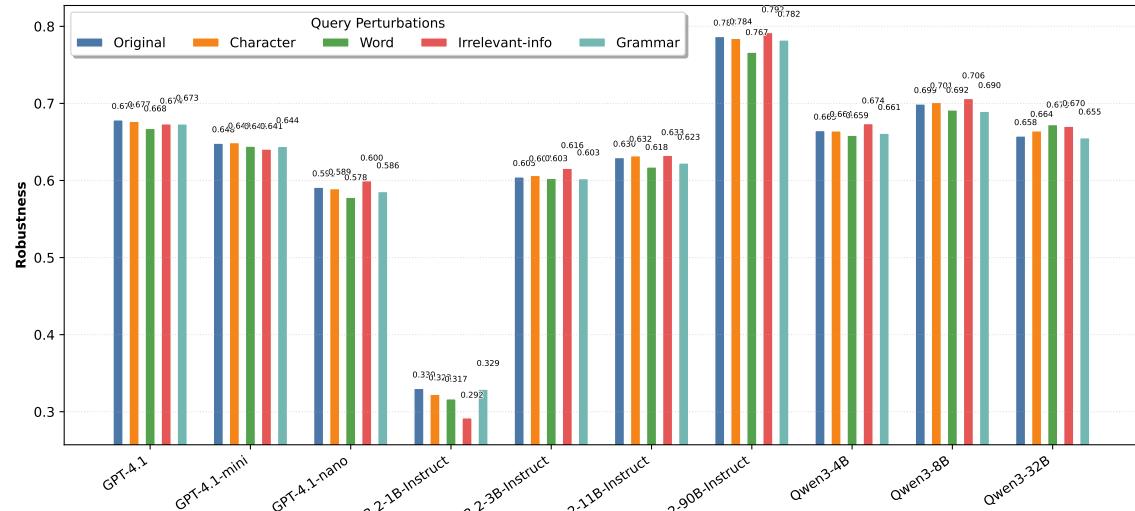
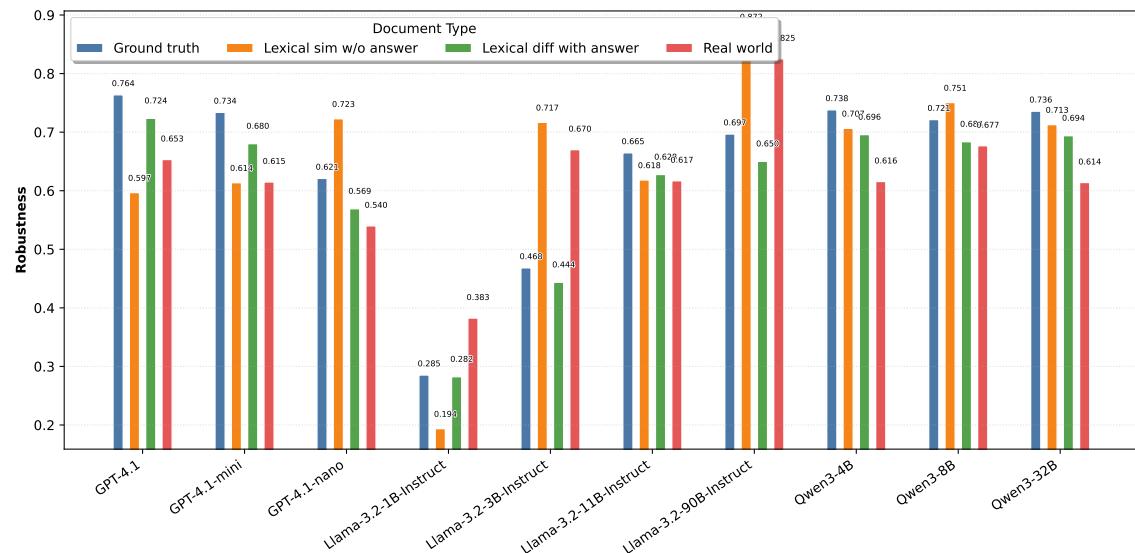


Figure 7: Average overall robustness scores from different domains and question types

1410 G.2 PERTURBATION-SPECIFIC ROBUSTNESS  
1411  
14121413 Figure 8: Average robustness score in different query perturbations vs. all types of documents.  
14141429 Figure 9: Average robustness score in different document perturbations vs. all types of queries.  
1430  
1431  
1432  
1433  
1434  
1435  
1436

1457 G.3 SIGNIFICANCE TESTS  
1458

1459	1460	Model 1	Model 2	Z-score	P-value		
1461	1462	<b>Qwen3 vs GPT</b>					
1462	1463	GPT-4.1	Qwen3-32B	6.312	0		
1463	1464	GPT-4.1	Qwen3-4B	5.987	0		
1464	1465	GPT-4.1	Qwen3-8B	-15.300	0		
1465	1466	Qwen3-32B	GPT-4.1-mini	11.772	0		
1466	1467	Qwen3-32B	GPT-4.1-nano	46.655	0		
1467	1468	Qwen3-4B	GPT-4.1-mini	12.096	0		
1468	1469	Qwen3-4B	GPT-4.1-nano	46.978	0		
1469	1470	Qwen3-8B	GPT-4.1-mini	33.362	0		
1470	1471	Qwen3-8B	GPT-4.1-nano	68.138	0		
1471	1472	<b>Qwen3 vs Llama</b>					
1472	1473	Llama-3.2-90B	Qwen3-32B	79.336	0		
1473	1474	Llama-3.2-90B	Qwen3-4B	79.016	0		
1474	1475	Llama-3.2-90B	Qwen3-8B	57.961	0		
1475	1476	Qwen3-32B	Llama-3.2-11B	23.295	0		
1476	1477	Qwen3-4B	Llama-3.2-11B	23.620	0		
1477	1478	Qwen3-8B	Llama-3.2-11B	44.860	0		
1478	1479	<b>GPT vs Llama</b>					
1479	1480	GPT-4.1	Llama-3.2-11B	29.599	0		
1480	1481	Llama-3.2-11B	GPT-4.1-mini	-11.529	0		
1481	1482	Llama-3.2-11B	GPT-4.1-nano	23.402	0		
1482	1483	Llama-3.2-90B	GPT-4.1	73.107	0		
1483	1484	Llama-3.2-90B	GPT-4.1-mini	90.924	0		
1484	1485	Llama-3.2-90B	GPT-4.1-nano	125.065	0		

Table 8: Pairwise two-proportion z-tests comparing models’ overall robustness scores. Every pair of p-value is less than 0.05, indicating our results’ high statistical significances.

1487 H THE USE OF LLMs  
14881489 We acknowledge the use of LLMs in the writing of this paper. They were used to check grammar and improve  
1490 sentence clarity. In addition, LLMs were utilized in our data generation pipeline and during the evaluation  
1491 stage. These uses are explicitly described in the corresponding sections.  
1492