Boosting RAG Performance with Noise-Infused Retrieval Strategies

TEAM:

- Sandhya Lotla
- Sindhu Gajulapalli
- Mallikarjuna Bandi

Project Topic :
Investigating the Role of Contextual Noise and Distractors in Retrieval-Augmented Generation Systems

Our Project Focus on:

- Designing and evaluating prompt compositions that mix gold/relevant documents with noise (random, unrelated text).
- Analysing how placement, quantity, and entropy of documents impact LLM accuracy.
- Comparing performance across retrievers (Contriever, BM25) and content types (Wikipedia, Reddit, synthetic).

Domain / Corpus:

Open-Domain Question Answering (QA) on Wikipedia (NQ dataset)

Dataset: Natural Questions (NQ-Open) — real-world questions answered from Wikipedia.

Corpus:

Wikipedia dumps - test_dataset.json and 10k_train_dataset.json

Synthetic noise datasets like:

- corpus_with_random_50_words.pkl – randomly generated word sequences
- Reddit-based noise (10k_random_results_at60.pkl)
- ADORE retriever outputs (diverse noisy responses)

Project Objectives

Evaluate

- Evaluate the impact of document relevance
  - Examine how different types of retrieved documents (gold, relevant, distracting, random) influence the accuracy of LLM responses in a RAG pipeline.

Test

- Test noise as a performance enhancer
  - Investigate whether adding noise—such as random or unrelated documents—can improve answer quality by reducing attention collapse in LLMs.

Analyse

- Analyse prompt structure and document positioning
  - Determine how the position of documents in the prompt affects attention allocation and accuracy.

Benchmark

- Benchmark different retrievers
  - Compare performance across retrievers like Contriever, BM25, and Adore to assess their effectiveness for RAG-style QA tasks.

Propose

- Propose retrieval heuristics for RAG systems
  - Develop actionable design strategies for prompt composition using a mix of relevant and noisy documents.

□Why This Project Is Worth Doing:

□1. Challenges Existing Assumptions
Most RAG systems assume that more relevant documents lead to better answers.
This project disproves that notion and shows that irrelevant or random documents can sometimes improve accuracy.

□2. Performance Gains with Simple Techniques
Noise-infused prompts improve model performance without needing additional training or retriever tuning.
Boosts accuracy by up to +36% in some models with simple prompt reshaping.

□3. Real-World Impact
Helps optimize enterprise RAG applications like chatbots, document summarization, and search assistants.
Saves compute and infrastructure costs by retrieving fewer high-quality documents plus noise.

□4. Improves Fairness & Robustness
Reduces model over-reliance on misleading context, making it more trustworthy.
Enhances generalization and reduces hallucination in diverse real-world settings.

□5. Adds Novelty to RAG Research
Introduces a new way of thinking: "noise as a feature, not a bug."
Contributes unique insights into prompt design, retrieval strategy, and LLM

Review of the State of the Art in RAG Systems

Retrieval-Augmented Generation (RAG): Overview

- Combines dense/sparse retrievers with large language models (LLMs)
- Popularized by Lewis et al., 2020 (Facebook AI) with the DPR + BART model
- RAG systems reduce hallucination by grounding LLMs in external knowledge

Key Developments:

- DPR (Dense Passage Retrieval) – Lewis et al., 2020
- REALM – Google Research (Guu et al., 2020): Joint retrieval and generation
- FiD (Fusion-in-Decoder) – Izacard & Grave, 2020: Late fusion for better accuracy
- RETRO (DeepMind) – Borgeaud et al., 2021: External memory + retrieval for training-free augmentation

Common Challenges in RAG:

- Distracting documents mislead the generator
- Retrieval ranking vs. generation utility mismatch
- Context overflow in long prompts reduces attention to useful content

Citations:

1. Cuconasu et al., 2024: "The Power of Noise: Redefining Retrieval for RAG Systems" https://arxiv.org/abs/2401.14887
2. Borgeaud et al., 2021: https://arxiv.org/abs/2112.04426
3. Lewis et al., 2020: https://arxiv.org/abs/2005.11401
4. Guu et al., 2020: https://arxiv.org/abs/2002.08909

Review of the State of the Art in RAG Systems

Limitations of Existing RAG Research

- Focus is mainly on retriever performance (recall/relevance), not prompt composition
- Most models assume "more relevant context = better results“

Paper Contributions of This Project (Inspired by Cuconasu et al., 2024)

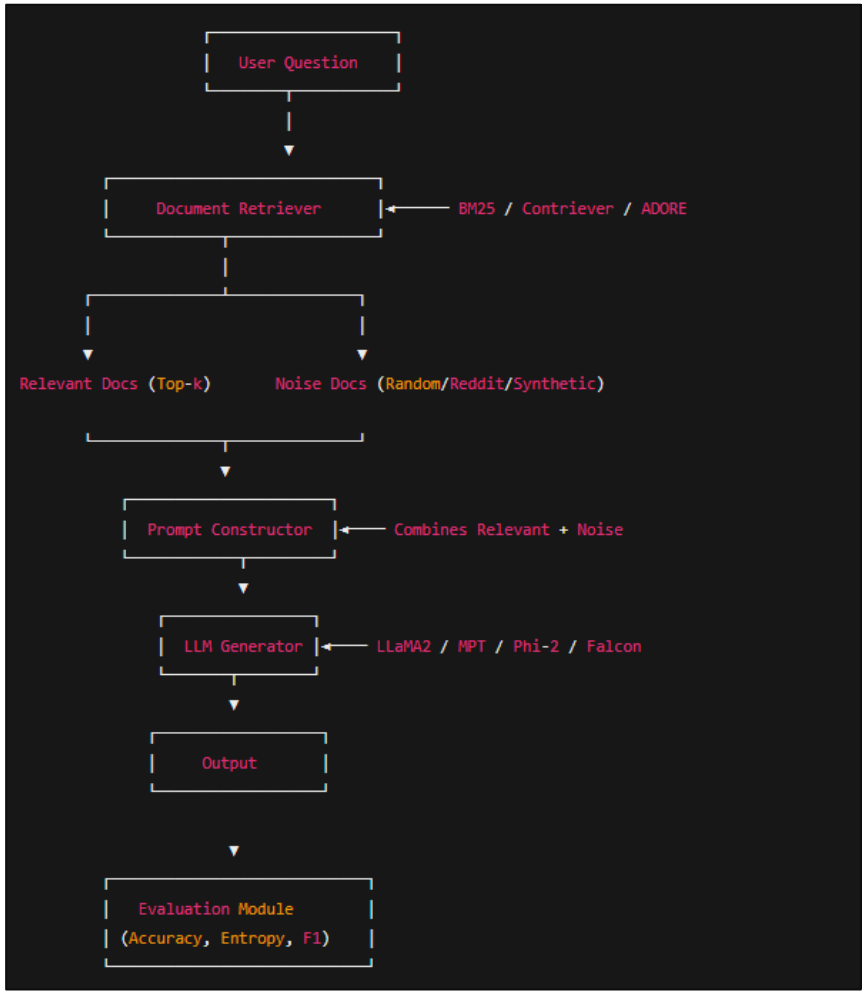
- Shows that random noise can improve accuracy more than irrelevant-but-related content
- Emphasizes document ordering and attention entropy as key factors
- Introduces concept of "entropy collapse" in RAG prompt

Core Citation:

Cuconasu, Trappolini, Siciliano, Filice, Campagnano, Maarek, Tonellotto, Silvestri. "The Power of Noise: Redefining Retrieval for RAG Systems." arXiv preprint arXiv:2401.14887, 2024.
https://paperswithcode.com/paper/the-power-of-noise-redefining-retrieval-foring

Proposed Approach

Build a **Retrieval-Augmented Generation (RAG)** system that challenges traditional relevance-based document selection. Introduce **noise-infused prompting strategies** based on the findings of Cuconasu et al. (2024).



LLMs to Be Used (Open Source)

1. LLaMA 2 (7B or 13B, 4-bit quantized) – Meta AI
2. MPT – MosaicML
3. Phi-2 – Microsoft
4. Falcon (7B) – Technology Innovation Institute (TII)

Datasets & Knowledge Base

Natural Questions (NQ-Open) – for question-answer pairs.
Wikipedia corpus (psgs_w100.tsv) – for dense/sparse retrieval.
Noise corpora:

- Random passages
- Reddit-based snippets
- 50-word synthetic sequences (from GitHub repo)

System Architecture

1. Retriever (BM25, Contriever, Adore)
2. Prompt Constructor (Mix of relevant + noise documents)
3. LLM Generator (e.g., LLaMA2, Phi-2)
4. Evaluation Module (binary accuracy, entropy attention maps)

Tools & Frameworks

- FAISS or ScaNN for retrieval index
- HuggingFace Transformers + Bitsandbytes for LLMs
- PyTorch / CUDA for model execution
- Attention heatmap tools (e.g., LIT or Captum)

Project Deliverables

1. Codebase for a Noise-Infused RAG Pipeline

- Complete RAG framework with modular scripts for embedding, retrieval, prompt generation, LLM inference, and evaluation.
- Includes support for multiple retrievers (Contriever, BM25, ADORE) and LLMs (LLaMA 2, MPT, Phi-2).
- Reproducible using argparse, clean architecture, and logging.

2. Dataset Variants & Retrieval Outputs

- Processed datasets: 10k_train_dataset.json, test_dataset.json
- Retrieval outputs from:
- Relevant retrievers (contriever, bm25)
- Noise sources (random, reddit, nonsense)
- Mixed retrieval results stored as .pkl for reproducible evaluation

3. Prompt Evaluation Results

- Generated answer logs with:
- Binary accuracy metrics
- Normalized string match evaluation
- Presence/absence of gold context
- Includes “only-query” (closed-book), “classic”, “mixed”, and “multi-corpus” prompts

4. Final Report & Presentation

- Summary of experiments and findings
- PPT deck with:
- Problem Statement, Objectives, State of the Art
- Approach, Evaluation, and Deliverables
- All code, data, and results shared via GitHub repository

Evaluation Methodology

Evaluation Focus

- Assess the effectiveness of different retrieval strategies (gold, random, distracting) in Retrieval-Augmented Generation (RAG).
- Measure LLM output accuracy when exposed to various prompt compositions.
- Analyze attention behavior and document placement effects.

Evaluation Process

1.Answer Generation
Prompts are constructed via PromptDataset, MixedDocumentsDataset, etc. based on query + context documents.
LLM (e.g., LLaMA2, Phi-2) generates short-form answer (max 5 tokens).

2.Answer Comparison
Normalize both LLM output and gold answers.
Check if any gold answer appears in normalized output.

3.Context-Aware Evaluation
Confirm if correct answer was actually present in any of the docs (grounded response).
Identify presence of gold doc using document_indices.

4.Batch Reporting
Results aggregated using read_generation_results.py into JSON & DataFrame
Accuracy printed and logged across prompt types (classic, mixed, multi-corpus, only-query)

Key Evaluation Metrics

Metric	Purpose
Answer Match (after norm)	Checks if generated answer matches any ground truth after normalization
Answer In Prompt	Confirms if the answer exists in any of the retrieved documents
Gold Document Retrieved	Verifies if the gold document is included in the retrieved context
Prompt Token Length	Tracks input size to evaluate model saturation or truncation
Accuracy	correct predictions / total queries — primary performance indicator