CentraleSupélec | université PARIS-SACLAY

CRÉDIT AGRICOLE CORPORATE & INVESTMENT BANK

Explainable Large Language Models for RAG Systems in Financial Applications

Ibrahim Al Khalil RIDENE

Kerrian LE CAILLEC

Lydia HAMMACHE

Table of contents

I. Introduction

II. Methodology

III. Retrieval Evaluation

IV. Generation Evaluation

V. Explainability

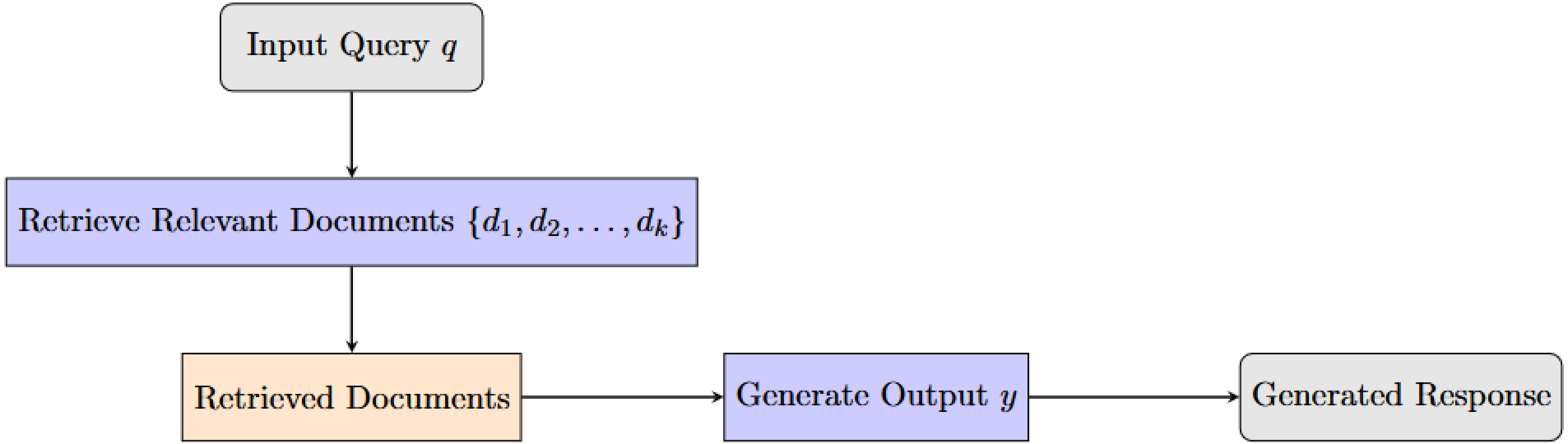
VI. Pipeline

VII. Appendix

I. Introduction

- Project in collaboration with AI Factory of CACIB
- LLMs used by professionals in investment banking (chatbots, text generation,…)
- **Problem :** How to ensure that an LLM's response is coherent and meets certain criteria ?
- Stake : To facilitate the tasks of CACIB's market finance professionals by providing LLMs that can be used as financial assistants, capable of delivering quick, reliable, and coherent responses.

RAG (Retrieval-Augmented Generation) architecture



Retrieval Module: Identifies the top-k documents from a corpus using a similarity metric.

Generation Module: Produces the response by conditioning on both the query and the retrieved documents.

Explainable AI

- **XAI:** methods designed to make AI predictions more understandable and interpretable.

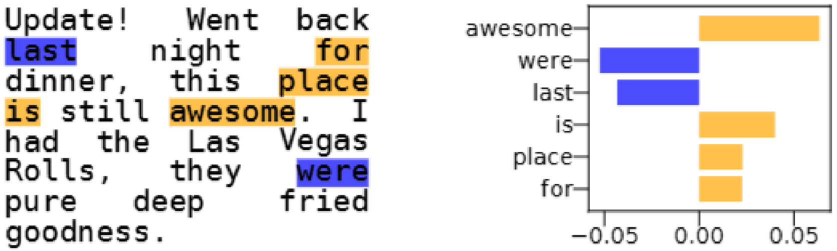
- **Methods**
  - Feature attribution
  - Counterfactuals explanation
  - Adversarial attacks

- **Problem**
  - How do they apply in the context of language models and RAGs ?

State of the art

- **Feature attribution methods**
  - LIME / SHAP(ley coefficients)
  - Integrated gradients

  ⇒ Not adapted to text-gen tasks

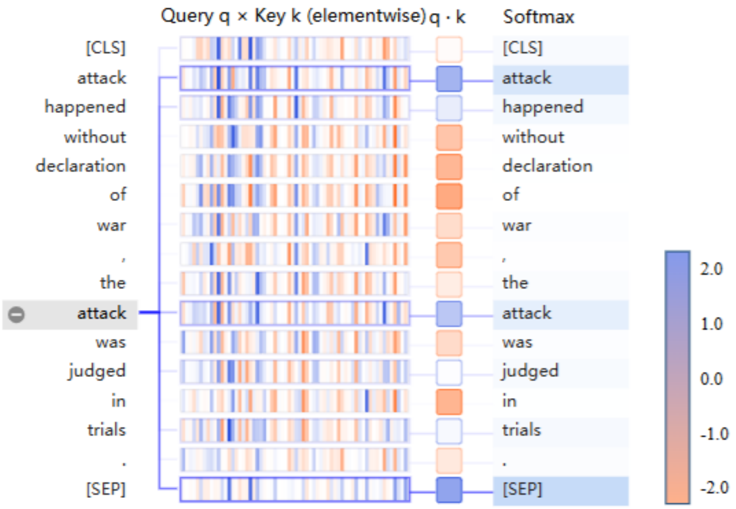


Update! Went back last night for dinner, this place is still awesome. I had the Las Vegas Rolls, they were pure deep fried goodness.

Mardaoui et al. [arXiv:2010.12487v]

- **Internal state dependent methods**

⇒ Requires access to the transformer Internal weights



Xingyu et al. [arXiv:2305.11498v]

II. Methodology

- **Project Overview:** Analyze a RAG system emphasizing transparency and explainability and design a pipeline for better clarity.

- **2 Main Axes:**

  - Retrieval Evaluation: “Why such documents have been retrieved?”
  - Evaluate the ability of various retrievers to return relevant and informative documents.

  - Generation Explainability: “Why did the model answer in this manner?”
  - Use LLM to answer finance-related questions based on the retrieved documents.
  - Analyze how each retrieved document contributes to the final answer.
  - Test the robustness of generated outputs (ablation testing).
  - Use a second LLM as a judge to detect hallucinations.

- Each RAG pipeline component is designed with explainability in mind, enabling performance measurement and justification of each response's reasoning

III. Retrieval Evaluation

- **Foundational Role of Retrieval in RAG**:
  - Directly impacts response quality, factuality, and trustworthiness.
  - Accurate retrieval grounds generated answers in relevant, contextual information.

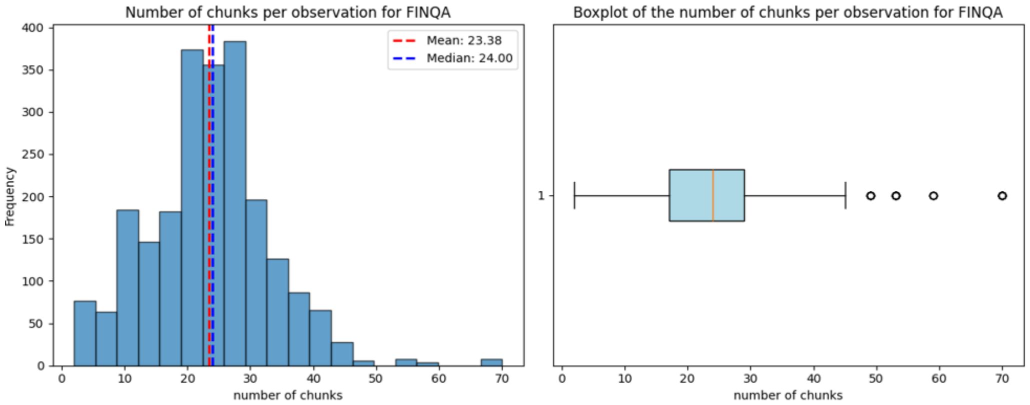
- **Focus of our Evaluation**:

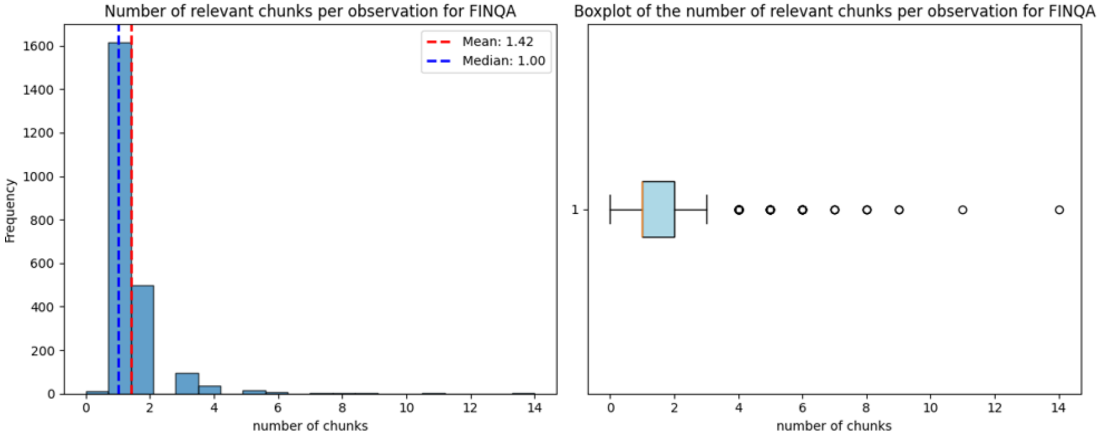
Evaluate retrieval strategies—models, embeddings, and similarity search—to identify optimal configurations in order to improve financial Q&A systems.

Datasets from RAG-Bench benchmark

FinQA : emphasizes deep reasoning over financial reports, including both textual and tabular data, and requires high financial literacy.

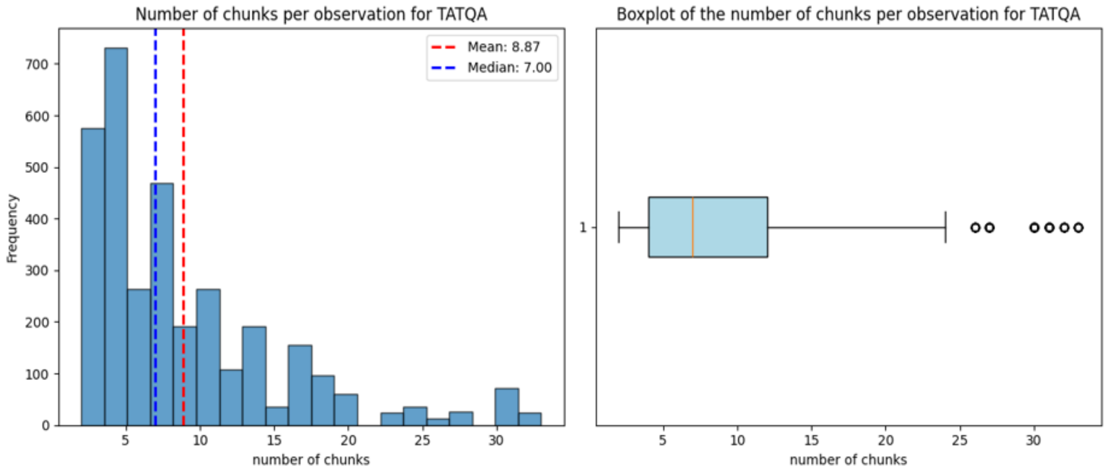
Distribution of total (left) and relevant (right) chunks per observation :

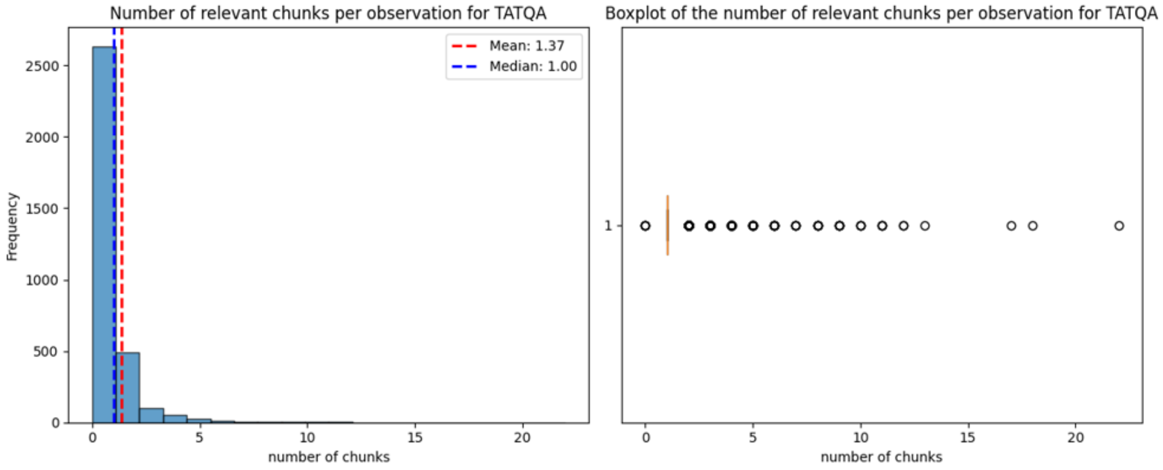




TAT-QA : involves broader domains and focuses on complex numerical reasoning over semi-structured tables.

Distribution of total (left) and relevant (right) chunks per observation :





- In both cases : need for an effective top-k retrieval.
- k = 5 ensures that relevant context is captured without introducing excessive noise.

Retrieval models

- **E5** : trained with weak supervision using contrastive learning. Optimized for zero-shot retrieval tasks, where positive pairs are selected based on top-ranked search engine results and negative pairs from low-ranked results.

- **Contriever** : trained using unsupervised contrastive learning. It builds sentence embeddings from adjacent text segments as positive pairs and random segments as negative ones. Its domain-agnostic nature allows general-purpose retrieval.

- **FinBERT** : A BERT-based model fine-tuned on large-scale financial corpora. It has shown strong performance on finance-specific tasks such as sentiment analysis, named entity recognition, and classification.

- **FinGPT** : A domain-specific language model tailored for financial tasks, including question answering, summarization of reports, and retrieval-augmented generation. It integrates a retrieval component into its fine-tuning process for better factual grounding.

Metrics used for retrieval model selection

- **Recall@K:** Measures the proportion of relevant chunks successfully retrieved among the ground-truth relevant chunks.

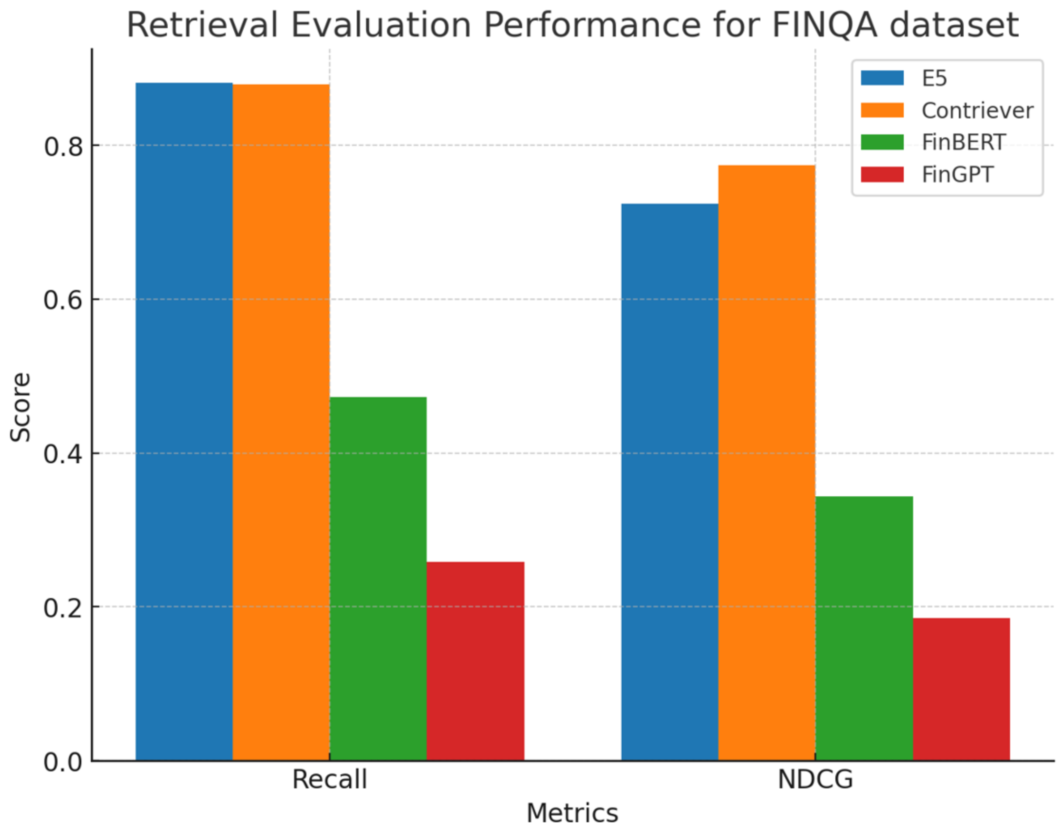
$$\text{Recall@K} = \frac{\text{Number of Relevant Retrieved Chunks in Top } K}{\text{Total Number of Ground-Truth Relevant Chunks}}$$

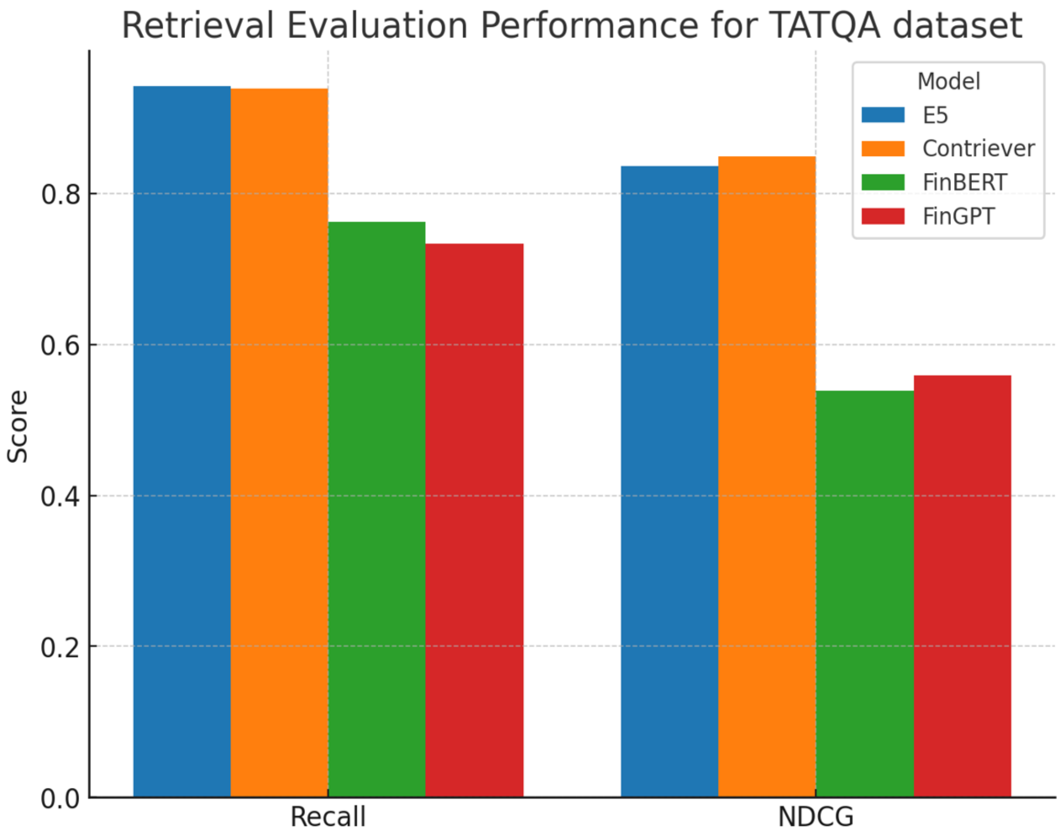
- **NDCG@K** (Normalized Discounted Cumulative Gain): Evaluates the ranking quality of the retrieved results by assigning higher scores to relevant documents that appear earlier in the ranked list.

$$\text{DCG}_K = \sum_{i=1}^{K} \frac{\text{Relevance}_i}{\log_2(i+1)}, \quad \text{NDCG}_K = \frac{\text{DCG}_K}{\text{Ideal DCG}_K}$$

Captures both the presence of relevant documents and their position in the top-k results

Comparison results





Retrieval model selected : Contriever

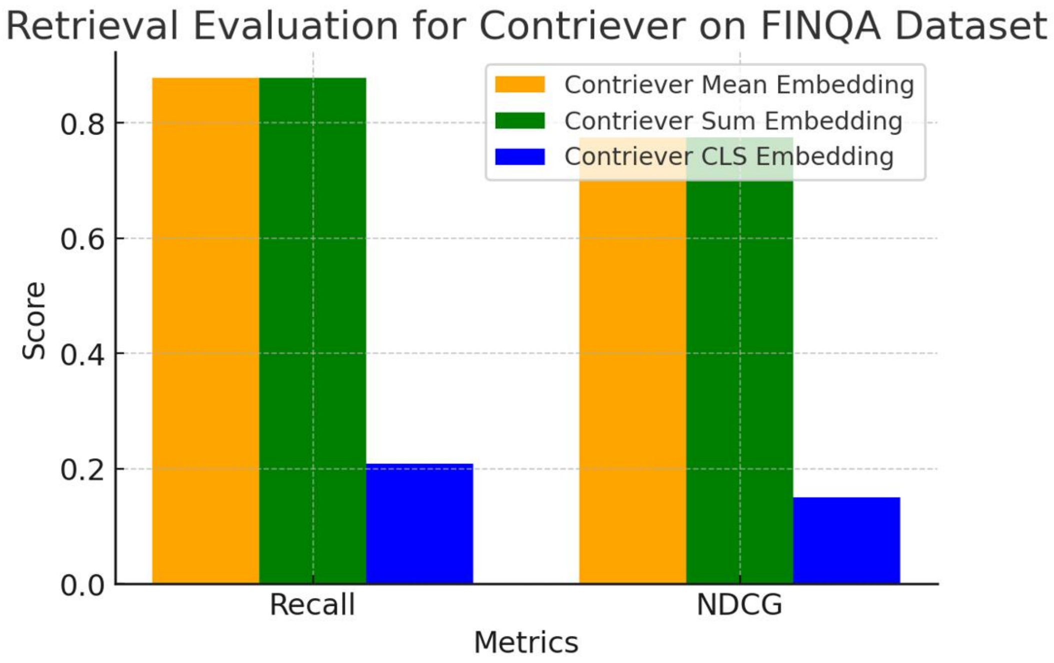
Embedding aggregation strategy

- **Mean Embedding**: The average over all token embeddings in the output.

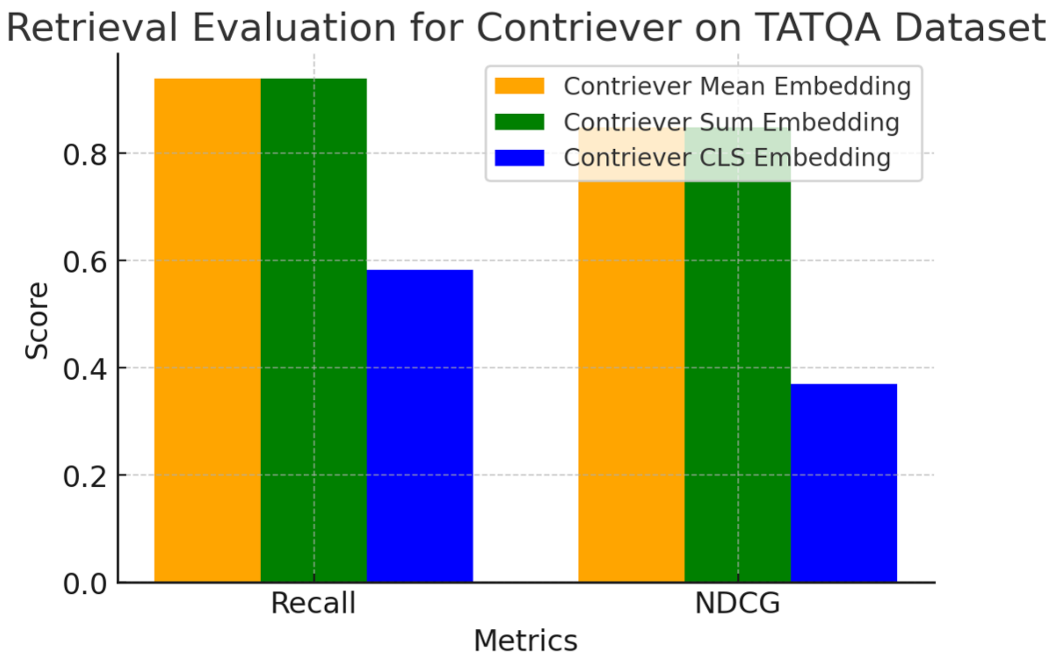
- **Sum Embedding**: The sum of all token embeddings.

- **CLS Token Embedding**: The embedding corresponding to the special classification token [CLS]

Comparison results



Retrieval Evaluation for Contriever on FINQA Dataset



Retrieval Evaluation for Contriever on TATQA Dataset

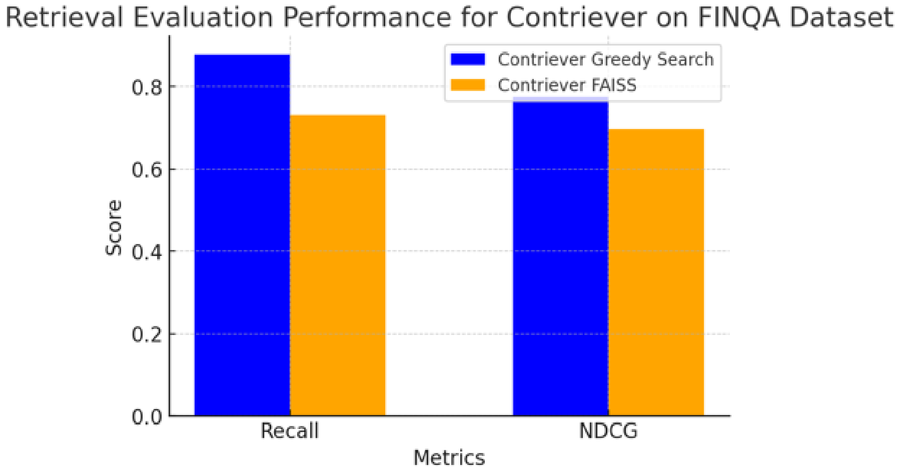
Strategy selected : Mean Embedding

Search method

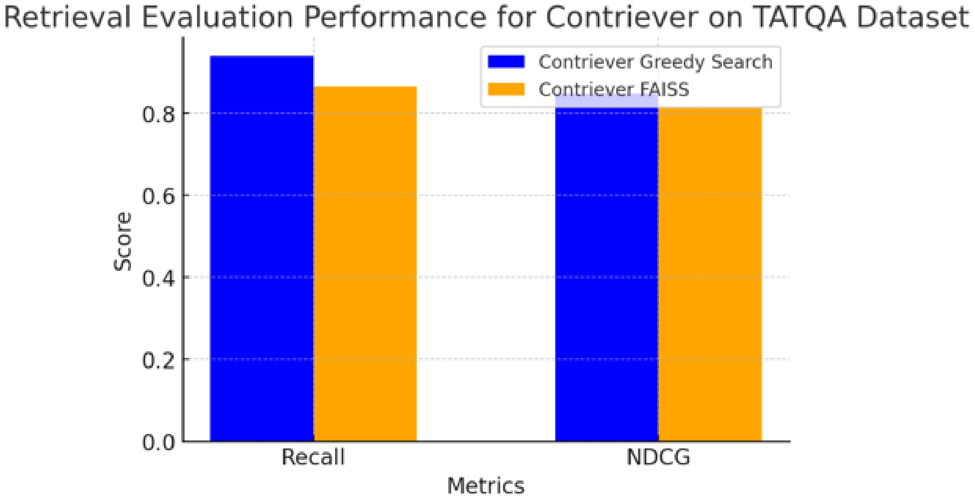
- Comparison of two document retrieval strategies for the Contriever retrieval model:

  - brute force search : standard cosine similarity

  - FAISS (Facebook AI Similarity Search) : an approximate nearest neighbor (ANN) method

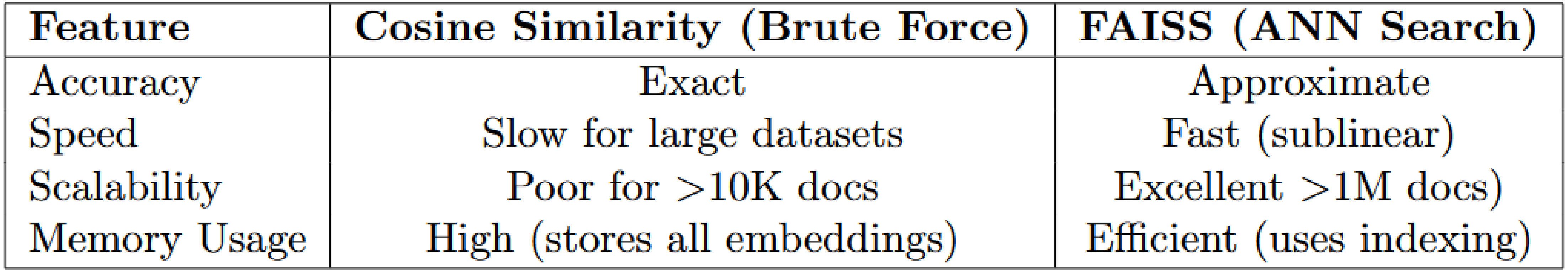


Brute force: 134.15s
FAISS: 133.97s



Brute force : 129.72s
FAISS: 130.30s

Comparison of brute force vs FAISS for retrieval



Feature	**Cosine Similarity (Brute Force)**	**FAISS (ANN Search)**
Accuracy	Exact	Approximate
Speed	Slow for large datasets	Fast (sublinear)
Scalability	Poor for >10K docs	Excellent >1M docs)
Memory Usage	High (stores all embeddings)	Efficient (uses indexing)

Strategy selected : Brute force search

Cosine similarity limitation

- **Cosine Similarity** : Computes a score purely based on angular distance between the query embedding and each chunk embedding. It selects top-k chunks with highest cosine scores

$$\text{cosine\_sim}(q, d_i) = \frac{\langle q, d_i \rangle}{\|q\| \|d_i\|}$$

- **Limitation** when ranking document chunks based on their semantic closeness to the query : it does not promote diversity among the retrieved documents.

- **Consequence** :Multiple top-k chunks may be semantically redundant or clustered around similar content

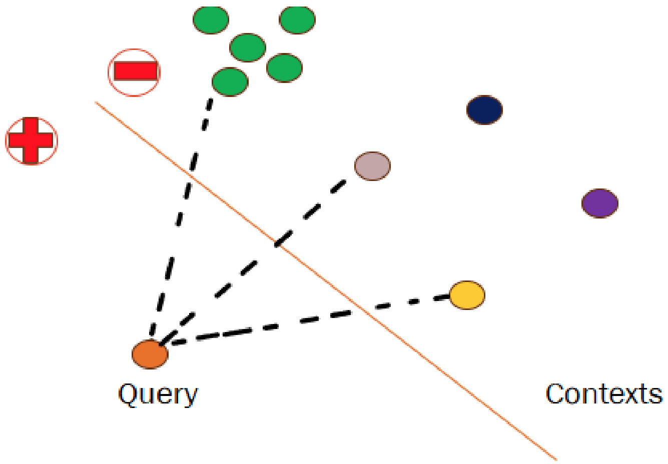
Linear SVM classifier as a reranking mechanism

- **Goal :** identify diverse yet relevant passages
- SVM-Based Retrieval trains a linear classifier with:

  -Positive class: the query embedding

  -Negative class: all chunk embeddings

- **Balance between similarity and diversity** : selection of chunks near the decision boundary (smallest absolute margin). Those are close to the query but also spread across the feature space.



IV. Generation Evaluation

- **Input for generation :** combines the user query and retrieved context to synthesize a coherent, informative answer.

- **Focus of our Evaluation :**
  - Assess the quality and reliability of the generated outputs, emphasizing factual consistency and explainability.
  - Understand how the generation process depends on the input and assess its alignment with the underlying context.

- **Twofold Evaluation Strategy:**
  - Ablation Testing
  - Large Language Model as Judge

Ablation testing

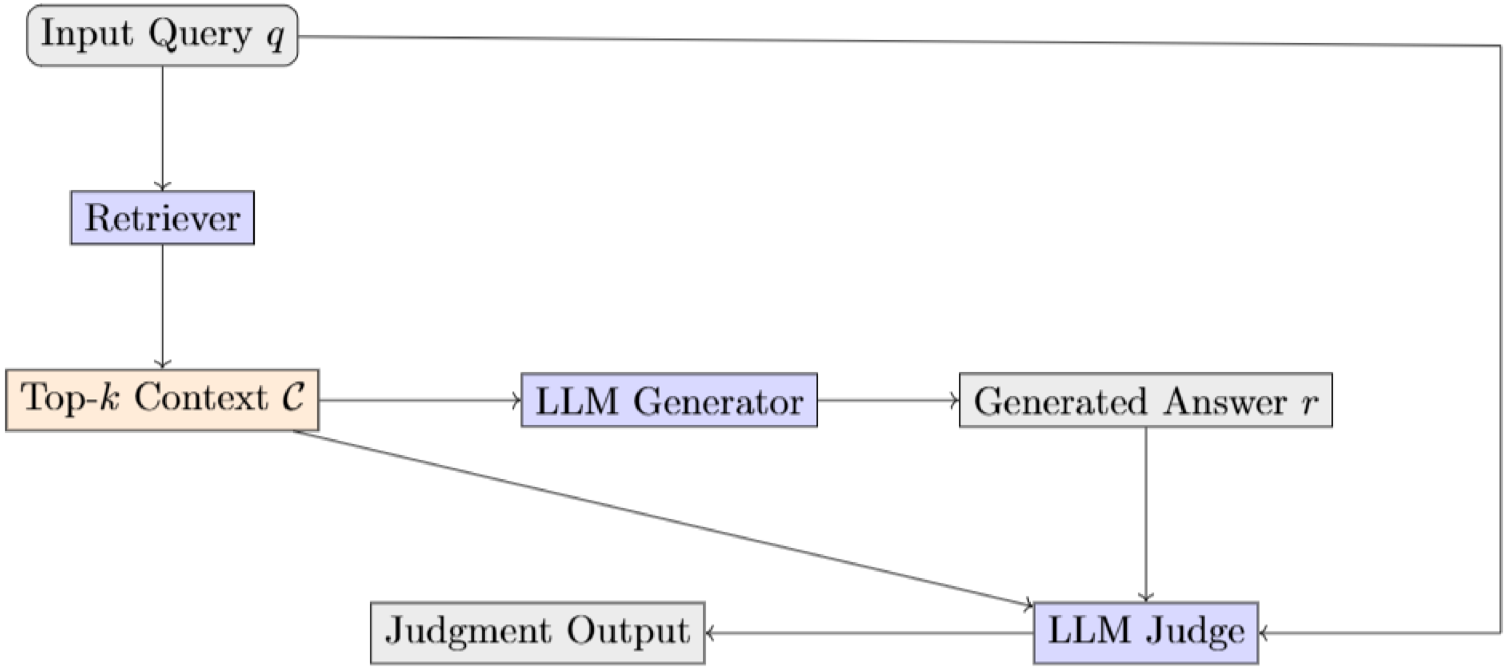
- **Objective** : Assess the influence of each retrieved context chunk on the final generated answer.
- **Method**:
  - Remove one context chunk at a time and regenerate the response for the same query.
  - Compare the reduced-context response to the original full-context response.
- **Notation**:
  - Original question: q
  - Retrieved context chunks: C={c1,c2,…,ck}
  - Original response: r=LLM(q,C)
  - Reduced context (without ci): C−i=C\{ci}
  - Reduced response: r−i=LLM(q,C−i)r−i =LLM(q,C−i).
- **Compute semantic similarity** using cosine similarity between sentence embeddings (e.g., SBERT).

$$\mathrm{cosine\_sim}(r, r_{-i}) = \frac{\langle \phi(r), \phi(r_{-i}) \rangle}{\|\phi(r)\| \|\phi(r_{-i})\|}$$

- Chunks whose removal causes a significant drop in similarity are deemed more important.

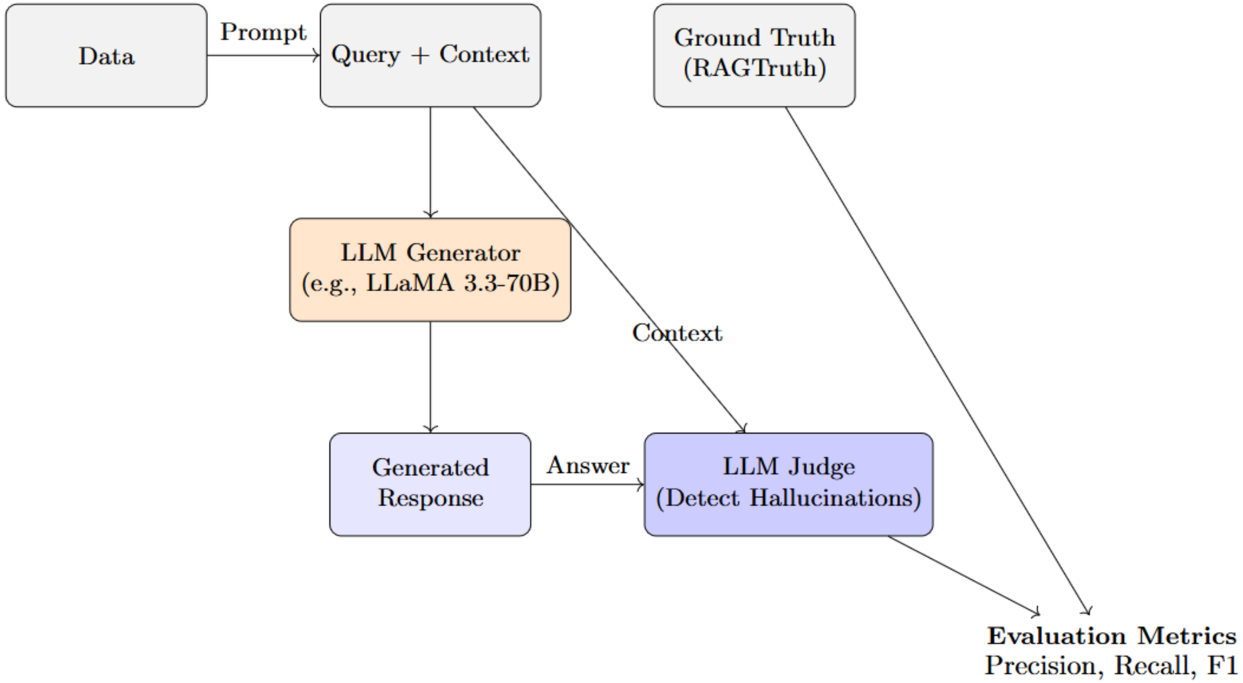
LLM as a Judge

- Use of a second LLM as an external judge to assess the factual correctness and trustworthiness of generated responses
- LLM as a Judge integrated within a RAG pipeline to evaluate factual consistency:

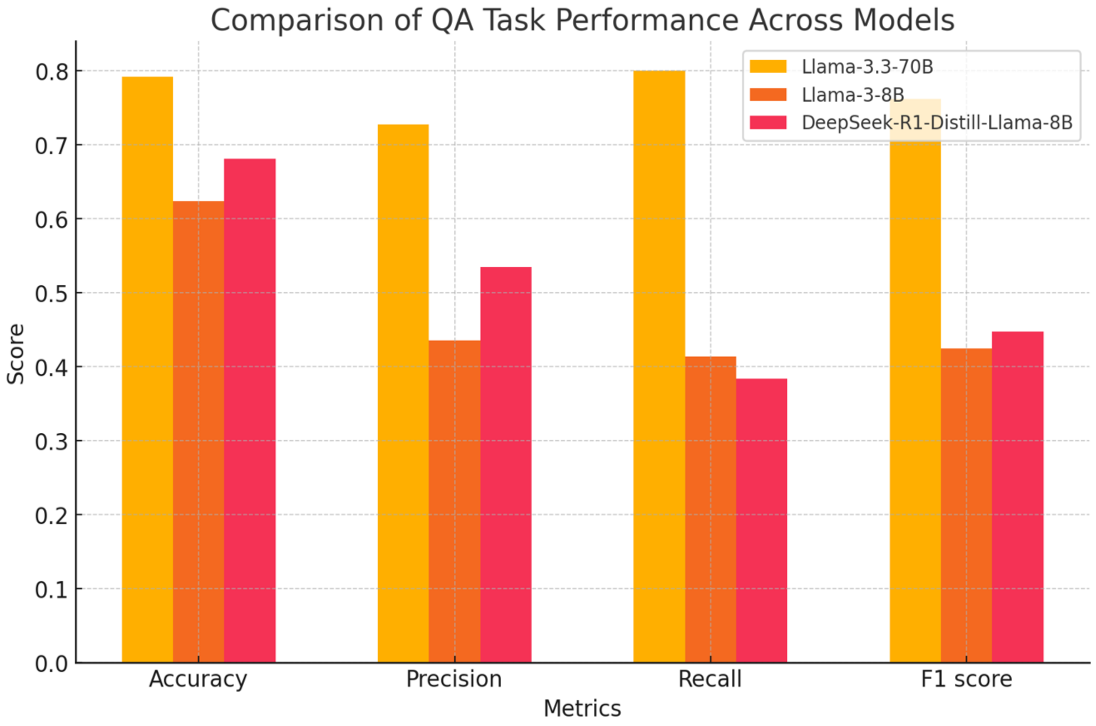


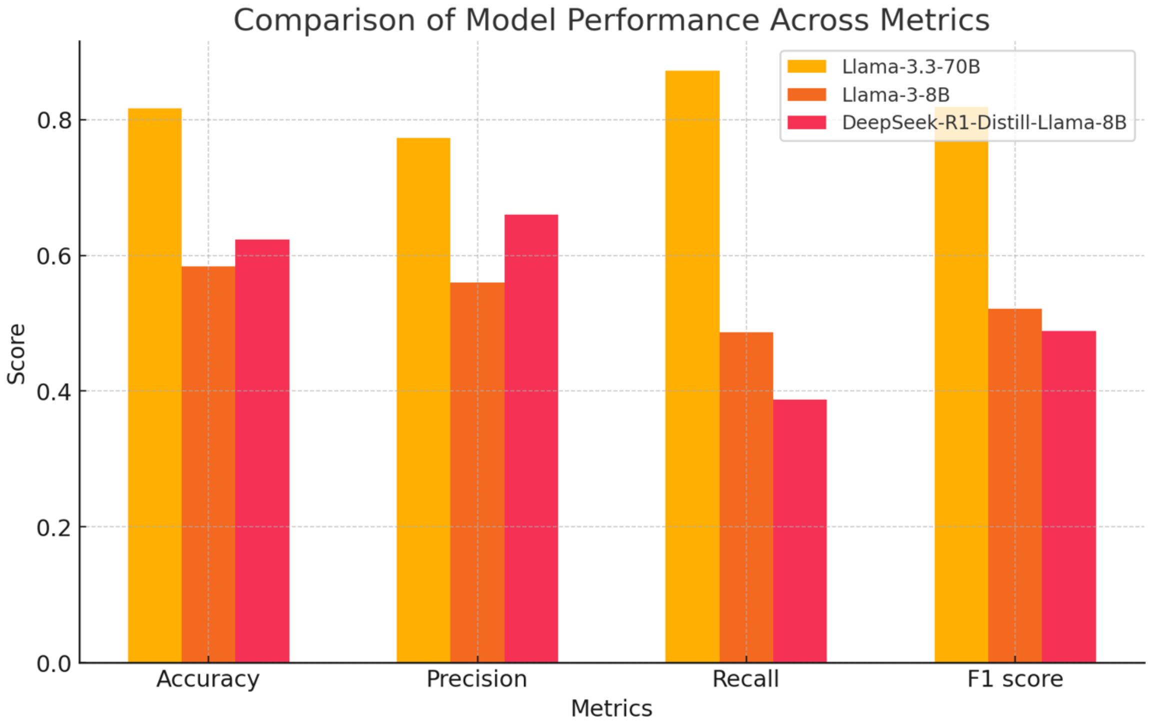
Hallucination detection

- **RAGTruth dataset :** to evaluate the hallucination detection capabilities of our LLM-based judge
- RAGTruth-based pipeline for evaluating LLMs as hallucination judges :



- **Results** of the comparison of hallucination detection models across tasks :





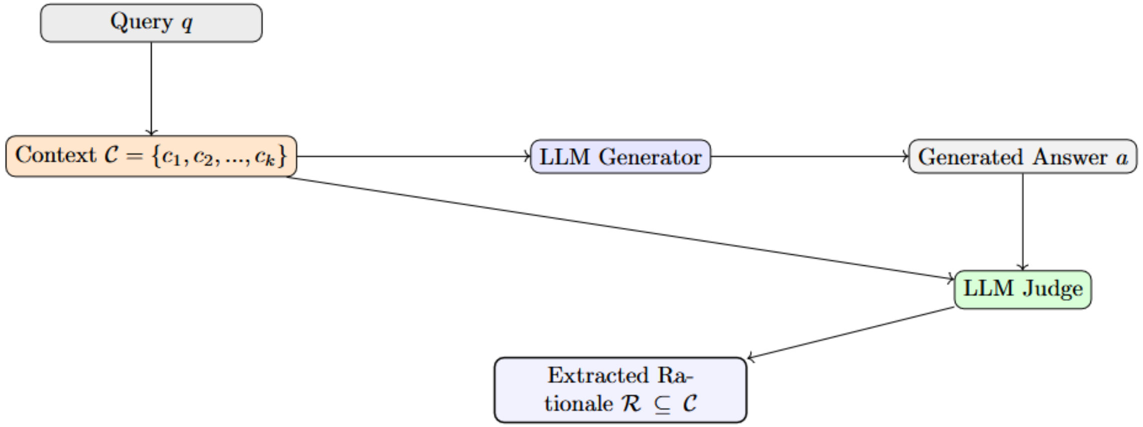
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \qquad \text{Recall} = \frac{TP}{TP + FN}$$

Hallucination judge selected : Llama-3.3-70B

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \qquad \text{Precision} = \frac{TP}{TP + FP}$$

Rationale extraction

- **Objective**: Enhance transparency by extracting **rationales** = specific context chunks that directly support generated answers.


Query q
Context $\mathcal{C} = \{c_1, c_2, ..., c_k\}$
LLM Generator
Generated Answer a
LLM Judge
Extracted Rationale $\mathcal{R} \subseteq \mathcal{C}$

- Given a query q, a generated answer a, and a set of retrieved context chunks C = {c1, c2, ..., ck}, the goal is to extract a subset R ⊆ C such that R justifies the answer a. (a | S) represents how well subset S of context chunks grounds the answer a.

$$\mathcal{R} = \arg\max_{\mathcal{S}\subseteq\mathcal{C}} \ \text{support}(a \mid \mathcal{S})$$

- In practice : Use of a powerful LLM (e.g., LLaMA-3.3-70B) as a judge to extract key supporting segments.

Metric rationales

- **Faithfulness :** measures how accurately an LLM-generated output aligns with the provided context. Using the LLM-as-a-Judge, faithfulness can be quantified by asking the model to:
  - Identify the number of true statements T in the output.
  - Compare these to the total number of statements S

$$\text{Faithfulness} = \frac{T}{S}.$$

- **Answer Relevancy (AR) :** evaluates how well the generated response answers the original query.

Using the LLM-as-a-Judge, the following process is employed:

- The query q is used to generate sub-queries qi from the provided output.
- The similarity between qi and q is computed using cosine similarity.

$$\text{AR} = \frac{1}{n} \sum_{i=1}^{n} \text{cosine\_sim}(q_i, q).$$

Metric rationales

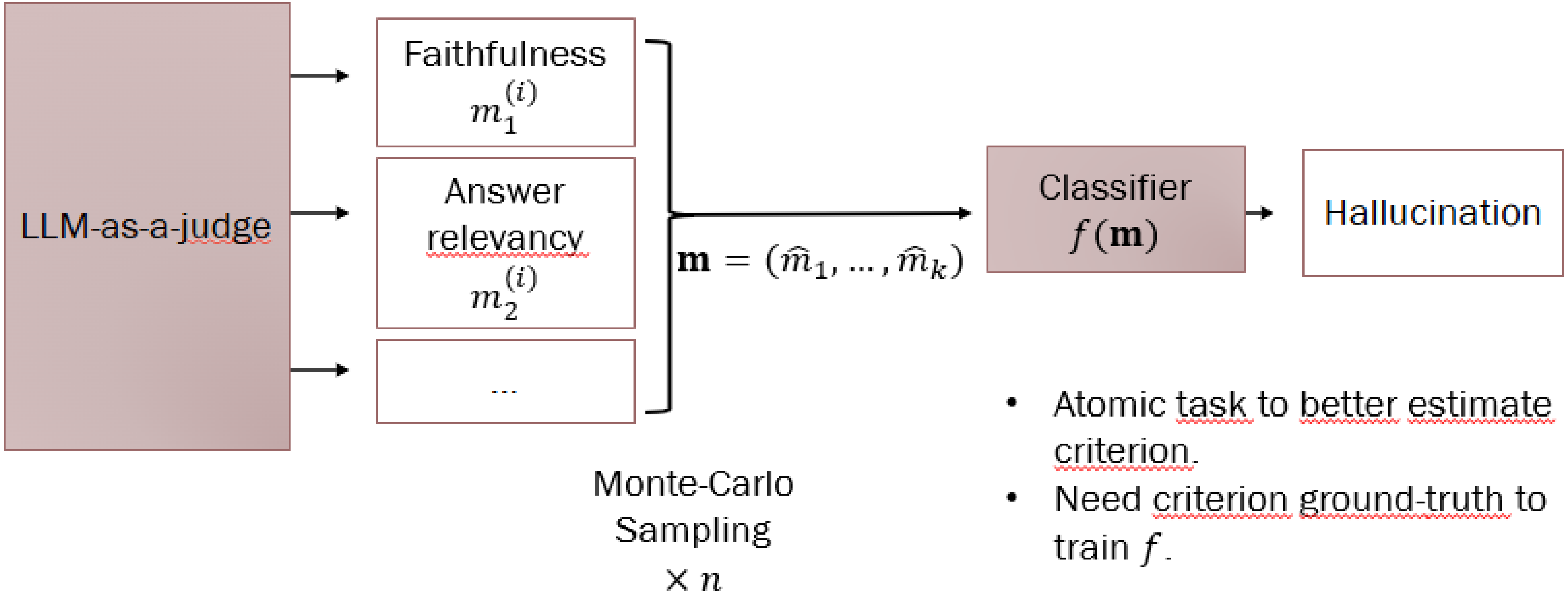
- **Completeness** assesses whether the output captures all relevant information from the context. One way to measure the completeness of a prompt is to compute the context relevancy (the LLM as a judge is used to identify and count relevant sentences based on their alignment with the Context)

$$\mathrm{CR} = \frac{\#\text{relevant sentences}}{\#\text{total sentences}}.$$

- ⇒ Score may not be as intuitive to interpret
- ⇒ Grade system may be better as we can input several criterion for each grade

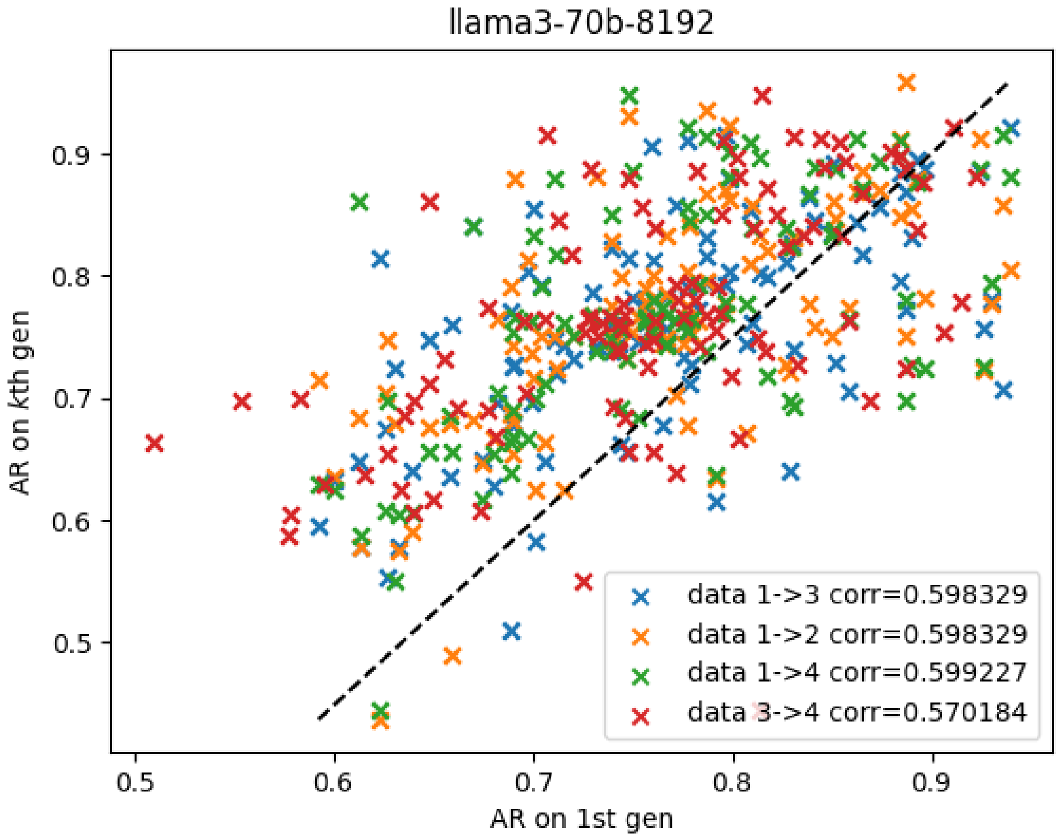
- **Problem**: how to recontextualize a metric score in overall setting?

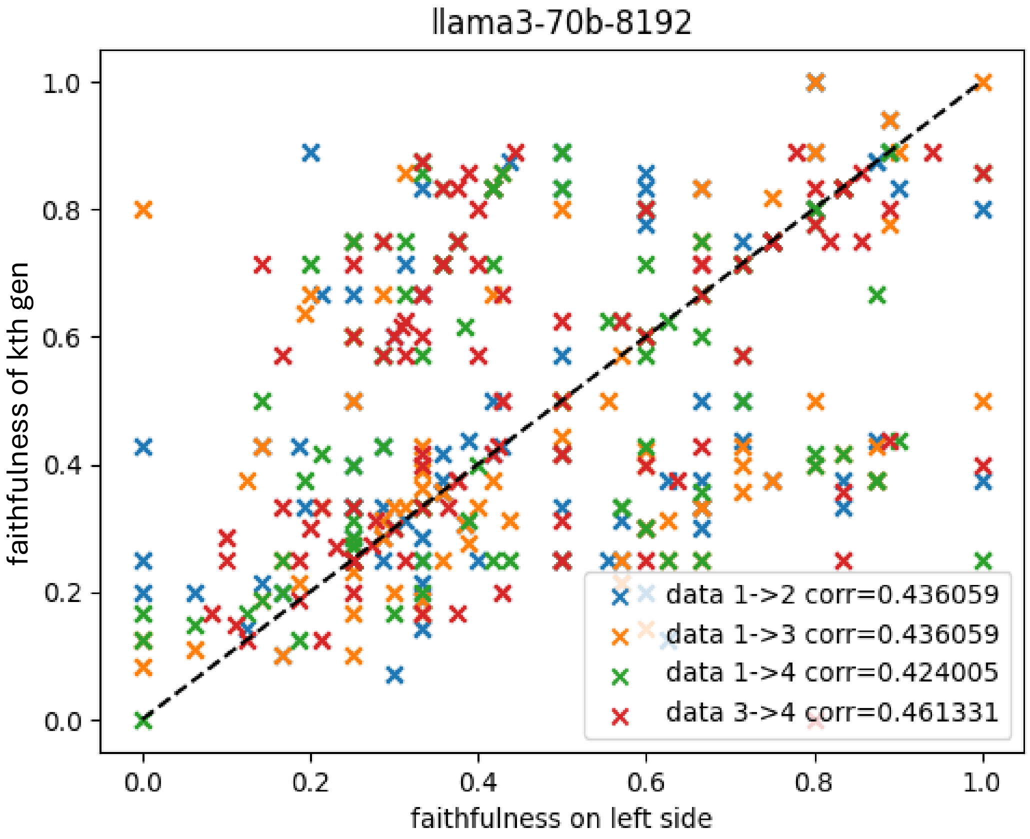
Factor-based classification



- Atomic task to better estimate criterion.
- Need criterion ground-truth to train f.

Illustration of the effect of the temperature T on the robustness of the metrics

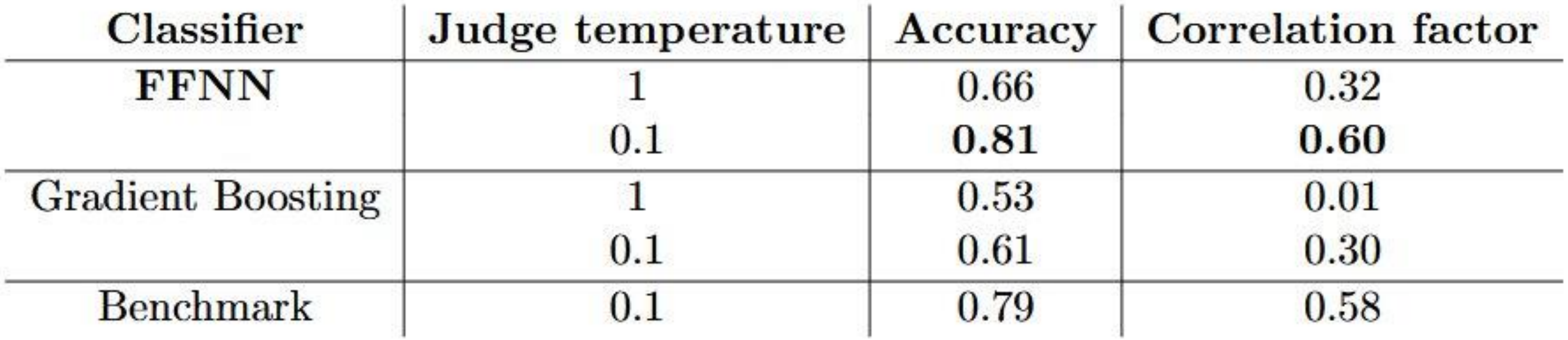




Hallucination detection using the RAGTruth dataset

- Metric set : (faithfulness, answer relevancy, context relevancy) used as the input of the classifier model.
- The confidence metric C is equal to 1 if the RAG output is hallucinated.
- Classifier model : simple feed-forward neural network (**FFNN**) with two hidden layers

Vs **benchmark model** LLM-as-judge outputting an hallucination score in {0,1}



Classifier	Judge temperature	Accuracy	Correlation factor
FFNN	1	0.66	0.32
	0.1	**0.81**	**0.60**
Gradient Boosting	1	0.53	0.01
	0.1	0.61	0.30
Benchmark	0.1	0.79	0.58

V. Explainability

- Rationale extraction evaluates RAG's retrieval and generation components to assess answer reliability but does not provide complete failure analysis.

- Metrics improve RAG behavior understanding but do not fully explain *why* errors occur.

- Broader explainability techniques are required for comprehensive RAG assessment.

Chain-of-thought (CoT) prompting

- CoT breaks down generation into discrete, interpretable steps, explicitly linking retrieved documents to logical/mathematical inference traces.

- CoT enables systematic evaluation of both final outputs *and* intermediate reasoning, reducing opaqueness and allowing modular validation/debugging.

- CoT is applied via a "meta-prompt" that explicitly instructs the RAG model to generate step-by-step reasoning alongside answers, given the retrieved context.

Judge explanation

- LLM-as-a-judge explains its evaluations by extracting key statements, including hallucinated parts.
- Use of RAGTruth dataset which contains annotated hallucinated sequences.
- Use of NLP metrics ROUGE and BLEU to compare model-generated and ground-truth hallucination statements.

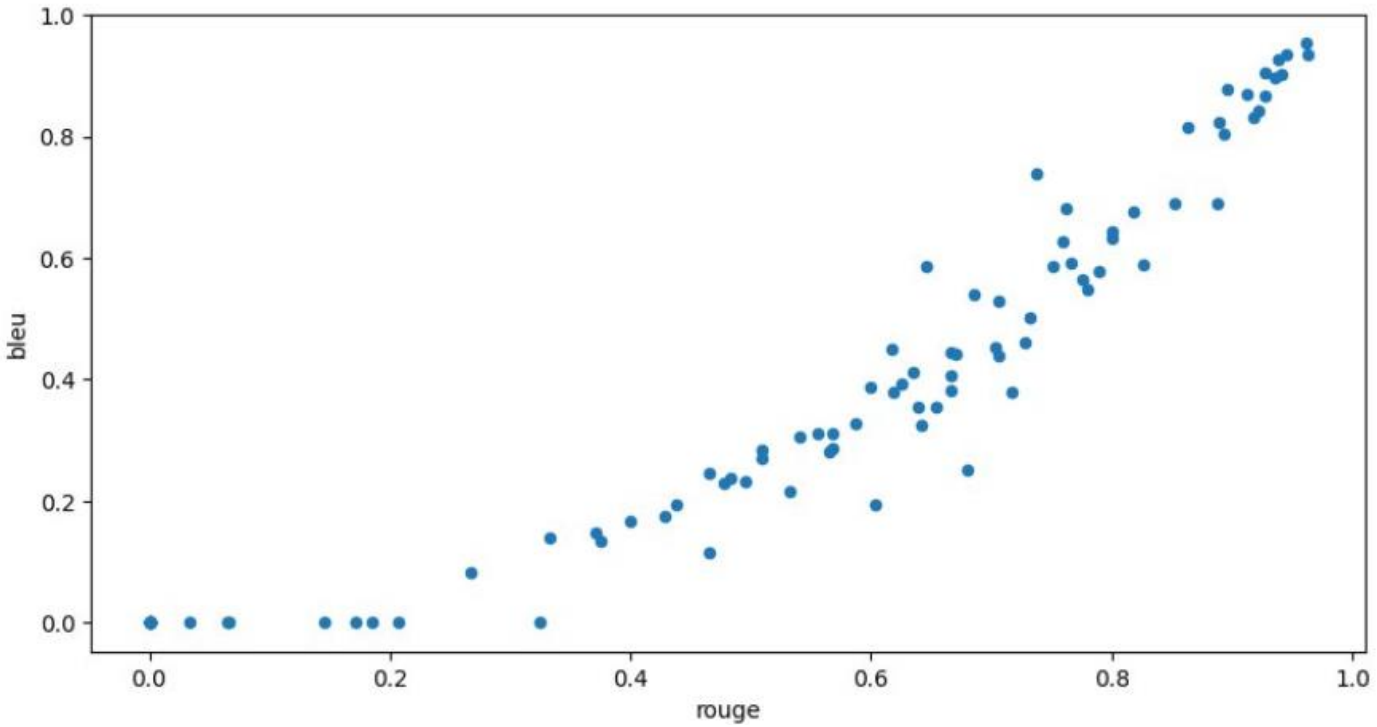
$$\text{BLEU} = e^{1-r/c} \exp\left(\sum_{n=1}^{N} \frac{\log p_n}{N}\right)$$

c : length of the generated text
r : length of the reference text
N maximum n-gram size (commonly N = 4)
n-gram precision : $p_n = \frac{\text{Number of matching } n\text{-grams}}{\text{Total number of } n\text{-grams in the generated text}}$

$$\text{ROUGE-}N = \frac{\sum_{S \in \text{References}} \sum_{n\text{-gram} \in S} \min(\text{Count}_n(\text{Generated}), \text{Count}_n(S))}{\sum_{S \in \text{References}} \sum_{n\text{-gram} \in S} \text{Count}_n(S)}$$

S : contiguous subset of the reference text
$Count_n$: number of occurrences of a specific n-gram in the input text

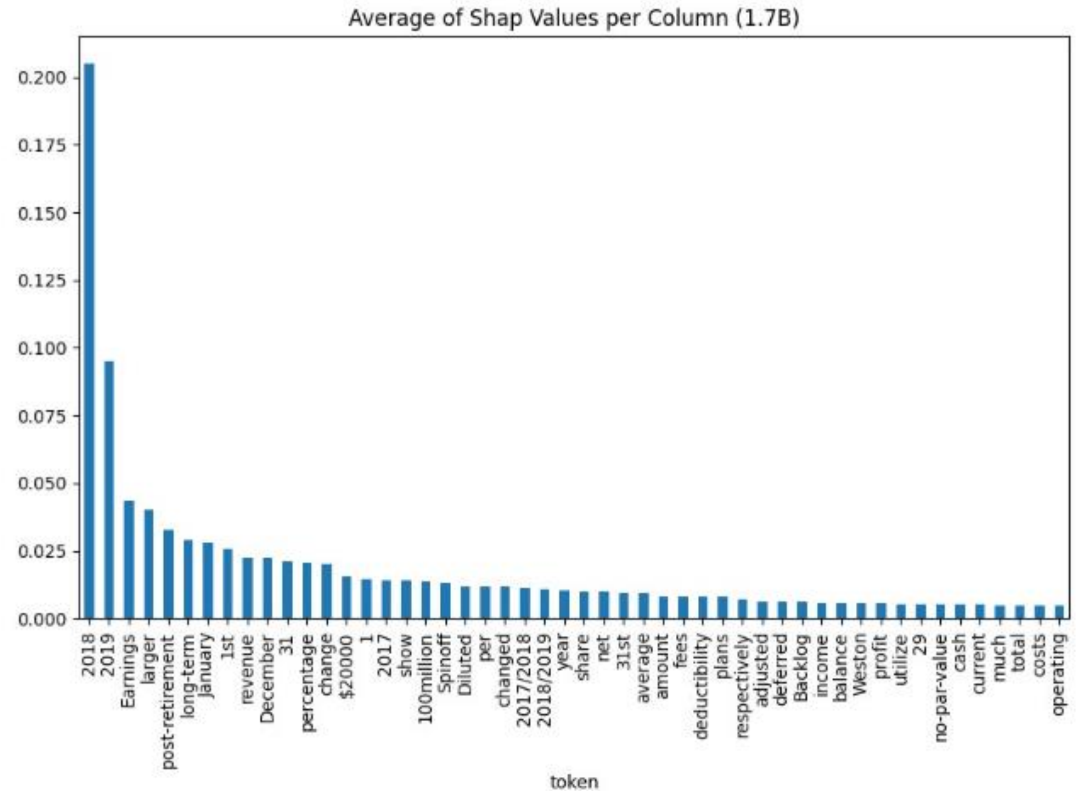
BLEU vs. ROUGE scores for the detected hallucinations in the RAGTruth dataset



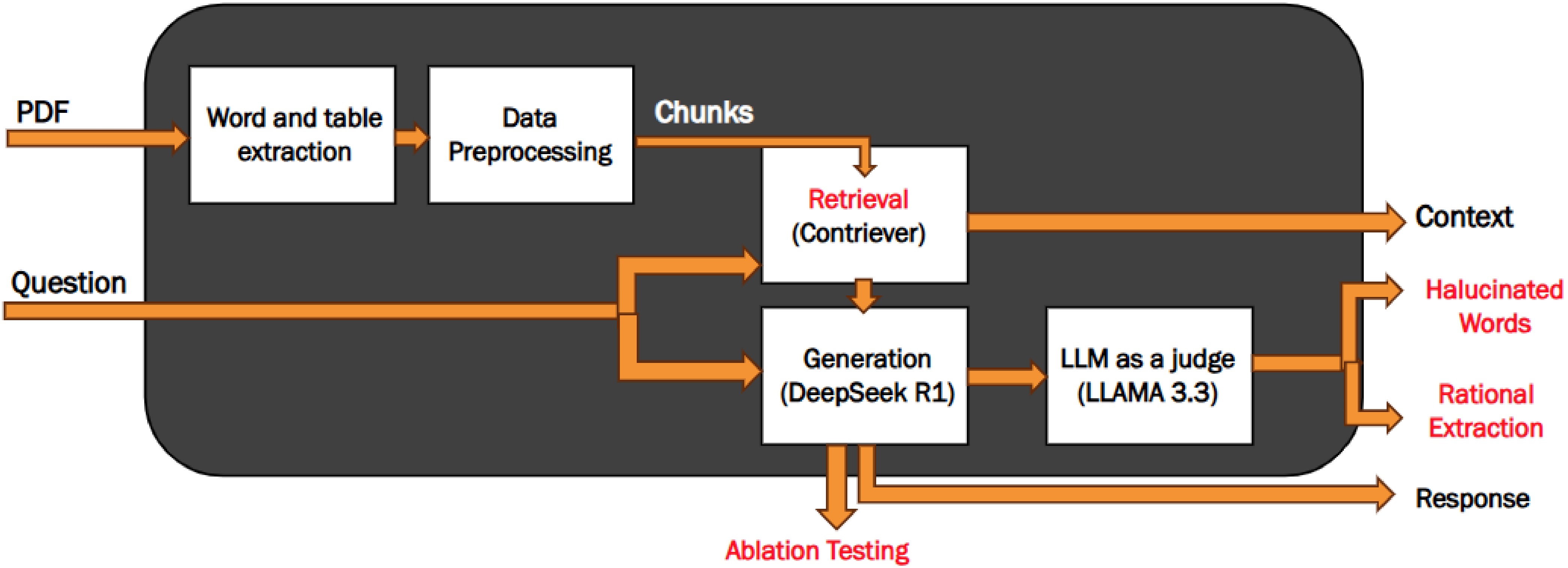
- BLEU and ROUGE measure how many words or n-grams two sequences share.
- 1 means an exact match, 0 means no overlap.
- Most detected hallucinations by the judge have non-zero scores, indicating general alignment with ground truth annotations.

Feature attribution

- To measure the importance of a specific word *i*, remove it from the prompt, compute the explainable LLM embedding v_{-i}, and c_{-i} = *cosine_sim*(b, v_{-i}) with the reference embedding *b*
- Importance of word i (N = lenght of prompt) : $imp_i = \frac{1}{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} c_{-j} - c_{-i}$
- Average importance values for detected hallucinations in the TAT-QA dataset :



VI. Pipeline & Conclusion

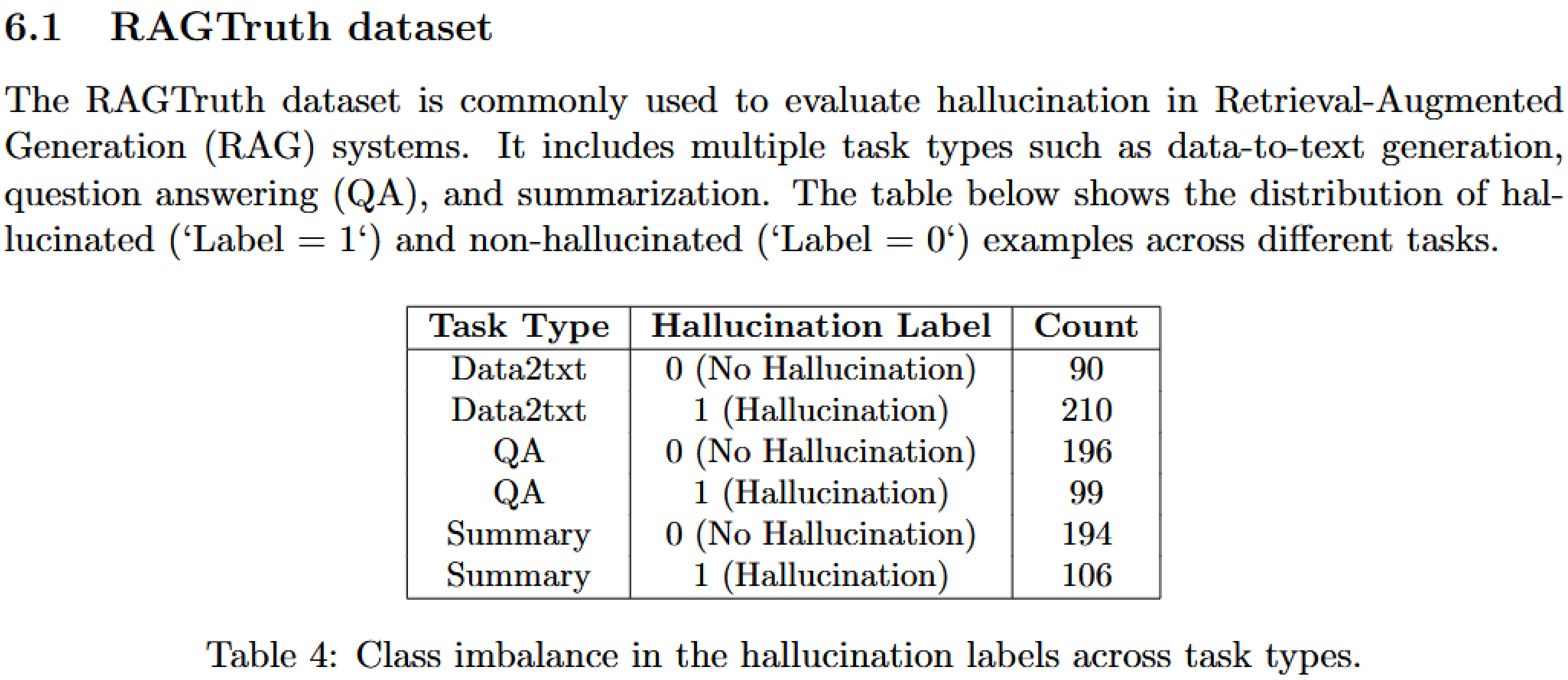


Components in red : where explainability is enforced

<table>
<tr><th>Question</th><th>Context</th><th>Similarity score</th><th>Answer</th><th>Ablation testing</th><th>Halucinated words</th><th>Rartional Extraction</th></tr>
<tr><td>What is the impact of rising interest rates on corporate investments and real estate?</td><td>• 'Asset: Bitcoin | 2023 Start Price: $16,500 | 2023 End Price: $28,400 | % Change: 72.1%'
• 'Asset: Ethereum | 2023 Start Price: $1,200 | 2023 End Price: $1,900 | % Change: 58.3%'
• 'Market Trends The S&P 500 and NASDAQ indices have shown mixed performance, with technology stocks leading gains while energy and traditional industries lagged.'
• 'Index: S&P 500 | 2023 Start: 3,800 | 2023 End: 4,250 | % Change: 11.8%'
• 'Table 1: S&P 500 and NASDAQ Performance Rising interest rates have affected borrowing costs, influencing real estate and corporate investments.'</td><td>• -0.119
• -0.29
• -0.306
• -0.46
• -0.58</td><td>Rising interest rates have increased borrowing costs, leading to reduced corporate investments and slower growth in the real estate sector. This has been evident as the S&P 500 and NASDAQ have shown mixed performance, with technology stocks outperforming while energy and traditional industries lag. The S&P 500 rose 11.8% and NASDAQ gained 72.1% in 2023.</td><td>• 0.7374
• 0.6583
• 0.8115
• 0.6279
• 0.578</td><td>['NASDAQ gained 72.1% in 2023.']</td><td>- Rising interest rates have affected borrowing costs, influencing real estate and corporate investments.
- Index: S&P 500 | 2023 Start: 3,800 | 2023 End: 4,250 | % Change</td></tr>
</table>

Question	Context	Similarity score	Answer	Ablation testing	Halucinated words	Rartional Extraction
'How did the S&P 500 and NASDAQ indices perform in 2023?	• 'Asset: Bitcoin | 2023 Start Price: $16,500 | 2023 End Price: $28,400 | % Change: 72.1%' • Asset: Ethereum | 2023 Start Price: $1,200 | 2023 End Price: $1,900 | % Change: 58.3%' • 'Market Trends The S&P 500 and NASDAQ indices have shown mixed performance, with technology stocks leading gains while energy and traditional industries lagged.' • 'Index: S&P 500 | 2023 Start: 3,800 | 2023 End: 4,250 | % Change: 11.8%' • 'Table 2: Cryptocurrency Performance in 2023 Cryptocurrencies and blockchain-based assets have gained traction	• -0.022 • -0.209 • -0.224 • -0.35 • -0.53	The S&P 500 rose by 11.8% in 2023, while the NASDAQ also saw gains.	• 0.8070 • 0.8592 • 0.7830 • 0.6542 • 0.8726	the NASDAQ also saw gains	- Index: S&P 500 | 2023 Start: 3,800 | 2023 End: 4,250 | % Change: 11.8%

VI. Appendix

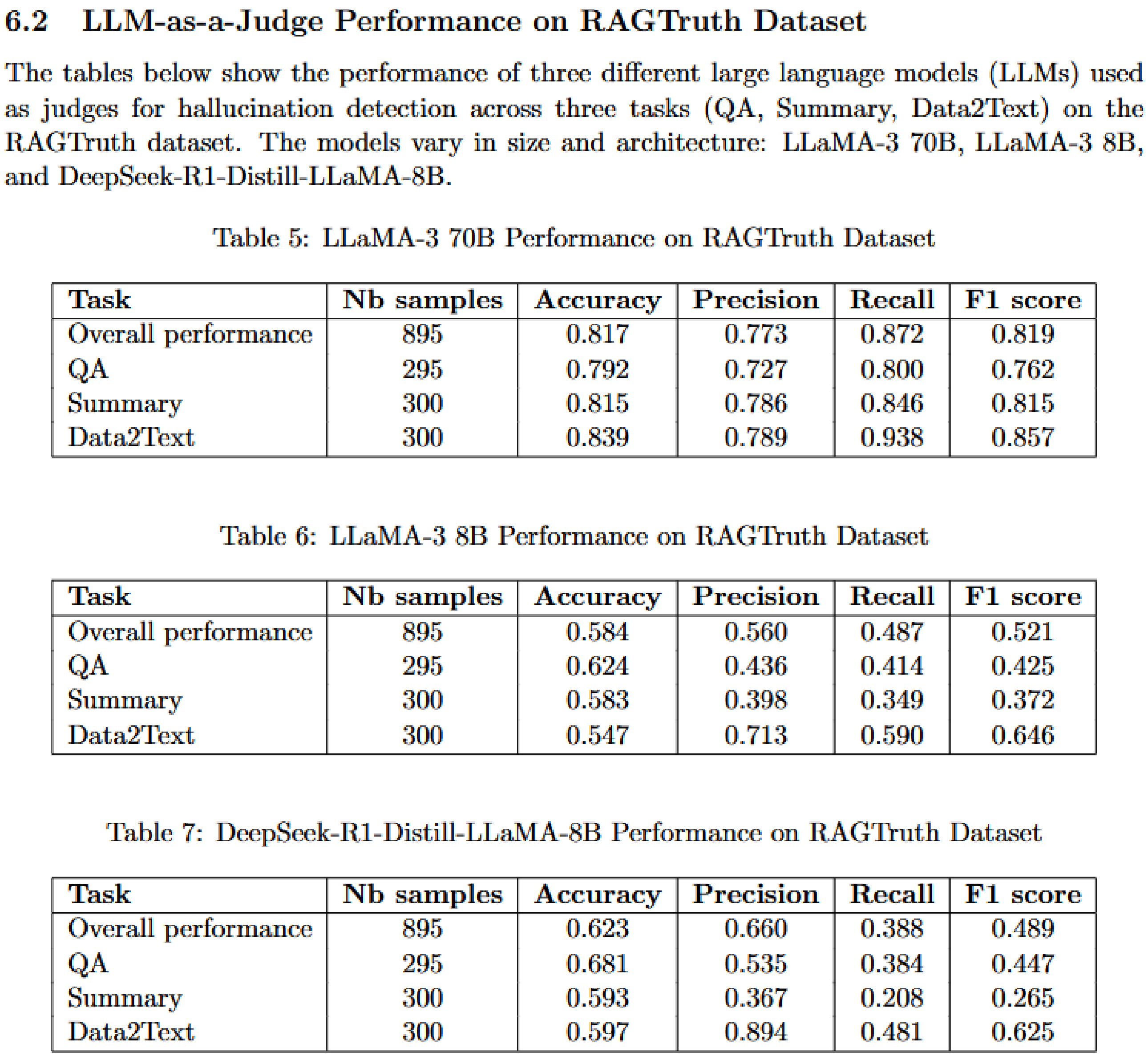


6.1 RAGTruth dataset

The RAGTruth dataset is commonly used to evaluate hallucination in Retrieval-Augmented Generation (RAG) systems. It includes multiple task types such as data-to-text generation, question answering (QA), and summarization. The table below shows the distribution of hallucinated (‘Label = 1‘) and non-hallucinated (‘Label = 0‘) examples across different tasks.

Task Type	Hallucination Label	Count
Data2txt	0 (No Hallucination)	90
Data2txt	1 (Hallucination)	210
QA	0 (No Hallucination)	196
QA	1 (Hallucination)	99
Summary	0 (No Hallucination)	194
Summary	1 (Hallucination)	106

Table 4: Class imbalance in the hallucination labels across task types.



6.2 LLM-as-a-Judge Performance on RAGTruth Dataset

The tables below show the performance of three different large language models (LLMs) used as judges for hallucination detection across three tasks (QA, Summary, Data2Text) on the RAGTruth dataset. The models vary in size and architecture: LLaMA-3 70B, LLaMA-3 8B, and DeepSeek-R1-Distill-LLaMA-8B.

Table 5: LLaMA-3 70B Performance on RAGTruth Dataset

Task	Nb samples	Accuracy	Precision	Recall	F1 score
Overall performance	895	0.817	0.773	0.872	0.819
QA	295	0.792	0.727	0.800	0.762
Summary	300	0.815	0.786	0.846	0.815
Data2Text	300	0.839	0.789	0.938	0.857

Table 6: LLaMA-3 8B Performance on RAGTruth Dataset

Task	Nb samples	Accuracy	Precision	Recall	F1 score
Overall performance	895	0.584	0.560	0.487	0.521
QA	295	0.624	0.436	0.414	0.425
Summary	300	0.583	0.398	0.349	0.372
Data2Text	300	0.547	0.713	0.590	0.646

Table 7: DeepSeek-R1-Distill-LLaMA-8B Performance on RAGTruth Dataset

Task	Nb samples	Accuracy	Precision	Recall	F1 score
Overall performance	895	0.623	0.660	0.388	0.489
QA	295	0.681	0.535	0.384	0.447
Summary	300	0.593	0.367	0.208	0.265
Data2Text	300	0.597	0.894	0.481	0.625