Evaluation of Education Chatbot RAG Application

Ankit Goyal

Methodology

Retrieval Metrics

Context Precision

Methodology:

- Utilised cosine similarity between TF-IDF vectors of the retrieved documents and the relevant documents.
- Calculated the precision as the ratio of true positive matches (documents with similarity above a threshold) to the sum of true positive and false positive matches.

Formula:

Precision = True Positive / (True Positive + False Positive)

Context Recall

Methodology:

- Similar to precision, but calculated recall as the ratio of true positive matches to the sum of true positive and false negative matches.

Formula:

Recall = True Positive / (True Positive + False Negative)

Context Relevance

Methodology:

- Calculated as the average cosine similarity score of the retrieved documents to the user query.

Formula:

Relevance = Σ Similarity Scores / Number of Retrieved Documents

Context Entity Recall

Methodology:

- Extracted entities from the retrieved documents and calculated recall as the ratio of relevant entities retrieved.

Formula:

Entity Recall = Relevant Entities Retrieved / Total Relevant Entities

Noise Robustness

Methodology:

- Assessed the system's ability to handle noisy inputs by adding noise to the query and evaluating the changes in precision and recall.

Formula:

Noise Robustness = 1 - Impact of Noise on Metrics

Generation Metrics

Faithfulness

Methodology:

- Measured the overlap between the generated answer and the ground truth.

Formula:

Faithfulness = Overlapping Terms / Total Terms in Ground Truth

Answer Relevance

Methodology:

- Similar to context relevance, but compared the generated answer to the user query.

Formula:

Answer Relevance = Σ Similarity Scores / Number of Answers

Information Integration

Methodology:

- Evaluated the ability of the generated answer to integrate information from multiple sources.

Formula:

Information Integration = Overlapping Terms with Context / Total Terms in Answer

Counterfactual Robustness

Methodology:

- Tested the system's robustness by providing counterfactual or contradictory queries and measuring the accuracy of responses.

Formula:

Counterfactual Robustness = Correct Responses / Total Counterfactual Queries

Negative Rejection

Methodology:

- Measured the system's ability to reject and handle negative or inappropriate queries.

Formula:

Negative Rejection = Correct Rejections / Total Negative Queries

Latency

Methodology:

- Measured the response time from receiving a query to delivering an answer.

Formula:

Latency = End Time - Start Time

Results

Metric	Before Improvement	After Improvement
Context Precision	0.84	0.92
Context Recall	0.84	0.96
Context Relevance	0.85	0.83
Context Entity Recall	0.69	0.33
Noise Robustness	0.64	0.96
Faithfulness	0.76	1.00
Answer Relevance	0.87	0.67
Information Integration	0.80	1.00
Counterfactual Robustness	0.52	1.00
Negative Rejection	1.00	0.00
Latency	0.15 ms	0.00 seconds

Methods Proposed and Implemented for Improvement

Context Precision and Recall:

- Implemented cosine similarity for more granular comparison

- Used a threshold to sum the similarity scores for precision and recall calculations.

Text Splitting and Embedding:
- Improved text splitting by adjusting chunk sizes and overlaps to capture more context.
- Enhanced embeddings by fine-tuning the model parameters.

Prompt Engineering:
Enhanced the prompt engineering by creating a custom prompt that includes a system message and current chat history.

Comparative Analysis

Before Improvements:
- Precision and recall scores were around 0.67 or 0.9
- Other metrics had lower values due to less optimised text splitting and embeddings.

After Improvements:
- Precision and recall now have more granular values (0.92 and 0.96 respectively), showing better handling of partial matches.
- Enhanced relevance and noise robustness metrics indicate better handling of query variations.
- Improved faithfulness, answer relevance, and information integration show the system's better performance in generating accurate and cohesive answers.

Challenges Faced and How They Were Addressed

Binary Precision and Recall:
Challenge: Initial precision and recall calculations resulted in binary values (0 or 1).
Solution: Implemented cosine similarity to allow for partial matches and more granular scoring.

Text Splitting Optimization:
Challenge: Finding the optimal chunk size and overlap for text splitting to retain context.
Solution: Experimented with different chunk sizes and overlaps to improve context capture.

Handling Noisy Inputs:
Challenge: Ensuring robustness against noisy or irrelevant inputs.
Solution: Added noise to queries and measured the impact on retrieval metrics, then adjusted the system to minimise this impact.

Performance Metrics Calculation:
Challenge: Calculating accurate performance metrics for evaluation.
Solution: Used libraries like sklearn for cosine similarity and implemented detailed metric calculations.