Member-only story

Evaluation of RAG (Retrieval-Augmented Generation) performance (Part 5 of RAG Series)

Quantifying the accuracy and relevance of the RAG output

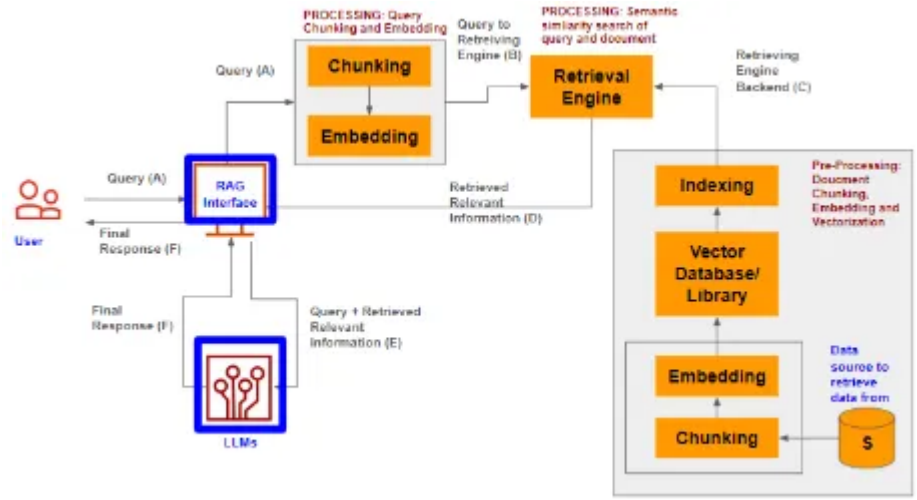
Chandan Durgia · Follow

9 min read · Feb 21, 2024

Photo by Luke Chesser on Unsplash

This is part 5 of the “Retrieval-Augmented Generation (RAG) — Basics to Advanced Series”. Links to other blogs in the series are at the bottom of this blog. Taking forward from part 1 (RAG Basics), part 2 (Chunking), part 3 (Embedding) and part 4 (“Vector Databases and Vector Libraries”). In this blog, we will focus on “Evaluation of RAG”.

Before we proceed into the details of this blog, I would like to conclude the story around the remaining components of the RAG architecture/ framework which we have been referencing in the past blogs. Taking from the last blog, once data from the Vector database is retrieved through the retrieval engine, the retrieved information is sent to the LLM through the RAG interface. LLM uses the retrieved information to “Generate” (note that this is the “Generation” part of the “Retrieval Augmented Generation”) the final output which is sent to the user through the RAG interface. (highlighted in Blue)



RAG Architecture (image by Author)

So, we have now covered an end-to-end mechanism of how RAG works and how each of the components play a role in the final output.

Though it sounds simple and intuitive till now, unfortunately it's not so !!

Developing a basic RAG with all the key components usually does not take more than an hour. The real challenge, like any other machine learning models, comes around improving the accuracy of the model output. i.e. whether the output from the RAG system:

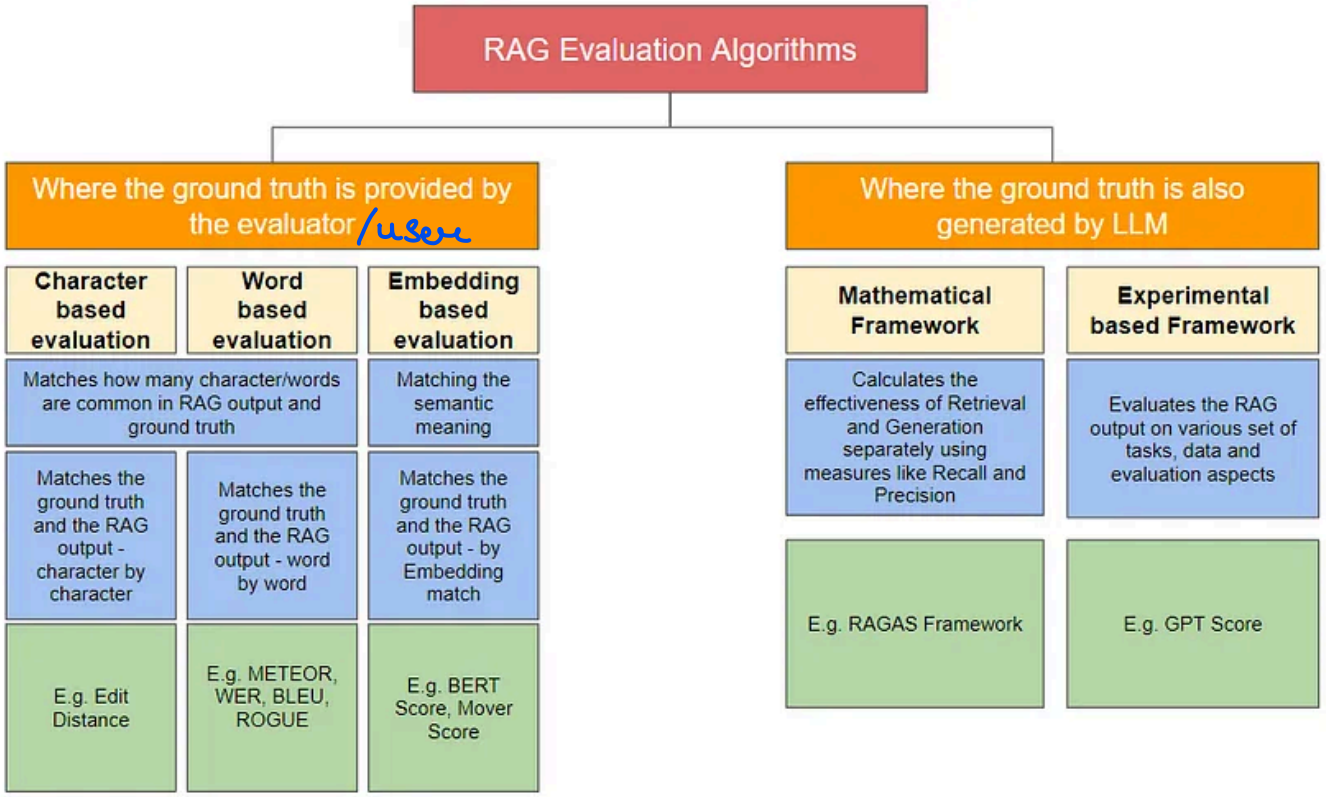
- Is of high-quality content, coherent and factually correct.
- Is relevant and complete
- Doesn't have lot of noise
- Is not harmful, malicious and toxic
- Is fast in terms of performance

But before we talk about improving the accuracy of any RAG system (next blog), it is important to establish the quantification/evaluation mechanism of the RAG systems. Unlike ML techniques where there are straightforward quantitative defined evaluation metrics (Gini, Precision, Recall, F1 score, R-sq, AIC, BIC, confusion matrix etc.), since the response in the RAG is unstructured text, the evaluation is done through a mix of qualitative and quantitative metrics.

Key caveat: Note that evaluation of LLMs is different from evaluation of RAG. RAG evaluation includes the evaluation of retrieval and the generation component with the specific input text. LLM evaluation covers a wide range of use cases and metrics. Some common frameworks for evaluating LLMs are HELM, OpenAI/eval, Alpaca Evaluation etc. To reiterate, this blog only focuses on the RAG evaluation.

The remaining part of the blog focuses on the common evaluation metrics and the conceptual/mathematical framework behind these metrics.

At a high level RAG evaluation algorithms can be bifurcated into two categories. 1) Where the ground truth (the ideal answer) is provided by the evaluator/user 2) Where the ground truth (the ideal answer) is also generated by another LLM. For ease of understanding, I have further classified these categories into 5 sub-categories — Character based evaluation, Word based evaluation, Embedding based evaluation, Mathematical Framework and Experimental based framework as depicted below.



RAG Evaluation Algorithms (image by author)

Let's deep dive into each of these evaluation categories:

1. Where the ground truth is provided by the evaluator

a. Character based evaluation algorithm

As the name indicates, this algorithm finds a score which is the character by character difference between the reference (ground truth) and the RAG

translation output. This algorithm is useful for instances where the RAG translation output is expected to be exactly the same as reference.

One of the key character based evaluation algorithms is:

i) Edit distance: It is a metric for which compares the RAG output and reference by counting the number of edits to be made to match both. The edits are measured as addition, deletions or amendments in the characters to match both machine translation and reference output.

Common use case: Language translation

b. Word based evaluation algorithm

As the name indicates, this algorithm finds a score which is the word by word difference between the reference (ground truth) and the RAG output. Again, this algorithm is useful for instances where the RAG translation output is expected to be exactly the same as reference.

Some of the most common word based evaluation algorithms are:

i) METEOR: This is a one of the simplest score based on explicit word-to-word matches between the RAG output and a given reference translation.

Common use case: Language translation

ii) Word Error Rate(WER): Similar to METEOR but rather than absolute matching it measures the percentage of words not matching. For example, 10% WER means 90% of the words are matching in the translation.

Common use case: Language translation

iii) BLEU Score: (Bilingual Evaluation Understudy Score): Again, this is very similar to METEOR with key difference being that rather than word by word comparison metric, BLEU compare the RAG output and reference sentence by matching n-grams (unigram, bigram etc.). Also, the comparison is agnostic of word order.

Note that BLEU works quite well for short sentences but have been shown to have relatively low correlation with human judgments, especially for tasks that require creativity and diversity

Use cases: Translation, Text summarization

iv. ROGUE score (Recall-Oriented Understudy for Gisting Evaluation Score) ROGUE score uses Precision and Recall over the “overlapping” words

between the translation and given reference.

Recall is the ratio number of overlapping words of RAG output and total words in reference summary (ground truth) — which basically tells how accurately the translation has happened. However, this is half of the story, it could be a possibility that the generated summary is very long and therefore captures all words in the reference summary — but there are many words which are just noise.

The other half of the story is captured by precision which measures the coherency and relevance of the output — it is the ratio of number of overlapping words of RAG output with ground truth and total words in the RAG output. Similar to the other ML techniques, one can create an F1 score on the basis of the precision and recall value.

There are various variants of ROUGE wherein instead of overlapping words, it could be overlapping n-grams (ROUGE-N), overlapping longest matching sequence (ROUGE-L) and overlapping skip grams (ROUGE-S)

Again similar to BLEU, ROUGE have been shown to have relatively low correlation with human judgments, especially for tasks that require creativity and diversity

Use cases: Summarization and Translation, ROGUE works well for short sentences.

c. Embedding based evaluation algorithms

Embedding based algorithms works in two steps:

Step 1: Create embeddings for both the generated text and the reference text using a particular embedding technique

Step 2: Use a distance measure (like cosine similarity) to evaluate the distance between the embeddings of the generated text and the reference text. Closer the distance, better the match which implies better RAG outcome.

Two most common Embedding based algorithms are:

i) BERT Score: Aligned with the above two steps, BERT score works as follows:

1. BERT score uses BERT embeddings (which capture “contextual” meaning of each word/phrase) and creates Embedding for both the generated and the reference text.

2. For the distance measure, these Embedings are compared for assessing semantic similarity using Cosine Similarity

Note that the Embeddings capture the contextual similarity and not word similarity. This is extremely useful as it understand the inherent meaning rather than focussing on the individual words. For example: consider two sentences 1) A person is sleeping with his pet cat. 2) He is resting with a cat which he takes care of. Not though the words in both the sentences are quite different, the contextual (semantic) meaning of both the sentences are similar.

BERT score also provides other metrics like Precision, Recall and F1 score.

This is done by comparing each token in the generate sentence is matched to the most similar token in the reference sentence, and vice versa, to compute Recall and Precision, which are then combined to calculate the F1 score.

Use cases: BERT score works well for small text output (e.g. chats)

ii) Mover Score: Again, aligned with the two steps, Mover score works as follows:

1. Again, both generated and reference texts are encoded by contextualized word embeddings finetuned on Multi-Natural-Language-Inference (MNLI)
2. For the distance measure, Earth Mover Distance is leveraged to compute the semantic distance by comparing two sets of embeddings resp. to the system and reference text

Use cases: Mover score works well for text generation tasks, e.g., machine translation, text summarization, image captioning, question answering and etc.

2. Where the ground truth is also generated by LLM (LLM assisted evaluation)

a. Mathematical Framework — RAGAS Score

RAGAS is one of the most common and comprehensive frameworks to assess the RAG accuracy and relevance. RAG bifurcates the evaluation from Retrieval and Generation perspective.

From the Retrieval perspective, it measures context Precision and Recall.

1. **Context Precision: Precision measures the relationship between the question and the context (information source).** There are various ways to think about this, but potentially the most preferable way is what is the degree of hallucination or given the question, how accurate the extracted output is. In other words, how relevant the retrieved **context (from source)** is to the **question**. I.e. h It basically showcases the signal to noise ratio of retrieved content and conveys quality of the retrieval pipeline.
2. **Context Recall: Recall measures the relationship between the context (from source) and the Ground truth i.e. can my retrieval engine extract all the** relevant information from the context (source) to answer the question correctly and completely. In other words, it measures the ability to retrieve all necessary information.

E.g. Question: Who won the 2023 and 2019 cricket world cup?

Ground Truth: Australia won the 2023 cricket world cup and England won the 2019 cricket world cup.

Model Output: Australia won the 2023 cricket world cup.

In this case, the model output is low context recall as though the answer is correct but its incomplete and the model doesn't retrieve all relevant information.

From the Generation perspective, it measures context Faithfulness and Relevancy

1. **Faithfulness:** Faithfulness captures the relationship between the **Answer and the Context** and gauges whether the answer is factually correct. Again, this measures hallucinations from a generation perspective.
2. **Answer Relevancy:** This captures the relationship between **Question and Answer** and assesses whether the answer is relevant given the question, to what extent it is "to the point" and whether the answer provides partial information or redundant information.

b. Experimental Based Framework — GPT score

The effectiveness of this approach in achieving desired text evaluations through natural language instructions is demonstrated by evaluating experimental results on four text generation tasks, 22 evaluation aspects, and 37 corresponding datasets. This approach helps overcome longstanding challenges in text evaluation, such as achieving customized and multi-faceted evaluations without the requirement of annotated samples.

Note that, in this blog we have just focussed on RAG output accuracy and relevance measures. In the real world applications, there are other measures which are considered which measure the RAG output characteristics for harmfulness, coherence, conciseness, fluency, maliciousness, fairness etc.

Conclusion: In conclusion, this blog has provided insights into RAG evaluation, focusing on accuracy and relevance. We have explored various algorithms and techniques to assess the performance of RAG models in these areas. However, it is important to acknowledge that RAG evaluation encompasses additional aspects that have not been covered, such as coherence, conciseness, maliciousness, fairness, and more.

As the field of RAG evaluation continues to evolve, we can certainly anticipate the development of new evaluation mechanisms that encompass these diverse factors. It is crucial for all RAG practitioners to remain attentive and adaptable to keep pace with advancements in evaluation techniques.

Here is a view of what's in the series, feel free to let me know if you would like me to cover any specific aspect.