Evaluation

Evaluation Overview

For evaluating our system we will use 20 different queries of different range of difficulties for the system to generate summaries for. We will then compare the generated summaries with the human hardcoded summaries to see how well the system performs. Also there are 5 wildcard queries that are not included in the 20 queries, they are purposely tricky or wrongly formulated queries to see how the system behaves in such cases, if there is an incoherent answer or no answer at all.

ROUGE metric for evaluation

The ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metric is widely used to evaluate the quality of text summarization and other natural language processing tasks. It measures the overlap between a generated text (e.g., a summary) and a reference text. ROUGE primarily focuses on recall but can also be configured to assess precision and F1-score.

Here are its main variants:

1. ROUGE-N: Measures the overlap of n-grams (e.g., sequences of n words) between the generated text and the reference text.
   - ROUGE-1: Unigram (single-word) overlap.
   - ROUGE-2: Bigram (two-word sequence) overlap.
2. ROUGE-L: Evaluates the longest common subsequence (LCS) between the generated and reference text, considering the sequence structure.
3. ROUGE-W: A weighted variant of ROUGE-L that assigns more importance to consecutive matches.
4. ROUGE-S: Measures the overlap of skip-bigrams, allowing words to match even if they are not adjacent.

ROUGE is particularly effective for tasks where the generated output needs to resemble human-written text closely. By including it in your evaluation section, you emphasize a quantitative and standardized approach to assessing the quality of generated text.

In this particular we will be looking at the ROUGE-1 and ROUGE-L scores for the generated summaries, particularly the recall score.

Evaluation Metrics

We will rank 3 systems:

1. Our system with a basic subgraph generation - let's call it GRAPH-BASIC
2. Our system with an advanced subgraph generation - let's call it GRAPH-ADVANCED
3. A simple RAG system with vector embedding of full text (the articles themselves) - let's call it RAG-TEXT

We will use the state-of-the-art GPT-o1 model as a baseline

As for the metrics we will use are the following:

- Accuracy: how well the system performs on the 20 queries. This will have 2 parts: the ROUGE-1 and ROUGE-L scores for the generated summaries; and a auto generated out of 5 score from the baseline model, ie. the GPT-o1 model will be asked to generate a score out of 5 from the generated summaries, the average score will be displayed as **LLM Score**.
- Robustness: how well the system performs on the wildcard queries. A well handled query adds one to the score, meaning a perfect score is 5.
- Performance: average time needed to generate the query

Results

Accuracy

Values are approximated to the third decimal place

	ROUGE-1		ROUGE-L		
Model	**Recall**	**F1**	**Recall**	**F1**	**LLM Score**
GRAPH-BASIC	0.655	0.212	0.473	0.153	4.48
GRAPH-ADVANCED	0.891	0.083	0.744	0.069	4.86
RAG-TEXT	0.501	0.189	0.362	0.131	4.22

Performance

Model	AVG Time Spent (in s)
GRAPH-BASIC	15.407271986007691
GRAPH-ADVANCED	25.086044921875
RAG-TEXT	15.621680946350098

Robustness

Model	Correct wildcard answers (out of 5)
GRAPH-BASIC	4
GRAPH-ADVANCED	5
RAG-TEXT	5

Summary

As we can see in the results in terms of accuracy the GRAPH-ADVANCED model outperforms the other models both in terms of ROUGE scores and the LLM score. This is understandable because of how it generates a very contextual subgraph and therefore the answer generated needs to fill less gaps in the knowledge. The second best in that regard is the GRAPH-BASIC model, which still outperforms the regular RAG model for this task. We can see that the F1 score is the highest in both the ROUGE-1 and the ROUGE-L metric, which means that the generated summaries are more precise and concise that its competitors.

This is also reflected in the robustness of the models, where the GRAPH-ADVANCED model and the RAG-TEXT model both score a perfect 5 out of 5, while the GRAPH-BASIC model scores a 4 out of 5. This is because the GRAPH-BASIC model doesn't provide as much information to the generation part as the other models, and therefore it is more likely to fail on wildcard queries where some variables are not clearly provided.

In terms of performance the GRAPH-BASIC model is the fastest due to being the smallest context token window (both in input and output) from the three models. And evidently the GRAPH-ADVANCED model is the slowest due to the complexity of the subgraph generation algorithm.