## Your names

Roee Ezer 312493281, Aviv Weidenfeld 315589937

## A description of your system and the choices you made. We should be able to understand how your system works in all the different parts, and especially the key choices and key points that are required to replicate it. If something worked particularly well, specify it. Structure this part with a dedicated section for each of the stages: pre-processing, indexing (sparse and dense), retrieval, RAG (unconstrained and budget-constrained).

## Our RAG system is composed of 4 components (each has its interface and implementations):

**pre\_proccessor** – process the html pages to WebTextUnit (interface).

Here we decided that each WebTextUnit will represent a paragraph or a “section” as defined in the script extract\_content.py and each section will contain the id of the doc and the id of the section (just the index of the section inside the doc) and the full index of the section will be:

self.doc\_id + "\_" + self.section\_id

This approach worked well for us and we didn’t tried to changes it except for a caching mechanism (pickle serialization) to avoid processing the html files in each run.

**index\_optimizers –** these optimizers manipulate the query or the documents or both to improve the indexing performance.

The changes made to the optimizers are relevant only to the indexing part. The “final\_answer\_retriever” for example will use the original query/documents.

We tried many kinds of optimizers as you can see in the folder components/IndexOptimizer. The one that worked the best was prefix\_suffix\_splitter\_optimizer.py which was our implementation of lemmatization with a few rules of thumb that achieved very fast and good results in the recall,mrr metrics.

**data\_indexer - @aviv fill this part**

**final\_answer\_retriever**

**This class has a one implementation that sends Gemini through the free tier API the query and the text to answer by in the following pattern:**

**“**

**תקרא את השאלה הבאה:**

**{Query}**

**תענה בקצרה על השאלה לפי הטקסט הבא:**

**{AnswerSource}**

**“**

## The evaluation results you obtained.

* 1. End-to-end evaluation, in which you go from query to final answer. This will have to be done manually. Choose 10-20 queries and manually evaluate their answers. Beyond accuracy, see if you can find common trends. Additionally, see if you can identify cases where the retrieval found the correct document but the overall system produced a wrong answer, or the other way around: cases where the retrieval failed but the system overall produced an adequate answer. Do things change from the unconstrained to the token-budget constrained cases?
  2. End-to-end evaluation, in which you go from query to final answer. This will have to be done manually. Choose 10-20 queries and manually evaluate their answers. Beyond accuracy, see if you can find common trends. Additionally, see if you can identify cases where the retrieval found the correct document but the overall system produced a wrong answer, or the other way around: cases where the retrieval failed but the system overall produced an adequate answer. Do things change from the unconstrained to the token-budget constrained cases?

## Can you think of ways to perform non-manual end-to-end evaluation, given the data you created? Note that we asked you to also include an annotated text-span containing the answer. Can you use this for automatic evaluation of the end-to-end RAG system? How? Discuss this in your report.

## If you experimented with different methods and ideas, it is good to describe not only what worked, but also what didn’t work.

## What would you have implemented if you had unlimited time and compute resources? What would be an ideal approach that you think would work very well? Be specific, e.g. don’t say general things like “I will train my own similarity model” but describe on what kind of data you will train it.

## Any additional thoughts you had based on the assignment, or ideas you may have.

## A discussion of the challenges of part 2 compared to part 1, what are their causes, and what do you think is needed to overcome them.