ASSIGNMENT 7

CORPUS Q&A TOOL

TABLE OF CONTENTS

Query Words In The Content Are Top Priorities

- Firstly, we store the corpus of ninety-eight books in a Trie for fast insertion, deletion and search implementation.
- Time complexity of insertion, deletion and searching for a string of length ‘k’ in a Trie data structure is:

$$O(k)$$

- We are already given the frequency of words in the general corpus which are to be used for scoring a word in the given corpus according to the input query.

Rating a word & a Paragraph

- We utilized and modified the dictionary from the previous assignment to search for words in the input query

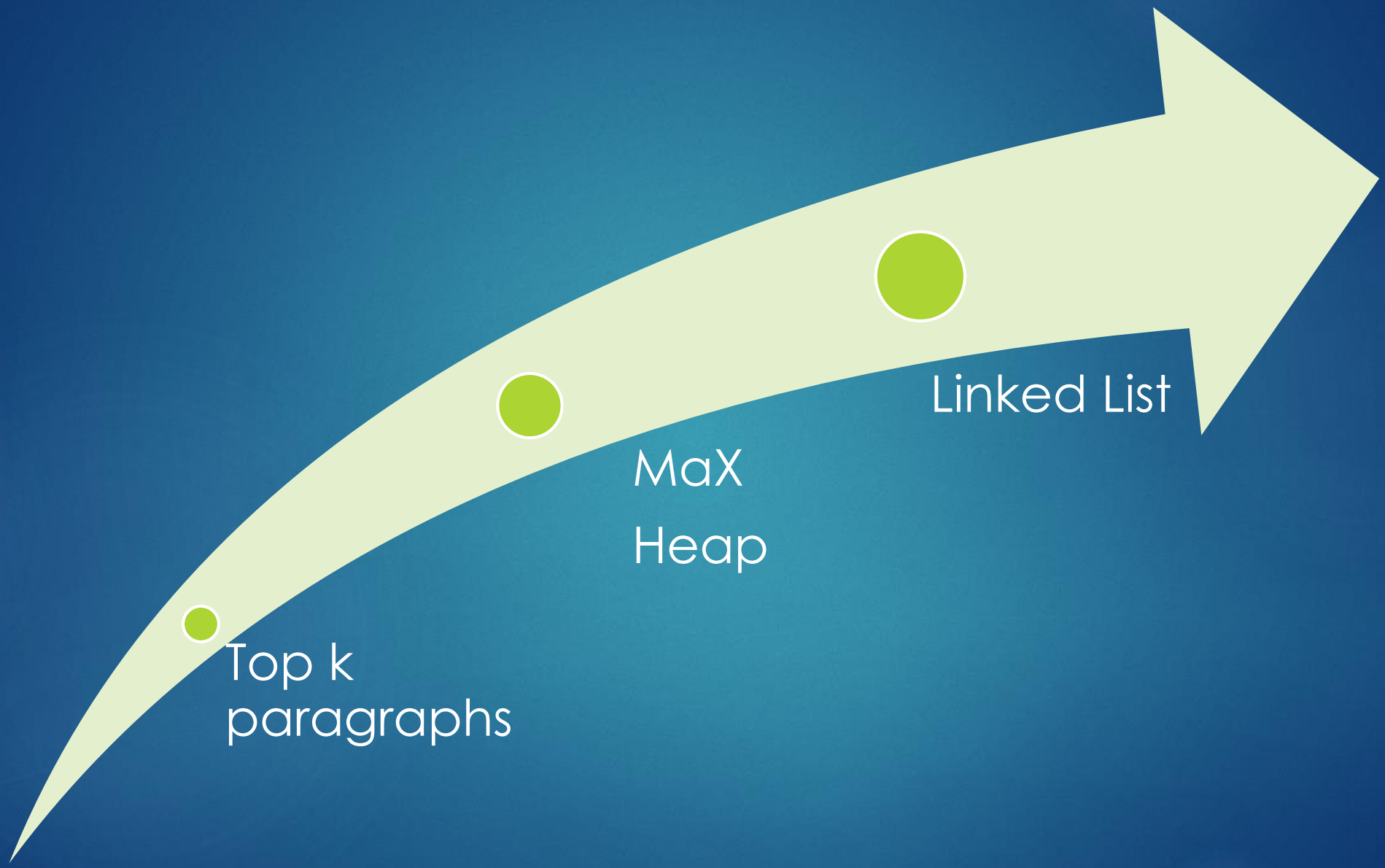
- As per the assignment, the scoring method has already been explained :

$$s(w) = \frac{\text{frequency in specific corpus} + 1}{\text{frequency in general corpus} + 1}$$

- Each paragraph was rated using a score assigned to every word in it.

Storage And Retrieval of Top-K Paragraphs

- After calculating the score for each paragraph using the formula provided in the assignment, we recorded the scores and stored them in the **scoretries.**
- All paragraphs with a score of 0 or higher are valid and retrieved from the MaxHeap data structure.
- These paragraphs are then passed on as linked list for processing through the LLM(ChatGPT).


Linked List
MaX
Heap
Top k
paragraphs

RESEARCH

- The maximum token limit of ChatGPT is 4096 tokens, i.e. it can process a maximum of 4096 characters in the worst case.
- Since a paragraph on average has 400 words we considered as an optimization to return the top 5 paragraphs.
- The data retrieval from a heap is maintained at a better rate than a Trie hence we retrieve it from a Max heap .

  Trie complexity – kn

  Max heap complexity – klog(n)

  here, k=number of para retrieved , n = para with valid scores

Prompt Engineering

- Since an AI tool has its own limitations regarding the amount of data that can be fed into it. Hence to increase its accuracy and processing time we used prompt engineering.
- So we developed prompt engineered code to sort out the relevant paragraphs to be fed to the LLM.
- The prompt engineering used by us were:

- **To decrease the score of common words from the input query like “a, an ,the, is ,am ”.**

Ex- If the input query is "When is Mahatma Gandhi's birthday celebrated?", the word 'is' does not increase the score in the trie.

- **We fed the input query to the LLM to find out the relevant words from the query and then further increased their scores.**

Ex- If we input "When is Mahatma Gandhi's birthday celebrated?" into the LLM, the relevant words "Mahatma" and "birthday" are scored higher.

Thank You!