QnA Tool Report

COL106 Assignment 7

Madhav Maheshwari 2022MT61975
Madhav Gupta 2022EE11737
Arunim Garg 2022EE32002
Tatsam Ranjan Sharma 2022MT61969

Data Structures Used

01 Tries

We are using tries for the corpora as Tries give us look up, get count feature in **O(|text|)**. This makes the algorithm pretty fast for large inputs.

02 Max Heap

Secondly we are implementing max heap for top k elements because it helps in extracting the top k in **O(n+klogn)**. This is preferred over sorting as sorting is **O(nlogn)**.

Algorithms Used

01

Tries

We maintained tries to store our main database of general counts.
Also to maintain the data about corpus and paragraph we maintained tries and a list of paragraphs.

02

Top K Function

In get top k function we calculated the scores and using max heap got the topk paragraphs based on their scores

03

Query Function

The algorithm for the query function is removing the irrelevant words(eg: what,and or) and then scoring the paragraphs in a different way and then sending the top 5 paragraphs to the LLM and answering the question.

Query Function

In query function to get further optimisations in getting top-k paragraphs

We implemented Porter stemming in the corpus and our question to get the top k paragraphs more efficiently

We also removed the Irrelevant words from the question to score the paragraphs.

Porter Stemmer

Stemming is the process of reducing inflected (or sometimes derived) words to their word stem, base or root form(form which is generally used).

Porter Stemmer is a Porter stemming algorithm.
It is an algorithm for suffix stripping.
Some of the rules of Porter Stemming algorithm are as follows:

- sses->es
- ies->I
- ational->ate
- tional->tion
- s->Ø

When there are conflicts the longest rule wins

Example:

- Economy,economic,economical
  -> economy
- Automates,automatic,automation
  ->automat

Pseudocode

```
QNA_tool(){
        make main database trie and intialize
        other instances
}

~QNA_tool(){
          destruct the tries formed
}

Insert_sectence(string sentence){
    if(condition for if there is new paragraph
        starting){
         insert new paragraph in paragraph list}

      break the sentence into words using
      sepearators;

       insert the words in para graph trie and
       corpus trie
}

Get_topk(question,k){
      initialize max heap;
      get the score for paragraphs;
      buildheap(array of paragraph data);
      delete max for k times
}
```

```
Query (question){
    remove the unwanted words and any
    other opimisations
    get top 5 paragraphs
    send the paragraphs and question to
    LLM
}
```

Time Complexity

```
QNA_tool(){
        make main database trie and intialize
        other instances
}

    ~QNA_tool(){
            destruct the tries formed
    }

Insert_sectence(string sentence){
   if(condition for if there is new paragraph
      starting){
       insert new paragraph in paragraph list}

    break the sentence into words using
    sepearators;

     insert the words in para graph trie and
     corpus trie
}

        Get_topk(question,k){
            initialize max heap;
            get the score for paragraphs;
            buildheap(array of paragraph data);
            delete max for k times
        }
```

O(|text in csv file|)

O(|sentence|)
≡
O(|text|)

O(|paragraphs| + klog|paragraphs|)

|paragraphs| is number of paragraphs

```
Query (question){
    remove the unwanted words and any
    other opimisations
    get top 5 paragraphs
    send the paragraphs and question to
    LLM
}
```

O(|paragraph| + 5log|paragraphs|)

We also did Prompt Engineering to get better results from LLM:
For eg: We have added instructions to the LLM to generate any possible interpretations from the provided text and give structured answers for subjective questions, rather than providing inconclusive results.

Also we used other LLM than chat gpt(Hugging Face)

Bonus Part

For this part we have used Hugging face for queries.

We compared its responses with those of chatgpt and after several followups ,we are getting similar and sometimes even better results than chatgpt.

```
Query: gandhi view on first prime minister of india
gandhi view on first prime minister of india
Book_code: 54 Page: 241 Paragraph: 4
Book_code: 77 Page: 132 Paragraph: 6
Book_code: 94 Page: 23 Paragraph: 3
Book_code: 72 Page: 99 Paragraph: 1
Book_code: 48 Page: 363 Paragraph: 1
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
ChatGPT:  Based on the provided excerpts, Mahatma Gandhi's view on the first Prime Minister of India is not explicitly mentioned. The excerpts touch upon var
ious topics such as India's independence struggle, governance, and financial matters, but do not directly address Mahatma Gandhi's view on the first Prime Mi
nister.

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
HugBot:   Based on the excerpts provided, it appears that Mahatma Gandhi had a critical view of the first Prime Minister of India, Jawaharlal Nehru. In the f
irst excerpt, Gandhi writes that Nehru's appointment as Prime Minister was "not made in accordance with the wishes of the people" and that it was "imposed up
on the Congress Party." This suggests that Gandhi believed that Nehru's position was not legitimate and that he did not have the support of the people or the
 party he represented.

In the second excerpt, Gandhi criticizes Nehru's salary, comparing it unfavorably to the salary of the British Prime Minister. He argues that the high salary
 paid to Nehru and other officials in the Indian government is a reflection of a corrupt system that prioritizes the interests of the ruling elite over those
 of the common people.
```