Task

Your objective is to develop a search algorithm capable of efficiently searching for phrases/sentences in a vast collection of documents and returning the corresponding fragments where those phrases or similar ones are found.

In **Section A** of the provided Colab notebook, there is a simple code that searches for phrases in documents and returns the fragments where the phrases are written, along with the names of the documents containing the corresponding fragments.

However, the current method has several drawbacks:

1. It iterates over all document fragments during each search, resulting in slow performance for large document collections.
2. If a middle word is deleted from the searched phrase, the algorithm fails to return any results.
3. Modifying word endings, using synonyms, or using lowercase text also does not yield any results.

Your task is to research and implement a search algorithm that overcomes these and other possible limitations and performs efficiently and rapidly on a vast collection of documents. Additionally, you need to explore other potential problems that might occur during the search process and propose solutions to avoid them.

Work Organization:

Code:

1. Code should be written in **Python**.
2. Write the code in the provided Colab notebook, in **Section B**. Ensure that you copy the Colab notebook to your Google Drive to enable auto saving of your code.
3. Create test examples to evaluate the performance and effectiveness of your algorithm.
4. Test your code on a dataset of 100,000 documents provided in Colab notebook.

Documentation:

After implementing the code, you must write documentation which includes the following sections (you can write the documentation in Armenian, Russian, or English).

Research Methods:

1. Describe the different methods you researched to improve the search algorithm.
2. Explain the criteria you used to select the specific methods used in your implemented code.

Testing Algorithm Performance:

1. Describe in details testing process used to assess the performance of your search algorithm.
2. Note that you only need to test the search part of your code. Preprocessing steps during document preparation do not require testing.

Results:

1. Present the results obtained from testing your algorithm.
2. Write about advantages and disadvantages of your algorithm.

Note: Use methods that you can explain, as during the interview, we may ask you questions about your code.

You must submit the Colab notebook (as .ipynb format) and documentation (in docx or pdf format) to this email address by 16.08.2023, 23:59 Yerevan Time.