To make the technical tasks clear for the final assignment, here are the steps:

1. Install Elasticsearch (more information),
2. Start it through the terminal (more information).
3. Install the Python library of Elasticsearch which connects you to Elasticsearch that you have started on your local machine in the previous step (python -m pip install elasticsearch==7.9.1)
4. Connect to Elasticsearch through python (more information)
5. Understand what is indexed in Elasticsearch (more information)
6. Mapping (more information) is the structure of your data (fairly like tables in databases). Prepare a Mapping JSON for your index. You only need one field for this task and can name it as “content” with the type “text”. Please be sure that you enable the term_vector for that field (more information). The Mapping JSON can be the one written below.
7. Create an Index based on your Mapping (create index)
8. Index documents of the Clinical Trial track (you can index queries in a separate index to be sure that the preprocessing for both queries and documents are equal) into the created index (an example of indexing with a sample document)
9. For each query, you need to call function “search” function of python library. The body for searching can be the one written below.
10. After retrieving 10k documents, you can re-compute the similarity scores with a chosen equation from the provided BM25 paper.
11. For evaluation, you can implement the evaluation functions yourself or you can use trec_eval or py_treceval.

P.S: Consider documents without qrels as irrelevant.

Mapping: You can use the below mapping for your index! It creates an index with one field named “content” and the default similarity function for the index is BM25. You can retrieve the top-10k documents from this index:

```
{
  "settings":{
        "index":{
              "number_of_shards":1,
              "number_of_replicas":1
        }
  },
  "mappings":{
        "properties":{
              "content":{
                    "type":"text",
                    "fielddata":True,
                    "term_vector":"with_positions_offsets_payloads",
                    "store":True,
```

```
                    "analyzer":"whitespace"
                }
        }
  }
}
```

Body: The below JSON can be used for “body” of your “search()” function:

```
bool_query = {
        "size": 10000,
        "query": {
                "bool": {
                "should": [
                {"match": {"Content": query_text}}
                ]
                ,"minimum_should_match": 1,
                "boost": 1.0
                }
        }
}
```
