1. **Which notions covered by lucene?(index construction, inverted indices, positional indices, index compression, vector space model, probabilistic scoring such as BM25, languagebased models, query expansion, spell correction, optimizations for top-k computation)**

Lucene supports -> Index construction, inverted index, top-k querying (MAKE SURE, SEARCH)

Lucene uses inverted indexes as an index structure to find the document which contains the term in the query. Lucene has three types of inverted index strategies which are batch-based, b-tree based and segment based strategies. Each segment index maintains following fields:

-Segment info

-Field names

-Stored Field values

-Term dictionary

-Term frequency data

-Normalization factors

-Term Vectors

-Per-document values

-Deleted documents

Lucene also uses index compression while indexing. Sorted words mostly have same prefixes. In index files, Lucene keeps one of the same prefixes for all and keeps differences sorted after them.

For scoring Lucene uses Vector Space Model and the Boolean model to find document relevancy to a query. VSM scores the document according to number of appearances of the query in the document. Boolean query being used for determining the documents which will be scored and after VSM is being used for final scoring. Lucene also has the support for probabilistic scoring, Okapi BM25. Okapi BM25 is a ranking function used for rank matching the documents based on their relevance to a given search query.

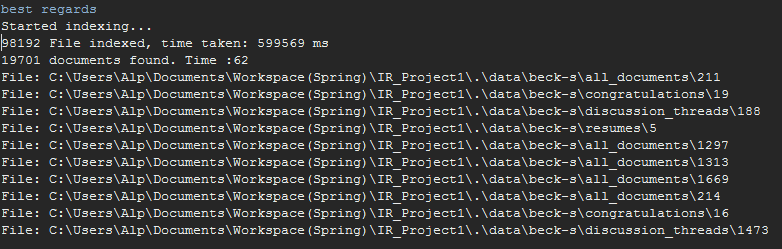
The application that we implanted, uses wildcard queries while searching but it does not make any spell correction.

2)

We decided to use “Enron E-mail Dataset” in this project. It contains approximately half million messages from 150 users. Dataset published by Federal Energy Regulatory Commission. Messages in the dataset does not include attachments and some of them removed due to privacy issues. Size of the dataset is X GBs so we decided to reduce it to two third of its size.

Performance Issues: Indexing time and search time. How many files are going to be shown to the user? How many of the shown results are relevant to the query?

In this example, since it was our first query system had to index the files first. After indexing has ended, system yielded the files that the search query appeared. In total 98192 files were indexed and it took approximately 10 minutes to index them. The search itself took 62 ms and in that time 19701 documents were found but only the top 10 of those files were presented to the user. The ordering of those files are based on the scores of the files. The console output can be seen in the picture.



References:

<https://www.tutorialspoint.com/lucene/lucene_first_application.htm>

<https://lucene.apache.org/core/3_6_2/api/all/index.html>

<http://www.lucenetutorial.com/lucene-in-5-minutes.html>