Retrospective Document
Sprint-4

Work & Test Progress

K-gram index selection mechanism implementation (completed 90%)
Extending logical plan generator (completed 95%)
Creating a base indexing structure with Lucene (completed 100%)

Apriori algorithm usage cases
Word count
K-gram generation in C and k-gram frequency
New test regexes to test recently added operators which are ‘[a-z]’, ‘[0-9]’, ‘\’, ‘\s’, ‘\a’, ‘\d’ , ‘{}’ and on.
Tests for Lucene indexing like query searching, recursive file indexing and Lucene’s n-gram tokenizer.

Team Progress

Oguzhan Demir 25%
Mustafa Guven 25%
Fatih Burak Belce 25%
Ozgur Baskin 25%

Left-overs (Backlog)

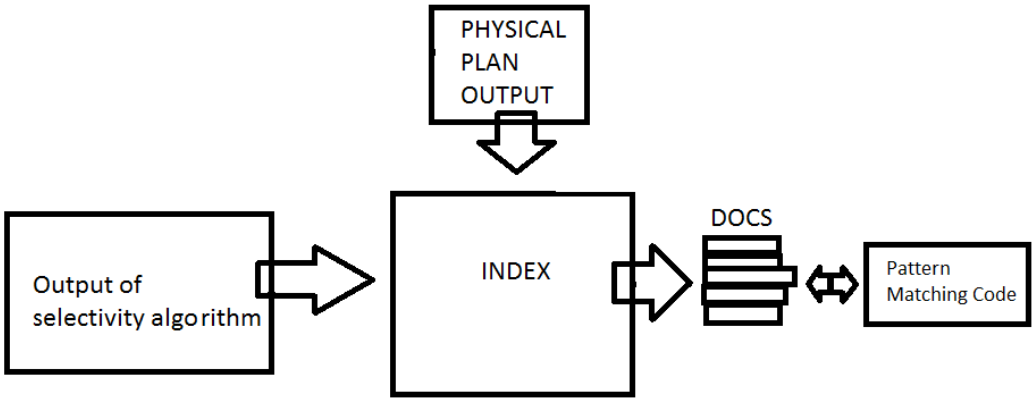
K-gram index selection mechanism implementation
This is mainly because index selector and index structure(Lucene) were both written in java, we needed to connect it to C++ code, then instead of connecting it, we decided to write one in c even though it was way harder to implement it, which had left the connection part not yet completed.

Extending logical plan generator
We extended the logical plan generator as we expected it to be. Though, the reason why we think we have not implemented it 100% was we haven’t implemented the error check for all of the operators. This would have taken a lot of time and we needed to focus on the other and more important milestones.

Creating a base indexing structure with Lucene
We now have a working “base index” in our hands. We can create an index out of the docs; we can add more docs and search queries in them. There is nothing to mention as a left over since we covered what we promised.

Next Sprint

A big milestone in this sprint would be integration of what we implemented thus far. This includes integrating apriori algorithm code, k-gram code, and frequency finder with the base index. We also need to make some decisions, whether to use Lucene as indexing or the other codes combined and converted into a new inverted index. Either way, we will need to apply selectivity algorithm before creating the index. Once index is created, the physical plan generator's result needs to know the created index. After this, search query will be made in the index to get the right docs. Last but not least; inside those docs regular expression pattern matching needs to be made. All these parts mentioned are working separately, but they need to be got together.



Comments

At this sprint, we had some hard decisions to make regarding the construction of the index and the use of algorithms. This led us spending most of our time thinking about the possibilities of the actions that we may take for the next sprint. All of the decisions had positive and negative sides and we couldn't predict the outcomes without completing it. Before the end of the sprint, as a result of the discussions at the weekly meetings, we chose a path and implemented the parts of the solutions one by one.

Assistant's Evaluation

Supervisors’s Evaluation