Team 4 Project README

1. Running our Disease Corpus Search Engine

Dependencies

i. JDK 1.8 or higher (already on c01)

ii. Maven 3.6 or higher (already on c01)

1. Maven will install any remaining dependencies (Lucene dependencies)

iii. You will need the following files for corpus data:

1. diseaseCorpus.txt: This is the corpus data set with all of the diseases, each with its class and own list of symptoms.
2. allQueries.txt: This has all queries that work in our program. They must be inputted exactly as queries.
3. RelevanceTextDocument: This is a different formatted qrel file that we use in the program to gather relevance data for language models.

   For the next documents, these are required for the Bayes program.
4. diseaseclassedspaces.txt
5. testDiseaseClasses.txt
6. testDiseaseList.txt
7. diseaseTestSet2.txt
8. diseaseTrainSet2.txt

Pulling our Code

iv. Download team-4-master from GitLab (https://gitlab.cs.unh.edu/cs753-853-2019/team-4.git)

v. Extract the files to a suitable location.

vi. Change your working directory to ".../team-4-master/finalProject/".

Installing our Code

vii. Type *"mvn clean"* and press Enter. You will see a lot of output that should end in "BUILD SUCCESS" when the command has finished running.

viii. Type "*mvn install"* and press enter. You will see a lot of output that should end in "BUILD SUCCESS" when the installation and compilation is complete.

Running our Code

a. Creating TREC run files with our search similarities

i. Type *mvn exec:java -Dexec.mainClass="LuceneIndex"* and press Enter. The program will display some output as it builds, When **"WELCOME TO TEAM 4'S PROJECT!"** is displayed, the program is running and ready for user input.

ii. The program will give you several prompts in this order:

1. **Enter location of corpus, training, test, and relevance files:**. Type in the file path to the directory in which these files are located and press Enter.
2. **Enter location to save TREC run file(s):** Type in the file path to the directory where you would like to save your files and press Enter. **This must be a file path that already exists on your system.**
3. **Enter search similarity(bm25 or experiments):**Type the search similarity you would like to use and press Enter. "Experiments" will run all similarities.
    a. If you type an invalid input, the bm25 similarity will be selected.
        i. We will be using the bm25 similarity as our control for this project.
4. **Enter analyzer to use (default or custom):**Type the analyzer you would like to use for indexing and query parsing.
    a. If you type an invalid input, the default analyzer will be selected.
5. **Enter location to save index:** Type in the file path to the directory where you would like to save your index and press Enter.
    a. You may input a directory that does not exist and the program will create the directory for you. *HOWEVER, you **MUST** create the directory in a directory that already exists or the program will crash!*
6. **Enter location of query file:** Enter the location of your query file and press Enter.

iii. Wait for the index to build.**"Building indexes. Please wait..."** will be displayed while the index is built.

iv. When the index is done building, the SearchEngine will search it with your query. **"Performing searches. Please wait..."** will be displayed while it is searching.

v. The top 5 diseases along with all of their relevant symptoms will be displayed as a result to the user along with their relevance scores according to the selected similarity. You will also see the number of total results the search engine returned.