CS 421 Term Project ReadMe file

Team members: Arnold Lee, Vineet Singla, Davlin Stoy

Installation Directions:

Copy "Part2.java", "sqlite.bin" and "question.txt" into the LoPar executable directory.  
  
Copy "part2.lex" and "part2.gram" in the Data folder within the LoPar executable directory.  
  
Question.txt is the input question file. Three example questions are included. Feel free to add more or modify them.  
  
To run the query, run the command below while in the LoPar executable directory (assume the Java compiler is properly installed on Linux):  
  
javac part2.java (to compile)  
java part2 (to execute)

The output will locate in the file called “sql\_output\_final.txt”.

How the Program works:

We have used Java as the platform for the code used in the project. The code reads the Question.txt file located in the same directory, break the question sentence it into separate words, and forward those words into LoPar for parsing. Once the parsing tree is generated the result will be saved as “LoPar-output.txt” in the same directory.

When LoPar exits, the code then continues to read the parsing tree from the LoPar output, and extract the POS tags and store them into an array per question input. Then, we use a custom algorithm to transform the tags and words into a valid SQL query. Our algorithm is based on the Edmond’s maximum matching algorithm, which has wide used in the field of Computational Molecular Biology, and maximum likelihood estimation algorithm, which adjust the behavior itself base on previous question input in the same question file.

The result SQL query will be written to a file, and the code will forward it to another program called SQLite (assume the user has properly installed it per the instruction) which will look up the table and return the results according to the query. SQLite will write the result to an intermediate answer file, which it gets feed back into our code for post-result processing, like transforming answers such as “0” to “No”, and so on. The final result will be located in the file called “sql\_output\_final.txt”.

Problems we have encountered:

The hardest problem is to come out with the algorithm to transform the POS tag and word information into proper SQL query. By applying a maximum matching algorithm, this can essentially be expanded into use beyond the limit of the vocabulary given.

The second problem we have encountered is the sluggishness of the server and the lack of GUI interface. By dealing with command-line interface we have spent tremendous amount of time just to deal with File I/Os and interfacing with external programs (LoPar, SQLite) between different steps in our code.