CS2302 Data Structures

Fall 2019

Lab Report #4

Due: 21 October 2019

Professor: Olac Fuentes

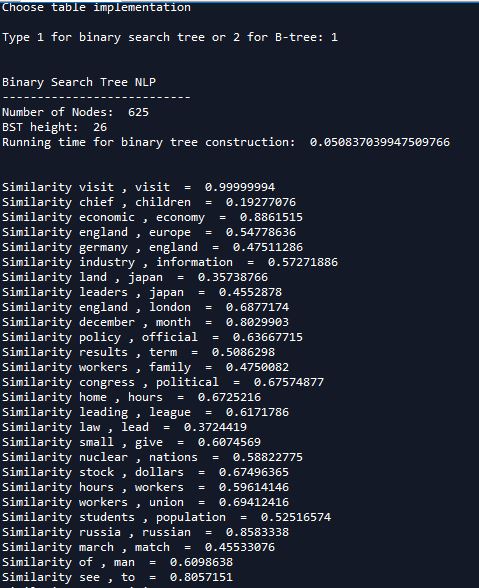
TA: Amindita Nath

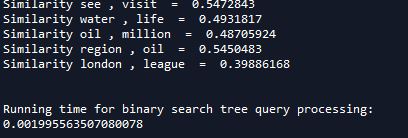
**Introduction**

The problem presented in this lab is relating to the similarity between words when AI process words, so given a large file of words with a list of floats that are embeddings for the word we have to implement a binary search tree as well as a Btree to search for the words and find their similarity. I created a Btree with attributes for the word and the embedding along with its normal attributes and created a Node which I used to implement into the Btree to hold the word and its embedding

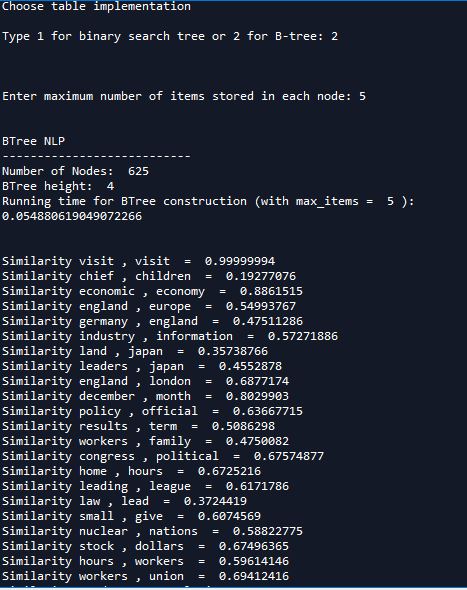
**Proposed Solution Design and Implementation**

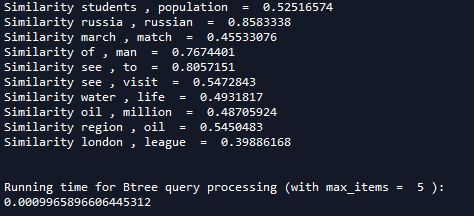
Design –

Binary Search Tree Output:



BTree output:





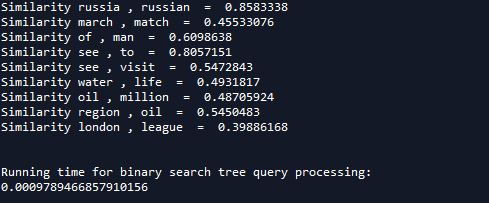
Operation #1: The first operation I completed was reading in the file which is where I read the whole file in and then split the file based on number because each word has 50 embedding floats, so I split them first into one attribute of the node and the next 50 into a different attribute of the node. Similarly, I read the pairs file and split the pair into different lists to use to test for similarity.

Operation #2: The second operation I completed was implementing a binary search tree. I inserted nodes, by comparing the value of the word, and used 2 separate attributes one for the word and the other for the 50-embedding list. Then I search for the node ,using the word, that has the words from the pair text and used the embeddings of those nodes to find the similarity between the pair of words.

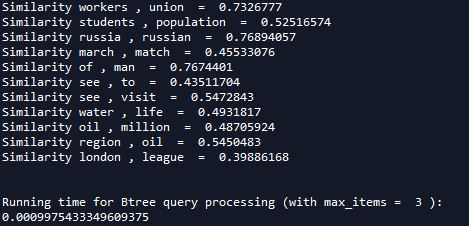
Operation #3: The third operation I completed was implementing a BTree. I made a node class that I insert into a data attribute of the Btree making a list of nodes. I then repeat what I did for binary search trees and search for the Node ,using the word, which is in the list of data and use the embedding of that node to find the similarity.

**Experimental Results**

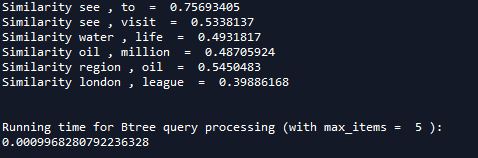
Binary Search Tree Query Time:



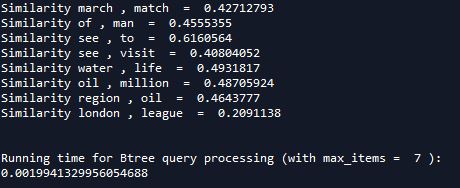
BTree Query Time (max items = 3):



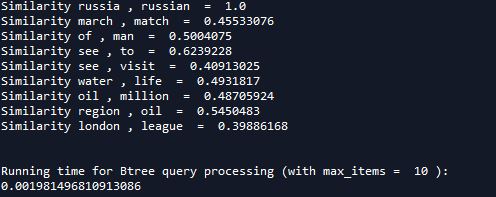
BTree Query Time (max items = 5):



BTree Query Time (max items = 7):



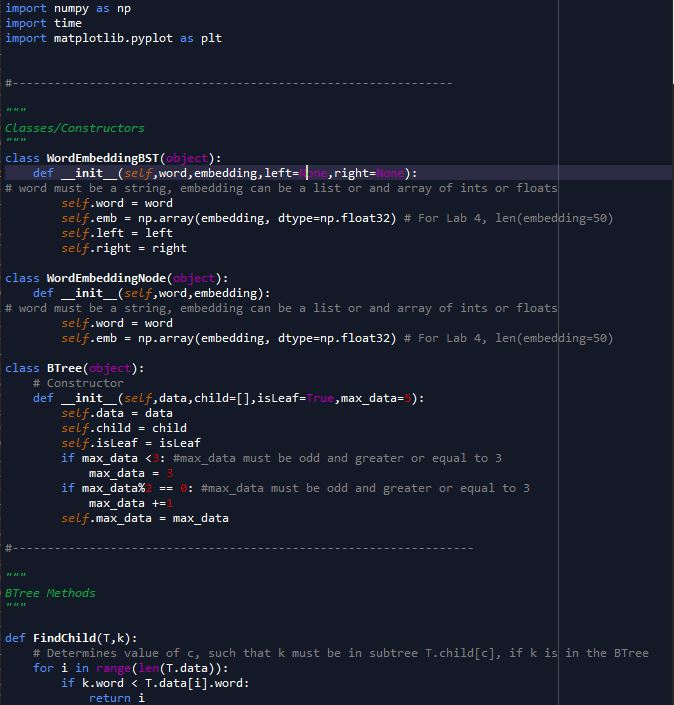
BTree Query Time (max items = 10):

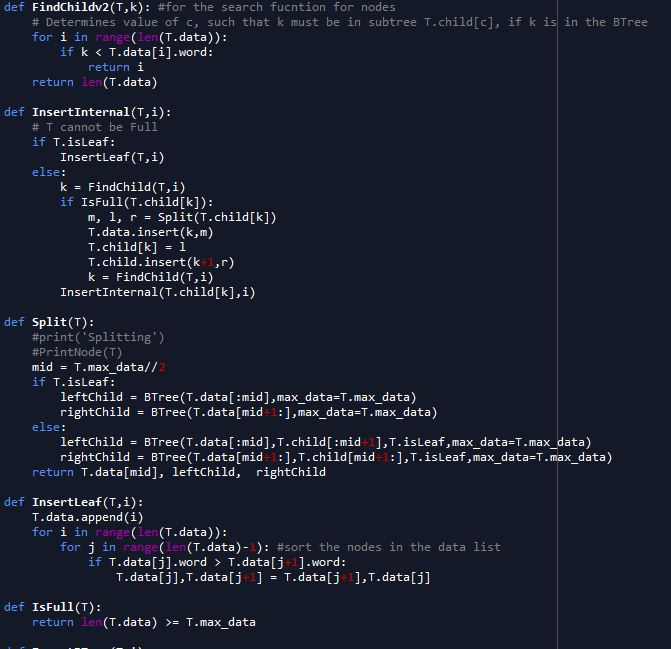


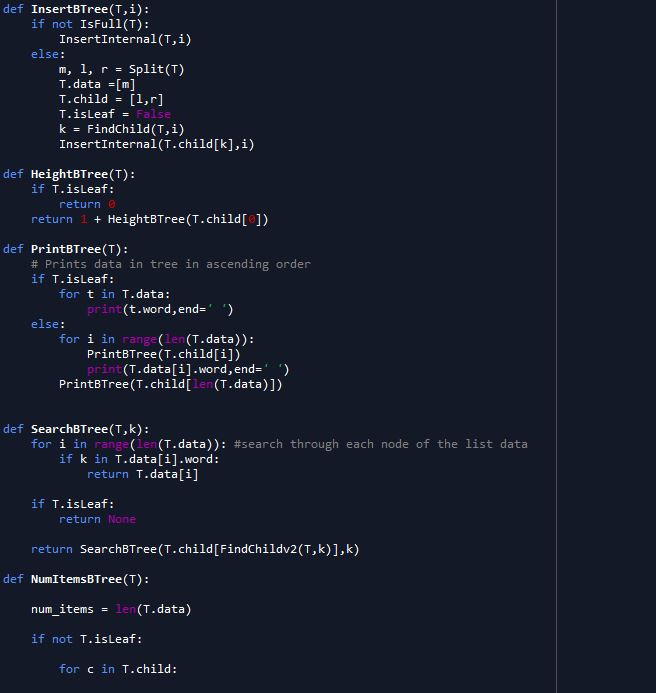
**Conclusion**

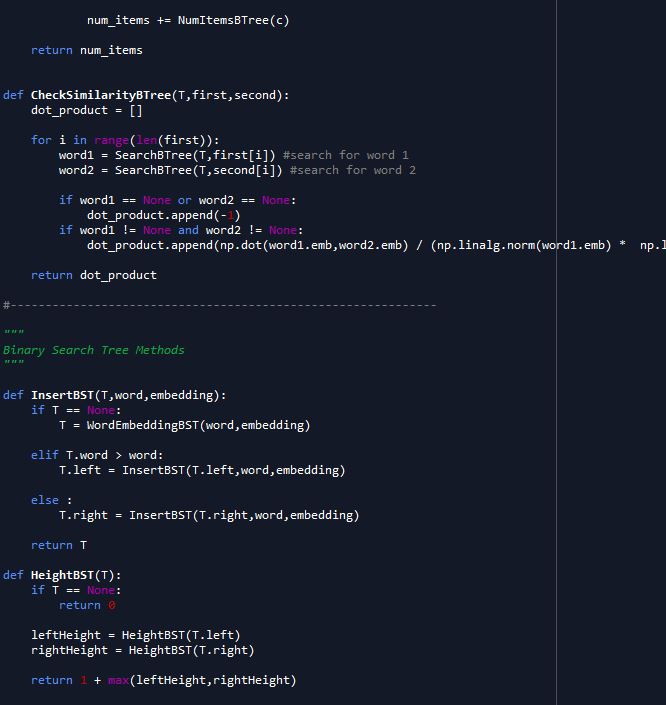
I learned how to efficiently work with both BTrees and binary search trees as well as learned how the parameters of a BTree can affect its efficiency. I learned a little bit about how AI process language at a surface level, I think using embeddings for each word and punctuation is a good way to determine the similarity of words since words can be related to each other and not look like one another.

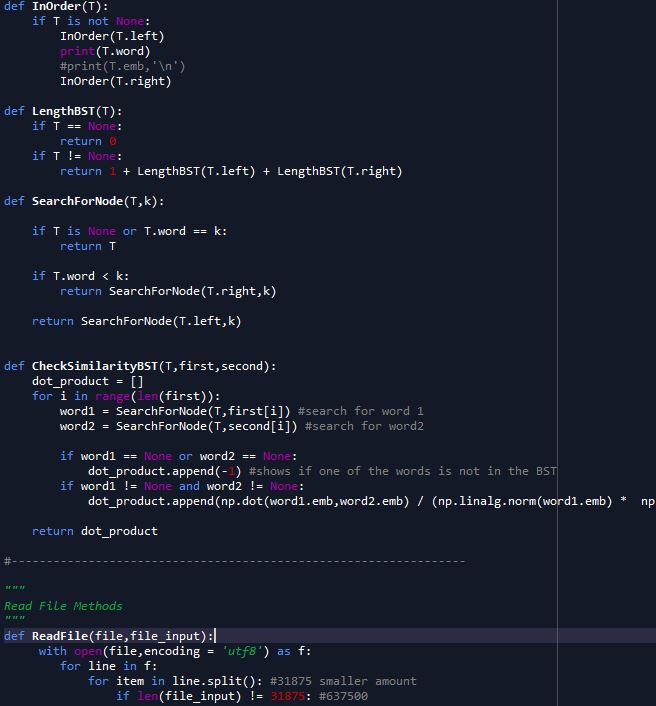
**Appendix**

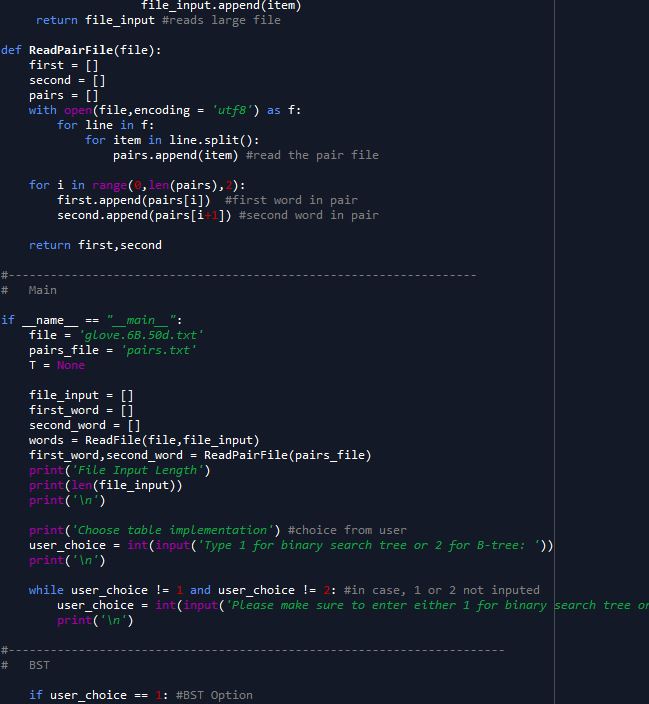
****

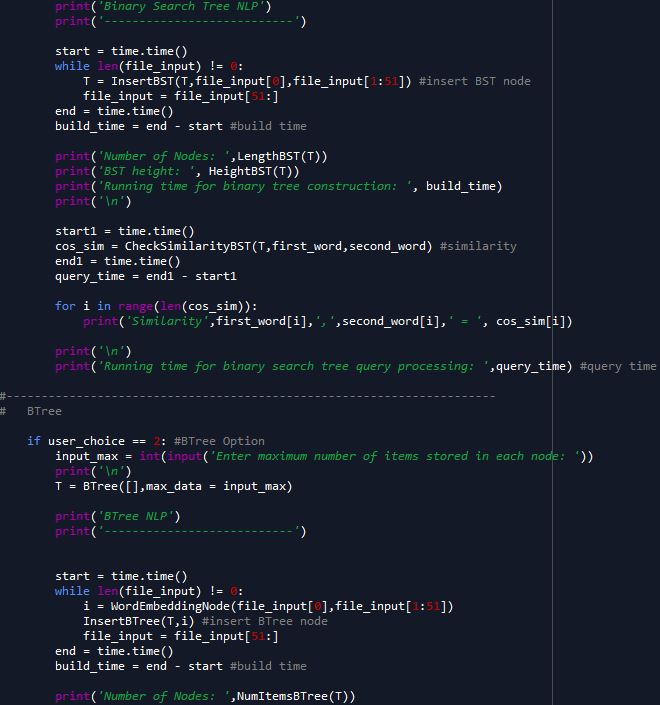
****

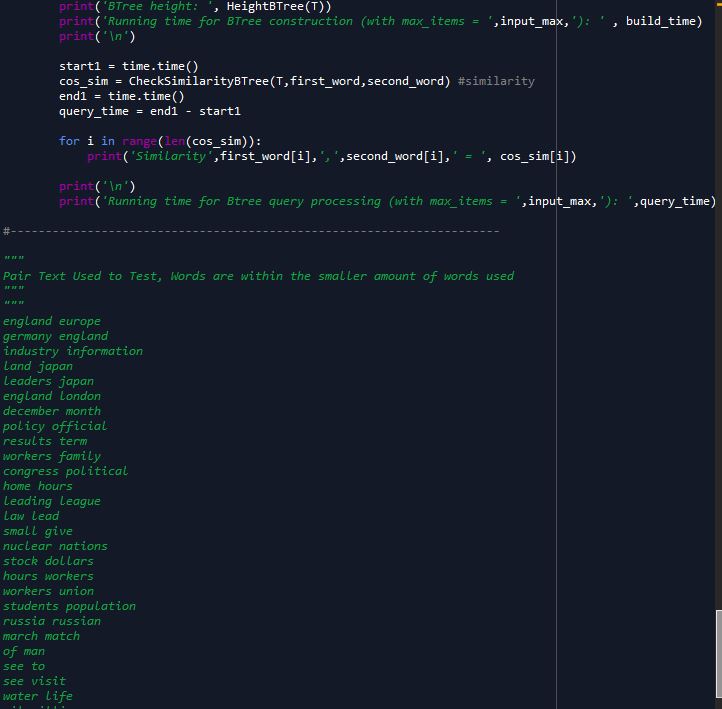
****

****

****

****

****

****

****

I certify that this project is entirely my own work. I wrote, debugged, and tested the code being presented, performed the experiments, and wrote the report. I also certify that I did not share my code or report or provided inappropriate assistance to any student in the class