Cooper Simpson & Zayna Sheikh
CSCI 2270 FInal Project

We present, as our final project for CSCI 2270, an application of weighted graphs for the creation of a novel phrase generation system. Using this data structure we were able to generate phrases from various distinct genres. The end result of our efforts is a working program that can be accessed and interacted with through the terminal. The user can choose from a selection of phrase genres indicating a starting word and a length, and based on this input the program will output a single pseudo-novel phrase within the selected genre. Partial inspiration for this project came from TA Alex Curtis. Help in accomplishing this goal was obtained from TA's Manish Shambu and Alex Curtis.

The graph data structure was the ideal tool for accomplishing our envisioned goal. The graph data structure consists of nodes and edges, where the edges connect various nodes together. Information pertaining to the data structure can then be contained in both the nodes and edges. A graph can be directed or undirected, and weighted or unweighted. In our implementation it was directed and weighted -- the reasons for which will be explained later in this document. To create the our proposed phrase generation system we sought large data-sets for specific genres of phrases (wise quotes, book titles, newspaper headlines, etc.). Parsing these data sets we were/are then able to create graphs to represent the flow of words. In each specific phrase in the dataset words were were read in a frame of two. That is each possible pair of words was read from a phrase. New nodes would be created in the graph for each new word encountered, and an edge would be added between word pairs. If an edge already existed then weight was added to that edge. This brings us back to the motivation for a directed and weighted graph. Direction is needed because there is a specific order of which word comes first in a pair. One word follows the other and it may not be the same the other way around. Weight is needed to represent how many times a pair of words are encountered and thus informs the probability (discussed more later). Each node and edge was created using a defined struct. Nodes contain data of the word, a list of its edges, and the total weight of these combined edges. Edges contain data on the word it is going from and going to, its weight, and its range (for probability purposes). Having now thoroughly expounded upon the graph structure, and the process of creating it we can now present the method for generating a phrase. By traversing the graph from a starting node informed by the user input until a certain end criteria is met (length, word) one may output a phrase. The important question here is, how to choose the path? By utilizing the weights of the edges an informed decision about which node to go to next. If an edge has a large weight then that pair of words is common in the data set. This means that the large the weight the more likely you are to go from the word you are on to the word that edge connects you with. In our program this is achieved by maintaining the weight of an edge, as described earlier, as well as maintaining the total weight of all edges leaving a node. Using this, one can then calculate a probability of taking a certain edge as the path for all edges. This is done by iterating through all edges and assigning a range of values to each edge according to its weight. The breadth of these values is given by the weight of that particular edge. For example, a node with weight 10 would have a range of 10. Effectively what one is doing is dividing the edges weight by the total weight of its origin node. A random number can then be generated and then whatever edge has the range that this number falls in is the edge that will be traversed. While this method does not take into account a large amount of other important linguistic information it does provide some probabilistically informed output.

As a user one may interact with the program through the command line and will be greeted with the following menu below. Selecting an option for a phrase will prompt the creation of a graph for that specific phrase. A user can then continue to create phrases as they wish. Output of various phrases is given below.

```
coopers-MacBook-Pro:Project forrestoandcoopie$ g++ -std=c++11 finalProject_driver.cpp phraseGraph.cpp -o run
coopers-MacBook-Pro:Project forrestoandcoopie$ ./run
Welcome to the Weighted Graph Phrase Generator
____ Please Choose a genre: ____
1. Newspaper Headlines
2. Book Titles
3. Movie Titles
4. Quiplash Prompts
5. Wise Quotes
6. Quit

1
Loading Newspaper Headlines Generator...
File read, graph built.
Please enter the first word: foreign
Please choose a length between 5 and 10 words: 10

Your generated phrase:
phrase:

foreign buyers exit


2
Loading Book Title Generator...
File read, graph built.
Please enter the first word: lord
Please choose a length between 2 and 5 words: 4

Your generated title:
phrase:

lord of the sun

5
Please enter the first word: art
Please choose a length between 5 and 20 words: 10

Your generated quote:
phrase:

art is a family actually, in art is an mgm
```