# DM Assignment 2

Shruti Wasnik (Roll no :20111062)

November 2020

## 1 Introduction

The dataset has been taken from `http://snap.stanford.edu/data/wikispeedia.html`. *Wikispeedia* is an online human-computation game that has been used to collect details of human navigation paths on *Wikipedia*. In this game, a player needs to navigate and reach an article from some other unrelated article by clicking on the Wikipedia links. The report diagrams some analysis of the concepts used in this game. Python programming language has been used to prepare the scripts. A manual named `README.txt` is attached which describes how to use the code and run the assignment. It includes all the necessary details about the programs, their plugins, and dependencies needed to run the program. The undirected graph has been used only in question 5, for others, directed graph has been used.

## 2 Analysis

The game is used to analyze the way in which the players navigate from source to target article. If the players chose to navigate via unrelated article, they may need to navigate more nodes in order to reach the target, or they may never reach the target. On the other hand, if they navigate through related articles, they may reach the target faster. The best path would be the shortest path possible to reach the target. It is not necessary that a player always chooses the shortest path for the navigation. The assignment analyzes the human traversed paths and also evaluates the shortest possible paths and distances from any source to target article. Here is the dataset statistics taken from the specified website:

| Dataset statistics | |
|---|---|
| Finished paths | 51,318 |
| Unfinished paths | 24,875 |
| Articles | 4,604 |
| Links | 119,882 |

## 2.1 Question 1

In this question, all the articles given in `articles.tsv` are assigned unique ids starting from `A0001` to `A4604`. The output would be stored in `csv-files/article-ids.csv`.

## 2.2 Question 2

In this question, all the categories given in `categories.tsv` are first converted into a hierarchy with `subject` as root node and are then assigned unique ids starting from `C0001` in breadth-first manner (sorted alphabetically). The hierarchy has three levels. The output would be stored in `csv-files/category-ids.csv`.

## 2.3 Question 3

The file `categories.tsv` provides the details of the categories in which an article falls into. We use this mapping and convert the article names and their corresponding category names into article ids and category ids, respectively. If no category has been specified to an article, it has been assigned to the root category `subject` with category id `C0001`. The output would be stored in `csv-files/article-categories.csv`.

## 2.4 Question 4

The file `shortest-path-distance-matrix.txt` contains a matrix specifying the shortest path distance from a source to target. This matrix has been used to construct the directed graph of articles in an edge adjacency list format. If the path distance is 1, this means that the nodes are directly connected to each other. Thus, these entries are used for finding the edges. There are total 119772 edges, and the output would be stored in `csv-files/edges.csv`.

## 2.5 Question 5

The output of previous question, i.e., `csv-files/edges.csv` has been used to create an undirected graph to find the connected components. The `networkx` library is used for all these works. Firstly, all the nodes (article ids) are added to to the graph. Then the edges specified in `csv-files/edges.csv` are then added. We then find number of nodes, number of edges and the diameter of each connected components. There are around 12 isolated nodes, i.e., they are not connected to any node.This means, that these articles can never be reached from any other article. For such nodes, the number of edges and diameter is specified as 0.

## 2.6 Question 6

Finished paths as specifies in the file `paths_finished.tsv` provides the details of the human paths which have successfully reached the target articles from the

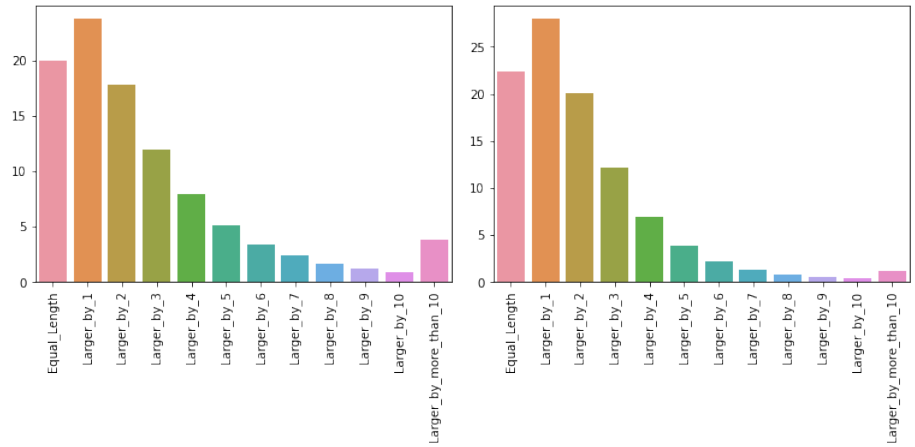
source. These paths need not necessarily be the shortest path. In this question, we have compared the path traversed by the human (for both, with and without back-clicks) with the actual shortest path possible between the nodes. A graph has been created (as in previous question), and for each path, length of shortest distance paths has been calculated using the predefined function of `networkx` library.

## 2.7 Question 7

This question uses the output files of question 6 to find the percentage of human paths which have:

- exactly same path length as shortest path
- path length is 1 to 10 more than the shortest path (each separately)
- path length is 11 or more than the shortest path

The following graphs depict the observations:




(a) percentage-paths-back (b) percentage-paths-no-back

In both the cases, there are more human paths which have path length 1 more than shortest distance.

## 2.8 Question 8

This question can help in analyzing the number of paths and number of times any category has been traversed in both human paths as well as in shortest paths. Categories of source article and target article can be found using the output file of question 3. Observations shows that the category `subject.Countries` with category id `C0005` has been visited the most in both shortest paths as well as in human paths. Following is the data obtained for category `subject.Countries`:

```
Unnamed: 0                          C0005
Number_of_human_paths_traversed     28375
Number_of_human_times_traversed     41955
Number_of_shortest_paths_traversed  20669
Number_of_shortest_times_traversed  24578
```

## 2.9 Question 9

Here we repeat the same calculation as in previous question, but it this question, for each category, we consider its subcategories as well. Since the category `subject` is the root category, it has been traversed the most. Here is the details obtained for the category:

```
Number_of_human_paths_traversed      51307
Number_of_human_times_traversed     305645
Number_of_shortest_paths_traversed   51306
Number_of_shortest_times_traversed  197328
```

The second most traversed category (or subcategory) is `subject.Geography`, and the data obtained for this category is:

```
Number_of_human_paths_traversed     39146
Number_of_human_times_traversed     99047
Number_of_shortest_paths_traversed  37114
Number_of_shortest_times_traversed  61255
```

## 2.10 Question 10

In this question, we use the data given in file `paths_unfinished.tsv` to find the source and destination category pair taking into account all the sub-categories under it and find the percentage of finished and unfinished paths for each category pair. This can help us in analyzing the relationship between each category pair for both finished and unfinished paths. There are many pairs with percentage_of_finished_paths =100. The pair with highest percentage_of_unfinished_paths =94.44 is `(C0009,C0080)`

## 2.11 Question 11

In this question, we needed to find the average ratio of length of human (without back clicks) paths to shortest paths for every source-destination category pair. This can be used for comparing the length of human traversed and shortest paths. The largest ratio of 17.7 has been found for the category pair `(C0041,C0090)`, while the lowest ratio is 1 and has been obtained for many pairs.

# 3 Conclusion

The main motive of this game is to analyze the visits on each article and each category of the Wikipedia links. This analysis can be used for finding the trends in which people visit any link. We can find the shortest path to visit any link, relationships between the categories, the most and the least visited articles and categories, etc.

# 4 References


- http://snap.stanford.edu/data/wikispeedia.html
- http://infolab.stanford.edu/~west1/pubs/West-Pineau-Precup_IJCAI-09.pdf