Big Data Assignment 2 Report

Mapper T1

The mapper loads the input file and reads each line from the standard input. Each of this line is split and the source node and destination node are separated and the source node is made the key and the destination node the value as the key-value pair.

Reducer T1

The path of the local file v is noted from the command line argument and the file is opened. The current_key variable will store the current key and is initialised to None and the string variable s which stores the source node and the list of adjacent nodes and is initialised to empty string. Each line from the standard input and the key, value pairs are split. If the current_key is None then the key is assigned to current_key and the variable s is assigned the key and the value in the given format. The key value is stored into the file v by assigning it 1 as the initial page rank. If the key matches the current_key, then the variable s is updated with the adjacent node for that key. The variable s is printed to the standard output. The file v is closed after the process.

Mapper T2

The similarity function calculates the similarity according to the formula. We have used loop unrolling to improve the efficiency. It takes in the page rank file and page embedding file and stores it in the memory. This is done using the open and json.loads functions and saved in the memory in the form of a dictionary. We go through each line from the standard input using the sys.stdin and a for loop. Each line in the standard input is split using the split

function into p_node and a list of q_nodes in a list. We print the p_node and a contribution of zero to handle the case of no incoming links. For each node in q_nodes we print the q_node and the contribution of q_node to p_node using the given formula.

Reducer T2

Variables sum and current_node_id are initialised with 0 and None respectively. For each line from the standard input line is split and the contribution values typecast to float and are stored in ind_contrib and the node id is stored in node_id. If node_id is the same as the current node_id then ind_contrib is added to the sum. If node_id is not equal to current_node_id the page rank is calculated using the given formula with sum as the total contribution to the node, node_id and the page rank is printed after rounding the page rank to 2 decimal places. Also sum and current_node_id are initialised to ind_contrib and node_id respectively. If the current node_id is none then, current_node_id is node_id and ind_contrib is added to sum.