*Construction of graph*

i. Nodes

We use the result of preprocessing in the previous steps to construct our sentence graph.The nodes represent the sentences

and their weight is equal to p\_neural.Weights assigned to nodes will be useful after we obtain strongly connected

components in the graph.

ii. Edges

The weight of the edge between sentences iand j is equal to the value of the entry Sij of similarity matrix S.

iii. Eliminating edges

Next we try to eliminate edges that have weight less than a treshold limit. The purpose of this step is to eliminate the

redundant edges and consider only those that truly indicate a link between various sentences.

iv. Strongly connected components

Next our aim is to obtain strongly connected components in the graph. This step helps us to obtain various components in the

article that are linked together in terms of the information they convey. It also helps us to select the sentences in a manner

that they represent the length and breadth of the article and not focused to a specific section of the We use the Tarjan’s algorithm

for this purpose text. In the following stages we work on these scc’s separately and also consider the weight assigned to each node in the neural network stage.

v. Ensuring smooth flow of information from one sentence to the other We apply the following concepts of

graphs to further ensure that our summary is well linked and sentences that finally occur in the summary convey

information in a manner that flows naturally rather than receiving discrete components of the article in an abrupt way and

not able to understand the linkage between them.

Now flow of information can be divided into a hierarchy

a) Flow among sentences in a scc (that represents a logically distinct section of the article) to enable the user to understand the subtopic thoroughly.

b) Flow among various scc’s to be able to link distinct sections of article First we consider part a) Scoring sentences within an scc. Let us say that the number of sentences in the scc is n.For each scc we consider the first k(=1 to n/5 in order that they appear in original article) sentences and pick the node(sentence) with maximum weight among them. We do this to make a presumption

that this sentence will surely get include in the summary as it introduces a subtopic. We label this as the first sentence of this

scc(subtopic)and follow a similar procedure to obtain the last sentence for this subtopic(scc) .Now we have to devise a

way to find the best ordered sequence of sentences from this first to the last and give them extra points for providing a smooth flow.

Longest path algorithm with some modification Considering intuitively a smooth flow can be ensured by simply using a

recursive implementation. Find the sentence that is best linked with this sentence and occurs after it in the original article. Do this

recursively for each sentence. Since linking between two sentences is related to their weight the higher the weight between two

sentences the longer is the path between two nodes .In this way we are basically doing a bfs traversal to find the longest path between the first and the last sentence in the scc considered. But, the modification is due to the following two restrictions

:

1. Only those neighbors of the currently visited node (sentence) will be considered that occur after it in the original article.

2. The longest path must not have more than a predefined number of nodes as that will increase the number of sentences in the

summary beyond proportion. So, we keep track of the count of the node that is currently extending a longest path and once it increases the predefined limit we don’t consider it any further. Sentences obtained from such a sequence will be given points

(predefined constant) in addition to those obtained in the neuralnetwork stage=p\_intrascc.

Now we consider part b)

To obtain sentences connecting together two distinct subtopics, we need to construct another graph and apply the same algorithm as above. This is how we construct the graph.

1. Now we have to select from sentences in between the subtopics (or scc’s).So we consider scc’s in the increasing order of their position which is the same as the position of the first sentence in it Let us say that there are total k scc’s.

Hence For I varying from 1 to k-1:

Consider the last node (sentence) in scc I and

first in scc I+1)

Construct a graph with source node as last sentence and destination as first. Use all sentences between them as intermediate nodes and edge weights as computed previously.Applying longest path algorithm as done previously we select sentences between these two scc’s. These sentences are again given additional points=p\_interscc (predefined constants).