TEXTRANK ALGORITHM

The textrank algorithm models the data text as a graph where the sentences are the NODES and the EDGES between the nodes are the similarity of two sentences.
Similarity of senetences is computed by some function.

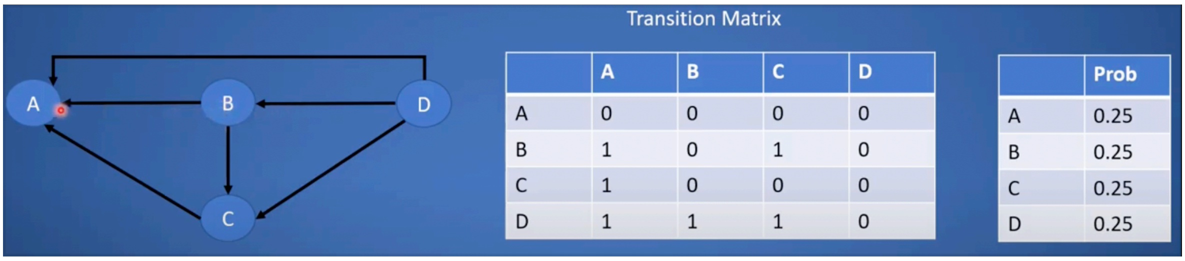
Sentence S(i) = w1(i)+w2(i)+w3(i)+w4(i)+...wn(i)
Sentence S(j) = w1(j)+w2(j)+w3(j)+w4(j)+...wm(j)

Similarity(Si and Sj)= ({ wk | wk belongs to Si AND Sj }) / (log|Si| + log|Sj|)

now a graph is created with —>   sentences as NODES
—> similarity as EDGES
from this graph, the PAGERANK algorithm is used to compute the importance of each NODE. The most important sentences are selected and presented in the same order as in the original data text.

PAGERANK ALGORITHM

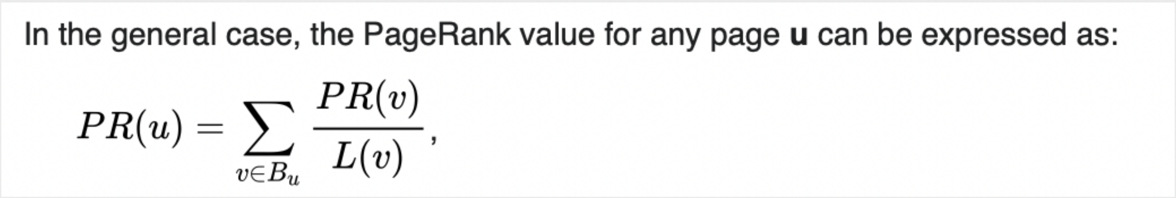
THe pagerank algorithm is used by Google Search **to rank web pages in their search engine results. It** gives a probability distribution of the likelihood that a person randomly clicking on links will eventually stop clicking and arrive at any particular page.let's say there are 4 websites A,B,C,D and the given graph shows the transition links between them.



Transition Matrix

	A	B	C	D
A	0	0	0	0
B	1	0	1	0
C	1	0	0	0
D	1	1	1	0

	Prob
A	0.25
B	0.25
C	0.25
D	0.25

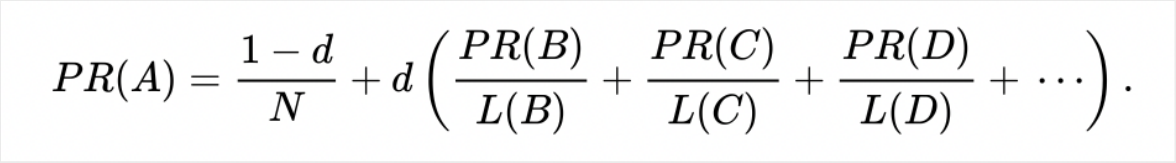


In the general case, the PageRank value for any page **u** can be expressed as:

$$PR(u) = \sum_{v \in B_u} \frac{PR(v)}{L(v)},$$

the PageRank value for a page **u** is dependent on the PageRank values for each page **v** contained in the set **B$_{\mathbf{u}}$** (the set containing all pages linking to page **u**), divided by the number *L*(*v*) of links from page **v**.

the probabilty, at any step, that the surfer will continue clicking is called the damping factor d(~0.85).

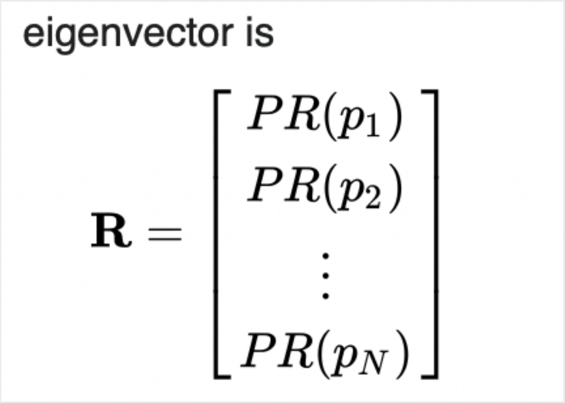


$$PR(A) = \frac{1-d}{N} + d\left(\frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)} + \cdots\right).$$

number of documents/sentences (*N*) in the collection

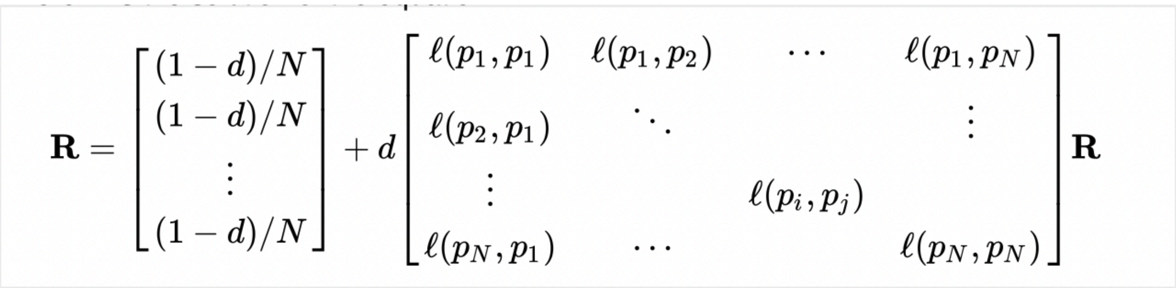
<u>in matrix notation</u>

The PageRank values are the entries of the eigenvector such that each column adds up to 1.



eigenvector is

$$\mathbf{R} = \begin{bmatrix} PR(p_1) \\ PR(p_2) \\ \vdots \\ PR(p_N) \end{bmatrix}$$



$$\mathbf{R} = \begin{bmatrix} (1-d)/N \\ (1-d)/N \\ \vdots \\ (1-d)/N \end{bmatrix} + d \begin{bmatrix} \ell(p_1,p_1) & \ell(p_1,p_2) & \cdots & \ell(p_1,p_N) \\ \ell(p_2,p_1) & \ddots & & \vdots \\ \vdots & & \ell(p_i,p_j) & \\ \ell(p_N,p_1) & \cdots & & \ell(p_N,p_N) \end{bmatrix} \mathbf{R}$$

ℓ(pi,pj) is the ratio between number of links outbound from page j to page i to the total number of outbound links of page j. the elements of each column sum up to 1.

PAGERANK COMPUTATION

PageRank can be computed either iteratively. The data undergoes iterations to assign more accurate pagerank value to each node.

Initially the probability distribution is assumed to be evenly divided among all the nodes. (t=0)

$$PR(p_i;0) = \frac{1}{N}.$$

at time t

$$PR(p_i;t+1) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j;t)}{L(p_j)}$$

in matrix notation

$$\mathbf{R}(t+1) = d\mathcal{M}\mathbf{R}(t) + \frac{1-d}{N}$$

PAGERANK FOR TEXTRANK

Transition matrix ———–—> Sentence Similarity matrix
Similarity can be calculated by various methods like longest common substring, cosine distance etc
the page rank algorithm is applied on the sentence similarity matrix which sort the sentences based on their pagerank values. Then the top N sentences can be extracted as the text summary.