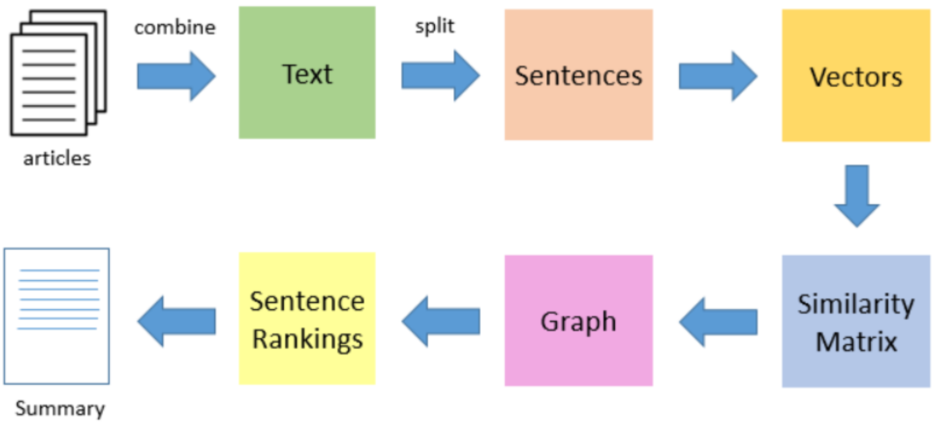
NLP-Text Summarization using TextRank Algorithm

Similarities between TextRank and PageRank Algorithms

1) In place of web pages, we use sentences
2) Similarity between any two sentences is used as an equivalent to the web page transition probability
3) The similarity scores are stored in a square matrix, similar to the matrix used for PageRank

TextRank is an extractive and unsupervised text summarization technique



Before Preprocessing split the text into sentences

Data Preprocessing

1. Converting all text to lower case for further processing
2. Parsing HTML tags
3. Removing text between () and []

4. Contraction Mapping — Replacing shortened version of words (for e.g. can’t is replaced with cannot and so on)
5. Removing apostrophe
6. Removing non-ASCII characters
7. Removing punctuations and special characters
8. Removing stop words using nltk library
9. Retaining only long words, i.e. words with length > 3
10. Removing URLs

Vector Representation of Sentences

1) Fetch vectors of size 100 elements for constituent words in a sentence

2) Take mean/average of those vectors to arrive at a consolidated vector for the sentence

Similarity Matrix

Create an empty similarity matrix and populate it with cosine similarities of the sentences and initialize the matrix with cosine similarity scores to compute similarity between a pair of sentences

Applying PageRank Algorithm

Convert the similarity matrix into a graph. The nodes of this graph will represent the sentences and the edges will represent the similarity scores between the sentences. On this graph apply the PageRank algorithm to arrive at the sentence rankings .Number of keywords extracted is relative to the size of the text (a third of the number of nodes in the graph) Adjacent keywords in the text are concatenated into keyphrases

Summary Extraction

Extract the top N sentences based on their rankings for summary generation.

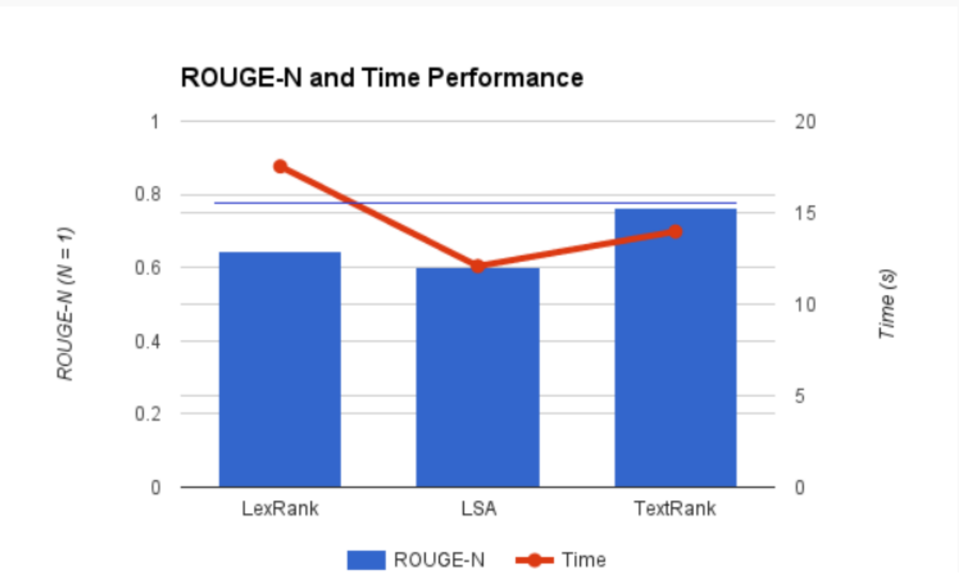
Conclusion:

Why TextRank?

To evaluate the performance of the algorithms, we need a set of ‘gold’ keywords to compare the extracted keywords against. Here i used the human generated summaries to extract most frequent keywords. Only the keywords that appeared in more than half of the summaries were selected as ‘gold’ keywords for that particular article. I also measured the running time of each of the algorithms on the dataset.

The evaluation metric we used was ROUGE-N metric from the ROUGE toolkit. Since we are considering only single-word keywords, N is set to unity. In essence, the ROUGE-1 parameters simply computes the ratio of number of keywords correctly indentified (i.e. keywords in sample that also occur in the gold set) to the total number of gold reference set keywords.

The following image summarizes the result



The blue horizontal line denotes the ROUGE-1 score of human generated summaries calculated by k-fold cross validation of each of the summa for every topic. As we can see, TextRank came out to be the winner. With a modest running time and a near-human ROUGE-1 score, TextRank was finally selected as our keyword extraction technique.