POC – Text Summarization

The Summarization includes applying two rules – Lexical Cohesion and Text Ranking

Lexical Cohesion -

Coherence in linguistics makes the text semantically meaningful. It is achieved through semantic features such as the use of deictic (a deictic is an expression which shows the direction. ex: that, this.), anaphoric (a referent which requires an antecedent in front. ex: he, she, it.), cataphoric (a referent which requires an antecedent at the back.), lexical relation and proper noun repeating elements (Morris and Hirst, 1991). These rules are then used to create meaningful chunks of sentences.

Rule 1 : The presence of connectives (such as accordingly, again, also, besides) in present sentence indicates the connectedness of the present sentence with the previous sentence. When such connectives are found, the adjacent sentences form coherent chunks.

Rule 2 : A 3rd person pronominal (such as presence of pronouns line He, She, It, They, etc.) in a given sentence refers to the antecedent in the previous sentence, in such a way that the given sentence gives the complete meaning with respect to the previous sentence. When such adjacent sentences are found, they form coherent chunks.

Rule 3 : The reappearance of nouns in adjacent sentences is an indication of connectedness. When such adjacent sentences are found, they form coherent chunks. For this rule we used a combination of **UnigramTagger,BigramTagger and TrigramTagger on treebank corpus** to train and this tagger was then used on the article to find nouns.

Rule 4 : Similar words found in the adjacent sentences means that the sentences are coherent in nature.

$$LC_{sa,sb} = (rule1_{sa,sb} + rule2_{sa,sb} + rule3_{sa,sb} + rule4_{sa,sb})/4$$

Text Ranking -

The proposed graph based text ranking algorithm consists of three steps:

(1) Words Affinity Analysis;
(2) String pattern based weight calculation algorithm;
(3) Ranking the sentences by normalizing the results of step (1) and (2).
(4) Combining it with the result of Lexical Cohesion step.

Step (1): Let the set of all sentences in document S= {si | 1 ≤ i ≤ n}, where n is the number of sentences in S. The sentence weight (SW) for each sentence is calculated by average affinity weight of words in it. The affinity is calculated very similar to Rule 3 of Lexical Cohesion but this time we will also average the words matched with total words found.

Formula Applied -

$$SW_(sa) = \sum_{i}^{n} wordsMatched(si, sj)/totalWords(sa)$$

Step(2): Using edit_distance function to find distance of 2 sentences. The edit_distance works by inserting, deleting one character at a time to create the another sentence. Example - **(train, gain)** will have **distance 2 as in first step**, **t changes to g and in next step r is removed.**

$$ED_(sa) = \sum_{i}^{n} editDistance(sa, si)/length(sa)$$

Step(3): Take average of the 2 steps above to find a normalized value.

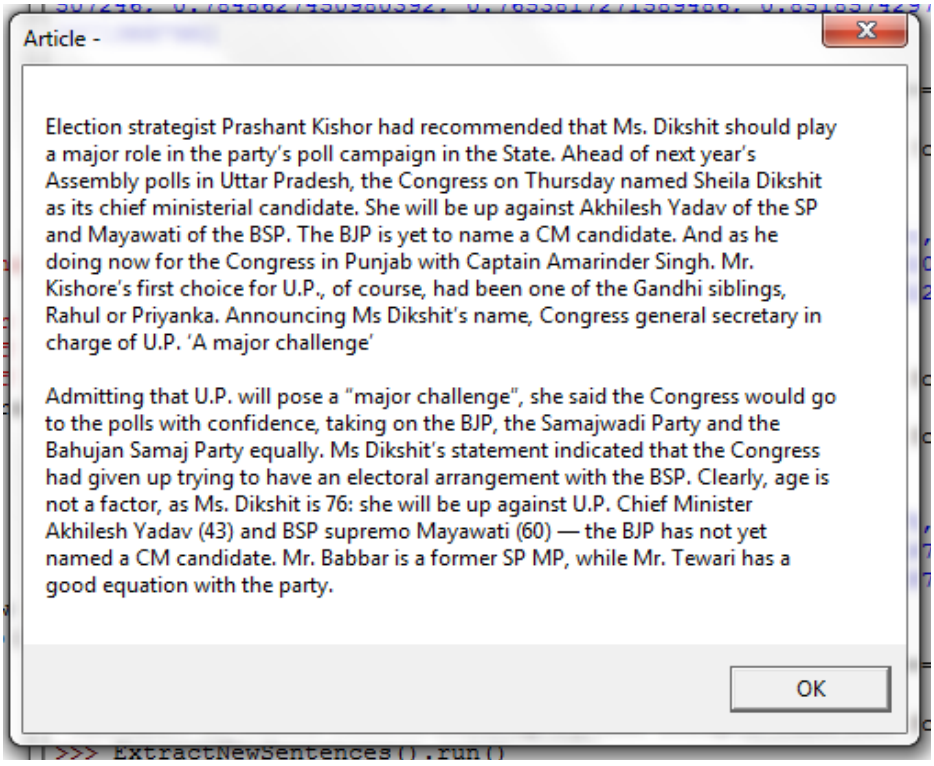
$$NV_{TR} = (SW + ED)/2$$

Step(4): Combine the results of Cohesion step and Text ranking. In my code I have given 0.35 weight to LC and 0.65 weight to NV.

$$finalScore = 0.35 * LC + 0.65 * NV_{TR}$$

Ouputs -

1. http://www.thehindu.com/news/national/sheila-dikshit-named-as-congress-chief-minister-candidate-in-uttar-pradesh/article8849432.ece?w=alauto



2. http://www.thehindu.com/news/national/charge-sheet-in-agusta-scam-to-be-filed-this-year-government-to-supreme-court/article8853920.ece