Automatic Text Summariser

November 23, 2015

Anand Bhoraskar (130050025)
Pradyot Prakash (130050008)
Maulik Shah (13D100004)

Approach

- 1. Document Preprocessing
- 2. Graph Construction
- 3. Ranking Algorithms
- 4. Summarizations

Graph Structure

- Every sentence is represented by a node in the graph
- The weights represent the extent of similarity between the connected sentences
- We choose from 4 available evaluation functions to populate the edge weights
- Then we have two options:
  - Add weights of edges connected to a node to judge its importance
  - Use TextRank algorithm (inspired from Google's PageRank algorithm)

Evaluation Functions Used

Cosine Similarity (Using Bags of Words as Vectors)

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

Dice Similarity

$$Dice(s_a, s_b) = \frac{2|w(s_a) \cap w(s_b)|}{|w(s_a)| + |w(s_b)|}.$$

Rada Mihalcea's Similarity Measure

$$Similarity(S_i, S_j) = \frac{|\{w_k | w_k \in S_i \& w_k \in S_j\}|}{log(|S_i|) + log(|S_j|)}$$

$$Jaccard(s_a, s_b) = \frac{|w(s_a) \cap w(s_b)|}{|w(s_a) \cup w(s_b)|}.$$

TextRank Algorithm

TextRank gives a weight to each node in the graph by running the following formula till convergence using one of evaluation functions to find edge weights.

$$WS(V_i) = (1-d) + d * \sum_{V_j \in In(V_i)} \frac{w_{ji}}{\sum_{V_k \in Out(V_j)} w_{jk}} WS(V_j)$$

Where d is a damping factor that can be set between 0 and 1, which has the role of integrating into the model the probability of jumping from a given vertex to another random vertex in the graph.

Undirected edges are considered as two way directed edges.

Learning from Project

- We learned that finding a large relevant corpus for supervised learning can be quite difficult for any project. That led us to change our approach significantly to heuristic based algorithm.

- There are two types of summarization : Abstract based summarization and Extract based summarization. While exploring them, we realised how vastly these two seemingly simple algorithms differ in terms of simplicity. While Extract based algorithms simply extract a few complete sentences out of the text by assigning them weights, Abstract based algorithms is currently a topic under research and requires hardcore NLP.

Suggestions for Future Work

- There are various features that can be added to an automated summarizer, including
  - Supervised learning models
  - Abstract based summarizer
- Adding a user friendly interface
- Adding features to summarise web pages

Bibliography

- PageRank : https://en.wikipedia.org/wiki/PageRank
- TextRank: https://web.eecs.umich.edu/~mihalcea/papers/mihalcea.emnlp04.pdf
- Himanshu Jindal’s Summarizer: https://github.com/himanshujindal/Automatic-Text-Summarizer