Text Summarization

|  |  |
| --- | --- |
| 000  001  002  003  004  005  006  007  008  009  010  011  012  013  014  015  016  017  018  019  020  021  022  023  024  025  026  027  028  029  030  031  032  033  034  035  036  037  038  039  040  041  042  043  044  045  046  047  048  049 | 050  051  052  053  054  055  056  057  058  059  060  061  062  063  064  065  066  067  068  069  070  071  072  073  074  075  076  077  078  079  080  081  082  083  084  085  086  087  088  089  090  091  092  093  094  095  096  097  098  099 |

|  |
| --- |
|  |
| Jorgie Bartelsi Permana - 1301153627  Puruso M. Hanunggul - 1301153680 |
|  |
|  |
|  |
|  |
|  |

Abstract

This document describes the text summarization and how it works, in an extractive way. This document also shows the results of text summarization research using some data sets and showing the results of its evaluation.

Credits

This project is based on “Textrank by davidadamojr” in github.

Introduction

In today's digital age, technology can be found everywhere to simplify human life, one of them is to facilitate the search for information and knowledge. Generally, information can be found in the form of articles.

The existence of technology makes it easy to read articles anywhere, anytime whether it is online or conventional. However, many people are reluctant to read an article. They prefer reading more interesting readings than reading an article.

With the summary of an article, can facilitate the reader to capture the information contained in the article.

TextRank

TextRank is an algorithm that uses Google’s PageRank to create an extractive summary of a text. As PageRank is based on graph, TextRank turn an entire text into graph, so PageRank can process it, then the result of PageRank is sorted and sentences with best scores will be used as the summary.

To create the graph, Text rank tokenizes the entire text by sentences, this mean that a tokens are consisted of a sentence from the text. Then it created an undirected graph with the tokens as nodes and combination is used to create edges, the weight of edges is calculated by using string distance formula. The most commonly used string distance is Levenshtein distance, but we also included Longest Common Substring(LCS) as an alternative. The graph will be used in PageRank to create a list of token and their scores. Top scored tokens will be used as the summary.

## Levenshtein Distance

The Levenshtein distance is a string metric for measuring the difference between two sequences. Informally, the Levenshtein distance between two words is the minimum number of single-character edits (insertions, deletions or substitutions) required to change one word into the other.

## Longest Common Substring Distance

The LCS work as finding longest identical substring from the compared string, Ex: The longest common substring of the strings "ABABC", "BABCA" and "ABCBA" is string "ABC" of length 3.

# PageRank

|  |  |
| --- | --- |
| 100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149 | 150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199 |

PageRank(PR) is an algorithm used by Google Search to rank websites in their search engine results. PR work by turning weight of edges of a node into the score of the node, the scores are calculated by the equation below:

Where is a set of pages that links to , is the number of outbound link on , and is total number of pages, and is the damping factor.

PageRank consider undirected edge as 2 edge with different direction, but the same weight.

# Evaluation

To evaluate the result of summarization, it using ROUGE method.

## Rouge

ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. It is essentially of a set of metrics for evaluating automatic summarization of texts as well as machine translation. It works by comparing an automatically produced summary or translation against a set of reference summaries (typically human-produced). Rouge will bring the output as a score between 0 until 1. The larger score result, the better result of summary. The score are calculated by the equation below:

The equation above can be used to calculate either recall or precision.

The table below show the result of Rouge method of comparing 3 sentences summary with gold standard.

|  |  |  |
| --- | --- | --- |
| File Name | Levenshtein | LCS |
| Test 01 | 0.6609 | 0.6609 |
| Test 02 | 0.4897 | 0.4897 |
| Test 03 | 0.7890 | 0.7890 |
| Test 04 | 0.7245 | 0.7890 |
| Test 05 | 0.6516 | 0.6516 |

And the table below compring 1 senteces summary with gold standard.

|  |  |  |
| --- | --- | --- |
| File Name | Levenshtein | LCS |
| Test 01 | 0.2446 | 0.2446 |
| Test 02 | 0.1237 | 0.1237 |
| Test 03 | 0.4771 | 0.4771 |
| Test 04 | 0.4286 | 0.4286 |
| Test 05 | 0.2787 | 0.2787 |

# Conclusion

As shown in the evaluation, the program score better more sentences is used in the summary. The use of levenshtein distance or LCS distance show no difference in the resulting summary.

References

Davidadamojr. 2018. *TextRank*. https://github.com/davidadamojr/TextRank.

Icoxfog417. 2018. *awesome-text-summarization*. https://github.com/davidadamojr/TextRank.

Metalaman. 2017. *Scripts-for-extractive-summarization*. https://github.com/davidadamojr/TextRank.

# Job Description

|  |  |
| --- | --- |
| Name | Work |
| Jorgie B. | TextRank, Dataset, Report |
| Puruso M H. | Evaluation, Debugging, Report |