An Alternative Approach to TextRank in Keyword Extraction

Student: Phuc Nguyen (phuc.nguyen@marquette.edu)
Mentor: Dr. Thomas Kaczmarek (thomas.kaczmarek@marquette.edu)
Marquette University – Mathematics, Statistic and Computer Science Department

INTRODUCTION

Text mining uses computational techniques to discover unknown information from text documents. Within text mining, automatic **keyword extraction** retrieves the most relevant terms and phrases. Despite being commonly used in search engines, manual keywords creation is onerous amid the massive amount of information available as texts.[1]

This project looks at TextRank – a graph algorithm that calculates the weighted score for each potential keyword. Specifically, we give an alternative approach that replaces the recursive formula by integrating ideas from the PageRank algorithm used by Google in web search.

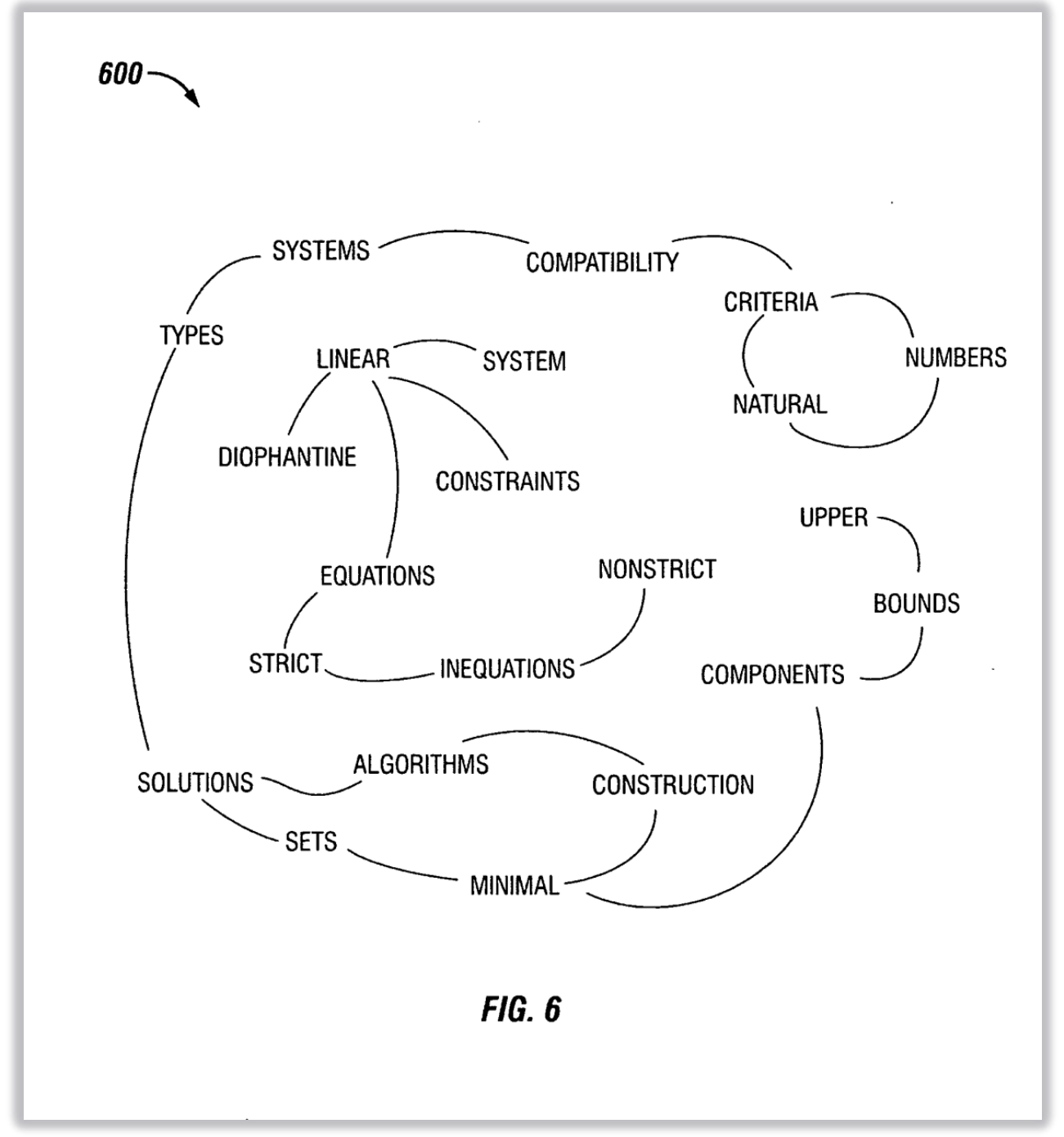

Figure 1: A sample graph from TextRank[2]

BACKGROUND

Definition: A directed graph *G = (V, E)* where the set of vertices *V* contains **potential keywords** of a document and the set of edges *E* denotes the relationship between vertices.

TextRank calculates the weighted score $S(V_i)$ for each vertex V_i in *G* by using the recursive formula:[2]

$$\boldsymbol{S(V_i) = (1 - d) + d * \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j)}$$

d: damping factor between 0 and 1

$In(V_i)$: set of vertices that point to V_i

$Out(V_i)$: set of vertices that V_i points to

OBJECTIVES

- Develop an alternative approach to perform TextRank
- Utilize tools from linear algebra and numerical analysis to calculate runtime complexity of the new approach

DEFINITIONS AND THEOREMS

Concepts: positive matrix, column (row) stochastic matrix, eigenvalue and eigenvector, weighted score vector, complexity theory, Gaussian elimination, theory of Markov chain, probability theory, graph theory, mathematical analysis

$$A = \begin{bmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix}, \quad \text{and} \quad A^t = \begin{bmatrix} 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 1 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{bmatrix}$$

Figure 2: Column and row stochastic matrix[3]

Perron-Frobenius Theorem (special case): If *A* is a positive column stochastic matrix, then:

- 1 is an eigenvalue of *A* with multiplicity 1. All other eigenvalues are less than 1 in magnitude
- There exists an eigenvector *v* of *A* corresponding to the eigenvalue 1 such that all entries in *v* are positive and that the sum of all entries equal 1. In addition, *v* is unique up to a scalar. The vector *v* is also called the **probabilistic eigenvector** corresponding to the eigenvalue 1

Power Method Convergence Theorem: Let *A* be a positive *n* by *n* column stochastic matrix with the probabilistic eigenvector v^* and $z \in R^n$ be the weighted score vector such that each entry of *z* equals *1/n*, then the sequence $Az, A^2z, A^3z, \ldots$ converges to v^*.

The rate of convergence is the magnitude of the second largest eigenvalue of *A*.[4, 5]

METHODS

- Create an *n* by *n* column stochastic matrix *A* where
  - $\boldsymbol{A_{ji} = \frac{1}{|Out(V_i)|}}$ if there is an edge between $\boldsymbol{V_i}$ and $\boldsymbol{V_j}$
  - $\boldsymbol{A_{ji} = 0}$ otherwise
- Create the *n* by *n* positive transition matrix

$$B = (1 - d)A + dP$$

where *d* is a damping factor between 0 and 1 and

$$P = \frac{1}{n}\begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}$$

- Apply the Perron-Frobenius Theorem and the Power Method Convergence Theorem to compute the final weighted score of each vertex of *B*

RESULTS

For abstracts with 100 potential keywords, this approach requires on average 1.416 milliseconds to converge compared to 2.004 milliseconds for the TextRank algorithm using the same convergence criteria.[6] Therefore, the new approach produces comparable results with potentially improved runtime efficiency.

REFERENCES

1) Hasan, K., and Vincent Ng. "Automatic Keyphrase Extraction: A Survey of the State of the Art." (2014) Print.

2) Mihalcea, Rada, and Paul Tarau. "TextRank: Bringing Order into Texts." (2004) Print.

3) Tanase, Raluca, and Remus Radu. "The Mathematics of Web Search." 2009. Web.

4) Langville, A., and C. Meyer. "Deeper Inside PageRank." (2004) Print.

5) Berkhin, Pavel. "A Survey on PageRank Computing." (2005) Print.

6) Rose, Stuart, et al. "Automatic Keyword Extraction from Individual Documents." (2010) Print.

ACKNOWLEDGEMENT

We would like to thank Dr. Thomas Kaczmarek for his helpful suggestions and thorough guidance throughout the research process. We also would like to thank Dr. Dennis Brylow and Dr. Kim Factor who organize the REU program at Marquette University. Finally, we would like to thank the Wehr Foundation and the MSCS Department for their financial support.