TextRank Algorithm

Pham Quang Nhat Minh

FPT Technology Research Institute (FTRI)
minhpqn@fpt.edu.vn

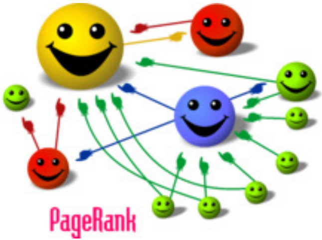
November 22, 2016

Table of contents

Table of Contents

Introduction

- In many problems, data points can be modeled as vertexes of a graph. Related data points are connected.
  - Users in social media, computers in a network, web pages, sentences in a document, etc
- We may want to identify **important vertexes** in the graph
  - E.g., "Hot" facebookers, a computer that gets many accesses.
  - The relative importance of a vertex in a graph depends on the graph structure.

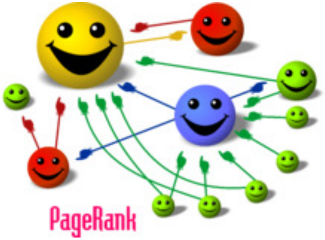


Graph-based algorithms

- Decide the importance of a vertex within a graph
  - Taking into account global information
    - Recursively computed from the entire graph
- Applications
  - Citation analysis
  - Social networks
  - Link-structure of the WWW
- In NLP
  - Keyphrase extraction
  - Extractive summarization
  - ...

Table of Contents

- Graph-based algorithms
  - A way of deciding the importance of a vertex within a graph
  - Based on global information
    - Recursively drawn from the entire graph
- Basic idea
  - Voting (Recommendation)
- The score of vertex
  - How many votes it gets?
  - Who votes for it?



The TextRank Model

- Score of a vertex

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

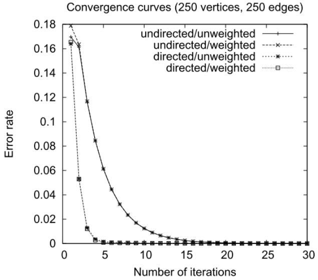
  - $S(V_i)$: Score of the vertex
  - V_i: Vertex
  - $In(V_i)$: the set of vertices that point to it (predecessors)
  - $Out(V_i)$: the set of vertices that the vertex points to (successors)
  - d: the damping factor, that is the probability of jumping from a given vertex to another vertex
    - Random surfer model
    - In PageRank $d = 0.85$

The TextRank Model

- Starting from arbitrary values assigned to each node in the graph
- The computation iterates
  - Until convergence below a given threshold is achieved
- Scores of vertices obtained after running the algorithm
  - Represent the tmportance of the vertex within the graph
  - Not affected by the choice the initial value
    - Only the number of iterations to convergence may be different

The TextRank Model: Undirected Graphs

- Recursive graph-based ranking algorithm
  - Traditionally applied on directed graphs
  - Can be applied to undirected graphs
    - The out-degree of a vertex is equal to the in-degree of the vertex.
- Convergence curve
  - As the connectivity of the graph increases
    - Fewer iterations
    - The convergence curve for directed and undirected graphs practically overlap



The TextRank Model: Weighted Graphs

- PageRank
  - Assuming unweighted graph
  - Page hardly include multiple or partial links to another page
- TextRank
  - May include multiple or partial link between the units
    - The graphs are built from natural language text
  - Incorporate the "strength" of connectivity
    - Weight of the edge

The TextRank Model: Weighted Graphs

- New measure

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

  - The final score differ significantly as compared to original measure
  - The number of iterations is almost identical
    - for weighted and unweighted graphs

The TextRank Model

- Build a graph
  - Represent the text
  - Interconnect words or other text entities with meaningful relations
    - Text unit of various size
    - Various characteristics: words, entire sentences, collocations, etc
  - The type of relations
    - Lexical semantic relations
    - Contextual overlap
    - Etc

The TextRank Model: Text as a Graph

4 steps of Graph-based ranking algorithms

- Identify text units
    - Best define the task at hand
    - Add them as vertices in the graph
- Identify relations
    - Connect such text units
    - Use these relations to draw edges
        - Directed
        - Undirected
- Iterate the graph-based ranking algorithm
    - Until converge
- Sort vertices based on their final score

Table of Contents

Keyword Extraction

- Automatically identify a set of terms
  - Best describe the document
- Use of extracted keywords
  - Building an automatic index
  - Classify a text
  - Concise summary
  - Terminology extraction
  - Construction of domain-specific dictionaries

Keyword Extraction: Possible approaches

- Frequency criterion
- Supervised learning methods
  - Parametrized heuristic (combined with a genetic algorithm)
    - Turney, 1999
    - Precision: 29.0% (five key phrases per document)
  - Naive Bayes
    - Frank et al., 1999
    - Precision: 18.3% (fifteen key phrases per document)

TextRank for Keyword Extraction

- Input: A document
- Output:
  - A set of words or phrases
    - Representative for the document
- Relation
  - Can be defined between two lexical units
  - Co-occurence relation
    - Two vertices are connected if their corresponding lexical units co-occur within a window of maximum N words.
    - N can be set values from 2 to 10 words.
- Syntactic filter
  - All open-class words
  - Nouns and verbs
  - Nouns and adjectives only

TextRank for Keyword Extraction: TextRank Process

- Text tokenization
  - Annotated with parts of speech
  - Preprocessing step required to enable the application of syntactic filters
  - Only single words as candidates for addition the the graph
    - To avoid excessive growth of the graph size
    - Multi-word keywords being eventually reconstructed in the post-processing phrase.
- Syntactic filtering
  - All lexical units that pass the filter are added to the graph
  - Edge is added between those lexical units that co-occur within a window of N words.
  - Initial score of each vertex is set to 1

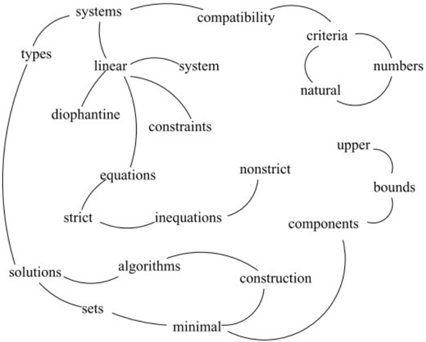
TextRank for Keyword Extraction

- Ranking algorithm
  - Is run the graph for several iterations
    - Until converges (usually 20 $\approx$ 30 iterations)
    - Threshold of 0.0001
- Sorting
  - Reverse order of their score
  - The top of T vertices are retained for post-processing
    - T may be set to any fixed value (usually ranging from 5 to 20)
    - By decides the number of keywords based on the size of the text
    - T is set to a third of the number of vertices in the graph
- Post-processing
  - Sequences of adjacent keywords are collapsed into a multi-word keyword
    - E.g) *Matlab code for plotting ambiguity functions*
    - If *Matlab* and *code* are selected as potential keywords
    - They are collapsed into single keyword *Matlab code*

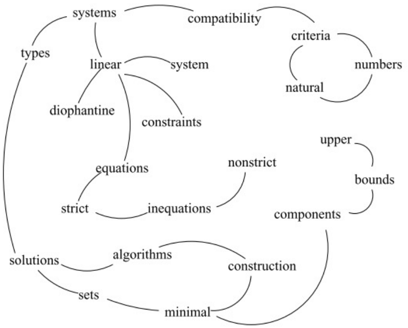
TextRank for Keyword Extraction

Sample graph

Compatibility of systems of linear constraints over the set of natural numbers. Criteria of compatibility of a system of linear Diophantine equations, strict inequations, and nonstrict inequations are considered. Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types systems and systems of mixed types.



Output of Keyword Extraction



Keywords assigned by TextRank:
linear constraints; linear diophantine equations; natural numbers; nonstrict inequations; strict inequations; upper bounds

Keywords assigned by human annotators:
linear constraints; linear diophantine equations; minimal generating sets; non–strict inequations; set of natural numbers; strict inequations; upper bounds

TextRank Implementations

- (iPython notebook) A Study of the TextRank Algorithm in Python: `http://tinyurl.com/h9tzytk`
  - Implement the key phrase extraction part of it using the **networkx** and **NLTK** packages, **matplotlib** to visualize graphs.
- (Python) davidadamojr/TextRank: `http://tinyurl.com/jy8evtl`

References

- Mihalcea, Rada, and Paul Tarau. "TextRank: Bringing order into texts." Association for Computational Linguistics, 2004.
- Slide: "TextRank: Bringing order into texts." `http://tinyurl.com/hjbt852`