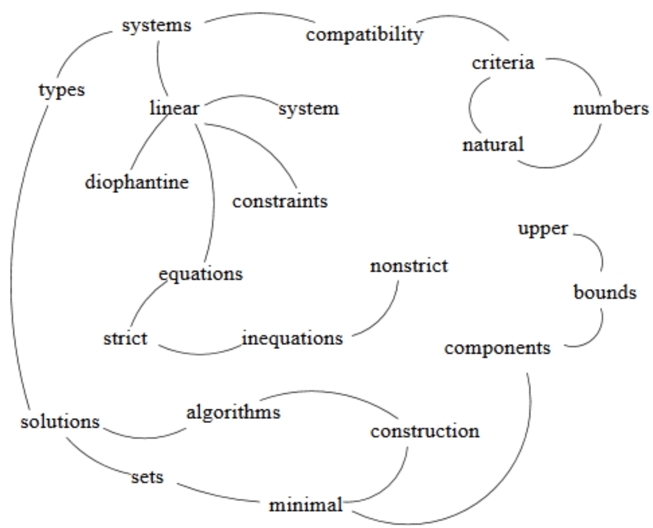
TextRank



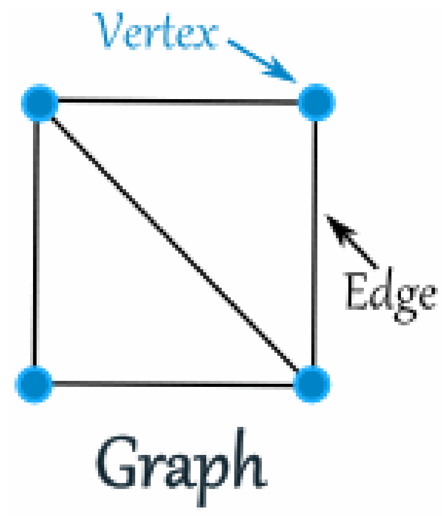
Chuck Chan

March 29, 2017

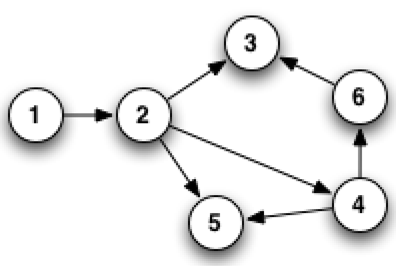
TextRank: Bringing Order into Texts

- Published 2004
- Rada Mihalcea and Paul Tarau
- Paper source: https://web.eecs.umich.edu/~mihalcea/papers/mihalcea.emnlp04.pdf
- Introduce TextRank algorithm
- Evaluate:
  - Unsupervised keyword extraction
    - Data Source: 500 abstracts from Inspec database
    - Evaluation: F-Score, precision, recall
  - Unsupervised sentence extraction
    - 567 news articles provided during Document Understanding Evaluation 2002
    - Evaluation: ROUGE Evaluation Tookit

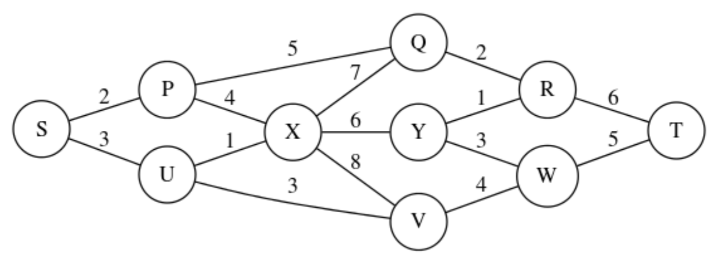
Quick introduction to Graphs terms



Directional graph



Weighted graph



Graph-based ranking algorithms

- Unsupervised
- Semantic graphs extracted from documents
  - Can be words or sentences
- Recursively ranks units within graphs
- Examples: PageRank, Kleinberg's HITS, Positional Function

Basic steps for TextRank

1. *Identify text units that best define the task at hand, and add them as vertices in the graph.*
2. *Identify relations that connect such text units, and use these relations to draw edges between vertices in the graph. Edges can be directed or undirected, weighted or unweighted.*
3. *Iterate the graph-based ranking algorithm until convergence.*
4. *Sort vertices based on their final score. Use the values attached to each vertex for ranking/selection decisions.*

Scoring of vertices

- Based on ranking model
  - When vertex link it cast a vote for other vertex
  - Higher the votes the more important

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

Score of vertex Vi

Damping factor set between 0-1 probability of jumping from a given vertex to another random vertex in the graph. Set at 0.85

Set of vertices that points to vertex Vi points to

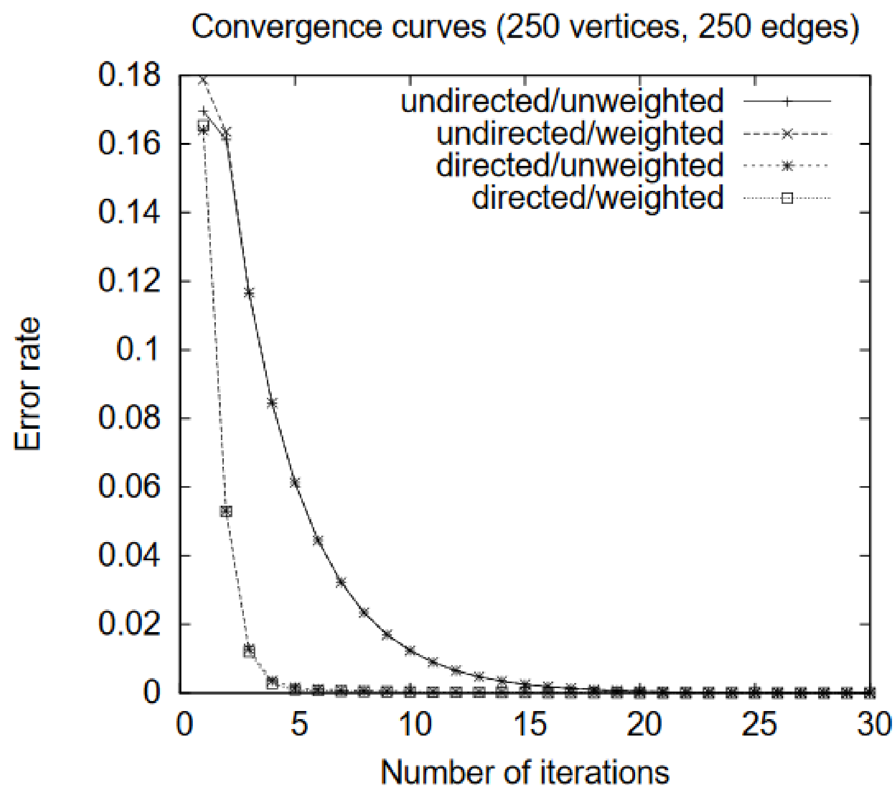
Set of vertices that vertex Vi points to (successors)

Recursive Computation

- Decide importance of a vertex in a graph by recursively computing global information from the entire graph
  - Arbitrary values assigned to each node
  - Iterates scoring computation until convergence threshold is achieved
- Score from each run represents importance

Graph Types & Convergence

- Unidirectional graphs
  - Out-degree vertices = in-degree vertices
  - More connectivity, fewer iterations for convergence
- Weighted graphs
  - Weights included between vertices (links between text)



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

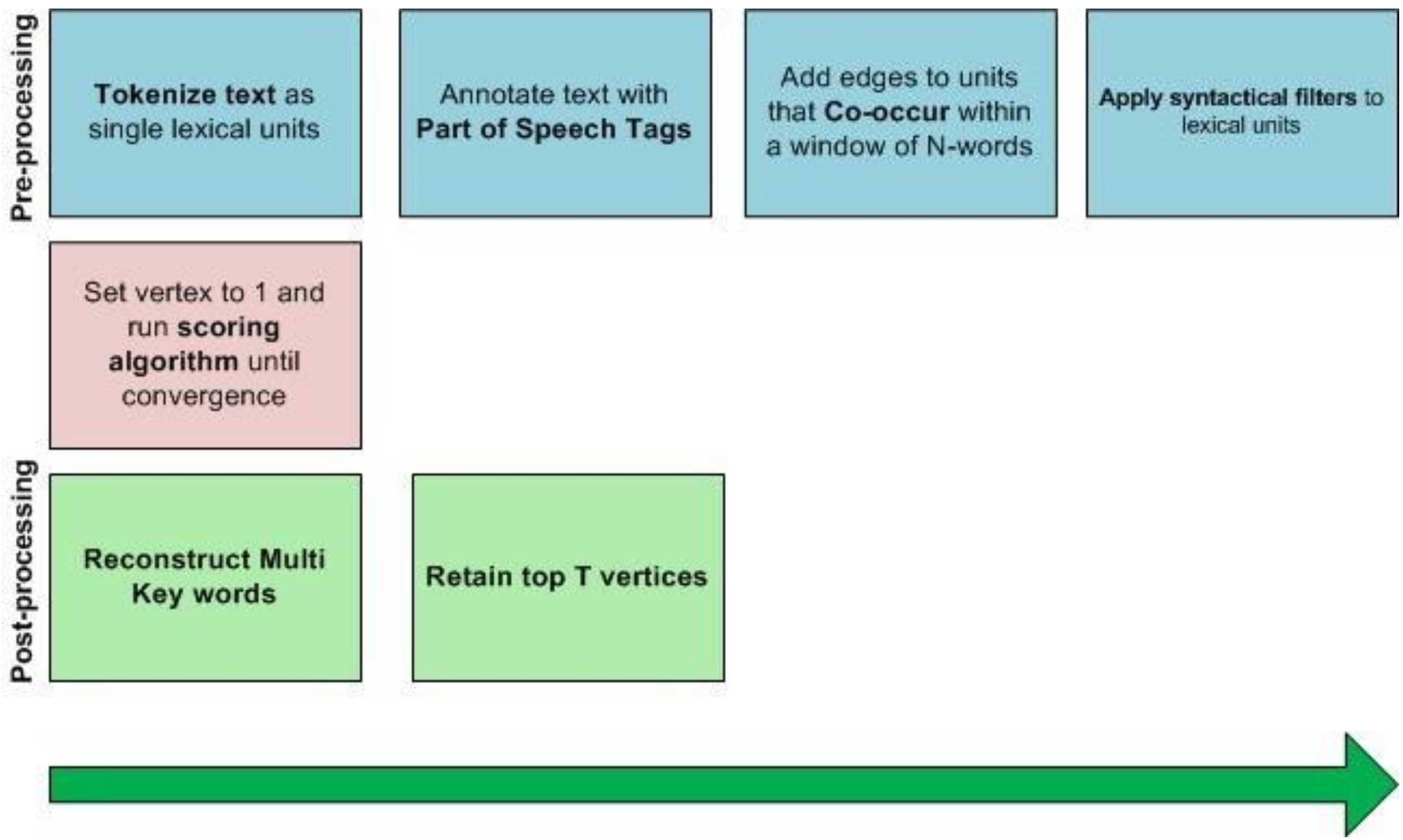
TextRank for Key Word Extraction

- Select of keyphrases representative for a given text
- Uses
  - Classifying text
  - Creating an automatic index for a collection
  - Concise summary for a document

TextRank for Key Word Extraction

- Key words extracted from 500 abstracts (dataset)
- Units ranked: sequences of one or more lexical units extracted from text
- Co-occurrence relation represented by a window of maximum words and is represented by an edge between vertices
- Restrictions with POS syntactical filters: all class words, nouns and verbs only, and nouns + adjectives (best result). This limits the number of edges in the graph

Key Word Extraction steps



Keyword Extraction Evaluation

Results compared against professional indexers and with Hulth (2003) supervise learning scheme that extracts keywords by:

1. Within documents frequency
2. Collection frequency
3. Relative position of first occurrence
4. Sequence of Part of Speech tags

Also use N-grams, NP -chunking, and POS patterns

Larger co-occurence window does not help, showing relationship is not strong enough to define a connection

No natural directions established between co-occurring words

Low recall possibly due to number of keywords selected

Method	Assigned Total	Assigned Mean	Correct Total	Correct Mean	Precision	Recall	F-measure
TextRank							
Undirected, Co-occ.window=2	6,784	13.7	2,116	4.2	**31.2**	43.1	**36.2**
Undirected, Co-occ.window=3	6,715	13.4	1,897	3.8	28.2	38.6	32.6
Undirected, Co-occ.window=5	6,558	13.1	1,851	3.7	28.2	37.7	32.2
Undirected, Co-occ.window=10	6,570	13.1	1,846	3.7	28.1	37.6	32.2
Directed, forward, Co-occ.window=2	6,662	13.3	2,081	4.1	31.2	42.3	35.9
Directed, backward, Co-occ.window=2	6,636	13.3	2,082	4.1	31.2	42.3	35.9
Hulth (2003)							
Ngram with tag	7,815	15.6	1,973	3.9	25.2	**51.7**	33.9
NP-chunks with tag	4,788	9.6	1,421	2.8	29.7	37.2	33.0
Pattern with tag	7,012	14.0	1,523	3.1	21.7	39.9	28.1

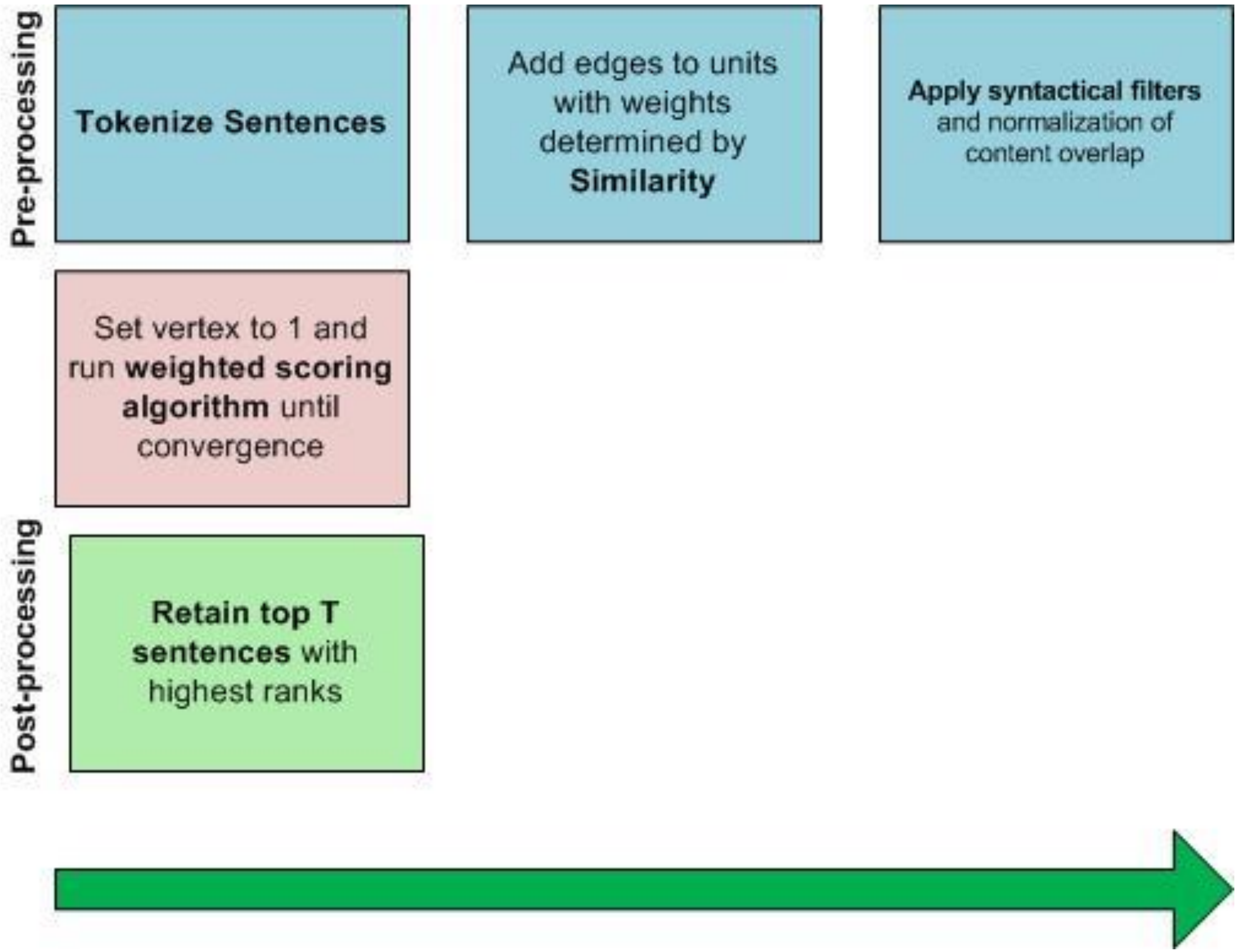
TextRank for Sentence Extraction

- Used for
  - Automatic summarization
- Each sentence represents a vertex
- Connections are made based on similarity relationships between sentences as a function of content overlap
- The relationships are used for ranking
- Syntactical filters limits number of edges by count only words of specific POS categories
- Sentences are normalized by dividing content overlap by sentence length

Metrics for similarity

- $Similarity(S_i, S_j) = \frac{|\{w_k | w_k \in S_i \& w_k \in S_j\}|}{\log(|S_i|) + \log(|S_j|)}$
  - S is sentence
  - w is each word
- Other metrics: String kernels, Cosine similarity, Longest common subsequence
- Similarity metrics are used as weights in the equation for scoring vertices for the ranking algorithm

Sentence Extraction Steps



Sentence Extraction Evaluation

- 567 news articles provided during Document Understanding Evaluation (DUC) 2002
- Use ROUGE evaluation toolkit, based on Ngrams
  - Basic, stemmed, stemmed no-stopwords
  - Lower score is better
- Compared with 15 different systems and DUC baseline

System	ROUGE score – Ngram(1,1)		
	basic (a)	stemmed (b)	stemmed no-stopwords (c)
S27	0.4814	0.5011	0.4405
S31	0.4715	0.4914	0.4160
TextRank	**0.4708**	**0.4904**	**0.4229**
S28	0.4703	0.4890	0.4346
S21	0.4683	0.4869	0.4222
Baseline	*0.4599*	*0.4779*	*0.4162*
S29	0.4502	0.4681	0.4019

Key Points

- Unsupervised, no training needed
- Can be used on short or long summaries since it only ranks
- Does not rely on local context , accounts for global information recursively from built graph

Libraries

- Python
  - Gensim summarizer
  - https://radimrehurek.com/gensim/summarization/summariser.html
- Java
  - From Paco Nathan ceteri
  - https://github.com/ceteri/textrank
- R
  - Package 'Rtextrankr‘
  - https://github.com/mikigom/Rtextrankr
- Javascript
  - TextRank for Node.js
  - https://github.com/nadr0/TextRank-node