A Language Independent Algorithm for Single and Multiple Document Summarization

Rada Mihalcea, Paul Tarau

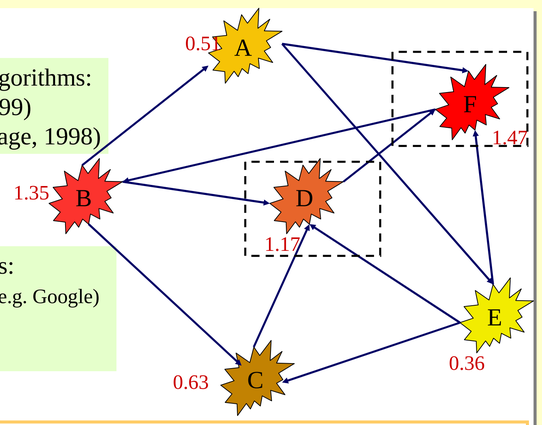
Department of Computer Science, University of North Texas, {rada,tarau}@cs.unt.edu

Text as a Graph

Background

Graph-based ranking algorithms:
- HITS (Kleinberg, 1999)
- PageRank (Brin & Page, 1998)

Traditional applications:
- Web-link analysis (e.g. Google)
- Social networks
- Citation analysis



Ranking algorithm	Vertex B links to vertex A ⇔ vertex B "votes" for vertex A Iterative voting ⇒ Ranking over all vertices

The Idea

Text as a graph
- lexical or semantic networks
  - semantic relations between concepts
  - definitional links among words
- graph models of text meaning
  - word senses connected by semantic relations
- graph models of text cohesion
  - text units (e.g. sentences) connected by their similarity

Graph-based ranking algorithms on text graphs
- ranking of word senses to identify the correct sense
- ranking of words in a text to pinpoint the important keywords
- ranking of sentences in a document to identify the most important ones

Main Steps

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 them to draw **edges** 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.

Mathematical Model

Ranking Algorithm

Terminology: G = (V,E) a directed graph with vertices V and edges E
$In(V_i)$ = predecessors of V_i
$Out(V_i)$ = successors of V_i

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

Assign a random initial value to each vertex in the graph
Iterate the scoring function until convergence (on text: 25-30 iterations)
Score based on PageRank (Brin and Page, 1998)
d – damping factor $\in [0,1]$ (usually 0.85)
– indicates the probability to jump to a random page

Undirected Graphs

Ranking algorithms are traditionally applied on directed graphs
Can be also applied to undirected graph ⇒ more gradual convergence

Weighted Graphs

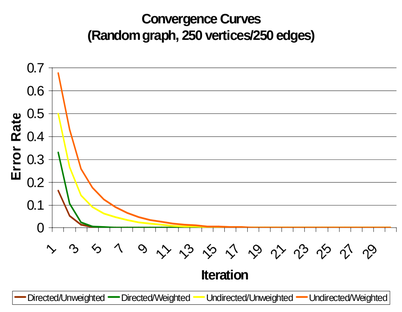
Weights can model the *strength* of the relations between textual units
Original definition of ranking algorithms assumes unweighted graphs
We introduce new ranking formula to take into account edge weights

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

Graph Structure

- *Undirected*
  a sentence can recommend any other sentence in the text
- *Directed forward*
  a sentence can recommend only sentences that follow in the text (movie reviews)
- *Directed backward*
  a sentence can recommend only sentences that precede it in the text (news articles)

Convergence



Single Document Summarization

The Problem

- Identify sentences that are "important" for the understanding of a given text
- Useful (needed?) for text summarization

Previous work

- DUC evaluations http://www-nlpir.nist.gov/projects/duc/
- E.g.: Supervised learning (Teufel 97), Unsupervised extraction (Salton97)

TextRank – fully unsupervised

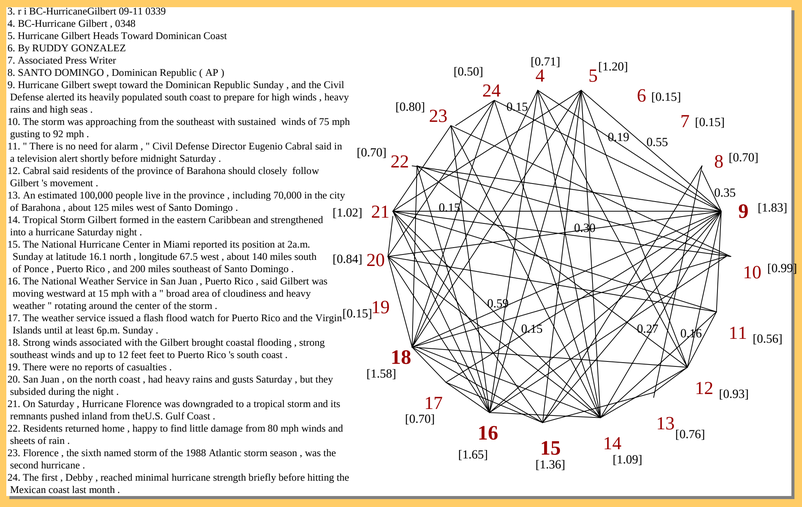
1. Build graph
   Vertices = sentences in the text
   Edges = similarity relation ⇒ weights

$$Sim(S_1, S_2) = \frac{|\{w_k \mid w_k \in S_i \wedge w_k \in S_j\}|}{\log(|S_i|) + \log(|S_j|)}$$

   other similarity metrics: cosine, string kernels, etc.
2. Ranking
   Run weighted ranking algorithm and keep top N ranked sentences

Table 1: Results for multi-document summarization (U = Undirected; DB = Directed Backward)

Table 2: Results for top 5 DUC 2002 multi-document summarization systems, and baseline.



Evaluation

English

- 567 news articles from DUC 2002 – create 100-word summaries
- Automatic evaluation with ROUGE (Lin & Hovy) – Ngram(1,1)
- 15 systems from DUC 2002 (table - top 5)
- Baseline = top sentences in each document

		Graph	
Algorithm	Undirected	Forward	Backward
$HITS_A^W$	49.12	45.84	**50.23**
$HITS_H^W$	49.12	50.23	45.84
PR_W	49.04	42.02	**50.08**

Top 5 systems (DUC 2002)					
S27	S31	S28	S21	S29	Baseline
50.11	49.14	48.9	48.69	46.81	47.99

Portuguese

- 100 news articles in the TeMario data set (Pardo & Rino, 2003)
  - 40 documents from Jornal de Brasil
  - 60 documents from Folha de Sao Paulo
- Summaries consisting of 25-30% of the original document

		Graph	
Algorithm	Undirected	Forward	Backward
$HITS_A^W$	48.14	48.34	50.02
$HITS_H^W$	48.14	50.02	48.34
PR_W	49.39	45.74	**51.21**

Baseline: 49.63

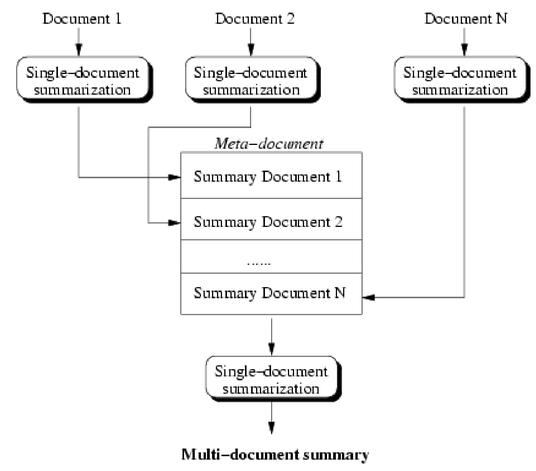
Multi-Document Summarization

The Problem

- Summarize all documents in a cluster
- Cluster identified manually / automatically

TextRank – fully unsupervised

1. Multi-document summaries are built using a ``meta" summarization procedure.
2. First, for each document in a given cluster of documents, a single document summary is generated using one of the graph-based ranking algorithms.
3. Next, a ``summary of summaries" is produced using the same or a different ranking algorithm.



Evaluation

- 567 news articles from DUC 2002 – grouped into 59 clusters
- Create 100-word summaries
- Automatic evaluation with ROUGE (Lin & Hovy) – Ngram(1,1)
- 10 systems from DUC 2002 (table lists top 5)
- Baseline = top sentence in each document

Single doc	"Meta" summarization algorithm			
summarization	PR^W-U	PR^W-DB	$HITS^W_A$-U	$HITS^W_A$-DB
PageRank^W-U	**35.52**	34.99	34.56	34.65
PageRank^W-DB	35.02	34.48	35.19	34.39
$HITS^W_A$-U	33.68	32.59	32.12	34.23
$HITS^W_A$-DB	**35.72**	35.20	34.62	34.73

S26	S19	S29	S25	S20	Baseline
35.78	34.47	32.64	30.56	30.47	29.32

Why TextRank Works

A "Recommendation" Process

- A text unit "recommends" another text unit
- Strength of recommendation recursively computed
- Preference given to recommendations made by the most "influential" units

- A sentence that addresses a certain concept gives the reader a recommendation to refer to other sentences in the text that address the same concept
- Highly recommended sentences are likely to be more important

- A similar process can be applied to other problems:
  - keyword extraction
  - document reranking
  - concept extraction

Text Surfing

PageRank: "random surfer model" – a user surfs the Web by following links from any given Web page

TextRank: "text surfing" – from a given concept C we are likely to follow links to related/connected concepts
– text cohesion (Halliday & Hasan 1979)
– text knitting (Hobbs 1974): facts associated with words are shared in different parts of the discourse; such relations serve to "knit the discourse together"

Cohesive text = "Web" of connections – approximates human memory models

All the pros …

- Unsupervised – information exclusively drawn from the text itself
- Goes beyond sentence connectivity (see sentence 15)
- Gives a ranking over all sentences in the text – can be adapted to longer/shorter summaries
- No training data required – can be adapted to other languages
- Can be used for both single and multiple document summarization