Graph-based Algorithms in NLP
Text as a Graph

• Vertices = cognitive units

• Edges = relations between cognitive units
Text as a Graph

- Vertices = cognitive units

  - words
  - Word sense
  - sentences

- Edges = relations between cognitive units

  - Semantic relations
  - Co-occurrence
  - similarity

  Word Sense Disambiguation

  Keyword Extraction

  Sentence Extraction

TextRank (Mihalcea and Tarau, 2004), LexRank (Erkan and Radev, 2004)
TextRank - Weighted Graph

• Edges have weights – similarity measures
• Adapt PageRank, HITS to account for edge weights
• PageRank adapted to weighted graphs

\[ WS(V_i) = (1 - d) + d \sum_{j \in \text{In}(V_i)} \frac{W_{ji}}{\sum_{V_k \in \text{Out}(V_j)} W_{jk}} WS(V_j) \]
TextRank - Text Summarization

Build the graph:
- Sentences in a text = vertices
- Similarity between sentences = weighted edges

Model the cohesion of text using intersentential similarity

2. Run link analysis algorithm(s):
- keep top N ranked sentences
- → sentences most “recommended” by other sentences
Underlining idea: A Process of Recommendation

• A sentence that addresses certain concepts in a text gives the reader a recommendation to refer to other sentences in the text that address the same concepts

• Text knitting (Hobbs 1974)
  - repetition in text “knits the discourse together”

• Text cohesion (Halliday & Hasan 1979)
Graph Structure

• Undirected
  – No direction established between sentences in the text
  – A sentence can “recommend” sentences that precede or follow in the text

• Directed forward
  – A sentence “recommends” only sentences that follow in the text
  – Seems more appropriate for movie reviews, stories, etc.

• Directed backward
  – A sentence “recommends” only sentences that proceed in the text
  – More appropriate for news articles
Sentence Similarity

• Inter-sentential relationships
  – weighted edges
• Count number of common concepts
• Normalize with the length of the sentence

\[
Sim(S_1, S_2) = \frac{\left| \left\{ w_k \mid w_k \in S_1 \land w_k \in S_2 \right\} \right|}{\log(|S_1|) + \log(|S_2|)}
\]

• Other similarity metrics are also possible:
  – Longest common subsequence
  – string kernels, etc.
Hurricane Gilbert swept toward the Dominican Republic Sunday, and the Civil Defense alerted its heavily populated south coast to prepare for high winds, heavy rains and high seas.

The storm was approaching from the southeast with sustained winds of 75 mph gusting to 92 mph.

"There is no need for alarm," Civil Defense Director Eugenio Cabral said in a television alert shortly before midnight Saturday.

Cabral said residents of the province of Barahona should closely follow Gilbert's movement.

An estimated 100,000 people live in the province, including 70,000 in the city of Barahona, about 125 miles west of Santo Domingo.

Tropical Storm Gilbert formed in the eastern Caribbean and strengthened into a hurricane Saturday night.

The National Hurricane Center in Miami reported its position at 2 a.m. Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo.

The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westward at 15 mph with a "broad area of cloudiness and heavy weather" rotating around the center of the storm.

The weather service issued a flash flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday.

Strong winds associated with the Gilbert brought coastal flooding, strong southeast winds and up to 12 feet to Puerto Rico's south coast.

There were no reports of casualties.

San Juan, on the north coast, had heavy rains and gusts Saturday, but they subsided during the night.

On Saturday, Hurricane Florence was downgraded to a tropical storm and its remnants pushed inland from the U.S. Gulf Coast.

Residents returned home, happy to find little damage from 80 mph winds and sheets of rain.

Florence, the sixth named storm of the 1988 Atlantic storm season, was the second hurricane.

The first, Debby, reached minimal hurricane strength briefly before hitting the Mexican coast last month.
Hurricane Gilbert swept toward the Dominican Republic Sunday, and the Civil Defense alerted its heavily populated south coast to prepare for high winds, heavy rains and high seas. The National Hurricane Center in Miami reported its position at 2 a.m. Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo. The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westward at 15 mph with a "broad area of cloudiness and heavy weather" rotating around the center of the storm. Strong winds associated with the Gilbert brought coastal flooding, strong southeast winds and up to 12 feet to Puerto Rico's coast.

Reference summary I

Hurricane Gilbert swept toward the Dominican Republic Sunday with sustained winds of 75 mph gusting to 92 mph. Civil Defense Director Eugenio Cabral alerted the country's heavily populated south coast and cautioned that even though there is no need for alarm, residents should closely follow Gilbert's movements. The U.S. Weather Service issued a flash flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday. Gilbert brought coastal flooding to Puerto Rico's south coast on Saturday. There have been no reports of casualties. Meanwhile, Hurricane Florence, the second hurricane of this storm season, was downgraded to a tropical storm.

Reference summary II

Hurricane Gilbert is moving toward the Dominican Republic, where the residents of the south coast, especially the Barahona Province, have been alerted to prepare for heavy rains, and high winds and seas. Tropical Storm Gilbert formed in the eastern Caribbean and became a hurricane on Saturday night. By 2 a.m. Sunday it was about 200 miles southeast of Santo Domingo and moving westward at 15 mph with winds of 75 mph. Flooding is expected in Puerto Rico and the Virgin Islands. The second hurricane of the season, Florence, is now over the southern United States and downgraded to a tropical storm.
Evaluation

• Task-based evaluation: automatic text summarization
  – Single document summarization
    • 100-word summaries
  – Multiple document summarization
    • 100-word multi-doc summaries
    • clusters of ~10 documents

• Automatic evaluation with ROUGE (Lin & Hovy 2003)
  – n-gram based evaluations
    • unigrams found to have the highest correlations with human judgment
  – no stopwords, stemming
Evaluation

• Data from DUC (Document Understanding Conference)
  – DUC 2002
  – 567 single documents
  – 59 clusters of related documents

• Summarization of 100 articles in the TeMario data set
  – Brazilian Portuguese news articles
    • Jornal de Brasil, Folha de Sao Paulo
  – (Pardo and Rino 2003)
Evaluation

- Single-doc summaries for 567 documents (DUC 2002)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Graph Algorithm</th>
<th>Undirected</th>
<th>Dir. forward</th>
<th>Dir. backward</th>
</tr>
</thead>
<tbody>
<tr>
<td>PR&lt;sub&gt;W&lt;/sub&gt;</td>
<td></td>
<td>0.4904</td>
<td>0.4202</td>
<td><strong>0.5008</strong></td>
</tr>
<tr>
<td>HITS&lt;sub&gt;A&lt;/sub&gt;&lt;sup&gt;W&lt;/sup&gt;</td>
<td></td>
<td>0.4912</td>
<td>0.4584</td>
<td><strong>0.5023</strong></td>
</tr>
<tr>
<td>HITS&lt;sub&gt;H&lt;/sub&gt;&lt;sup&gt;W&lt;/sup&gt;</td>
<td></td>
<td>0.4912</td>
<td>0.5023</td>
<td>0.4584</td>
</tr>
</tbody>
</table>

Top 5 systems (DUC 2002)

<table>
<thead>
<tr>
<th></th>
<th>S27</th>
<th>S31</th>
<th>S28</th>
<th>S21</th>
<th>S29</th>
<th>Baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score</td>
<td>0.5011</td>
<td>0.4914</td>
<td>0.4890</td>
<td>0.4869</td>
<td>0.4681</td>
<td>0.4799</td>
</tr>
</tbody>
</table>
Evaluation

• Summarization of Portuguese articles
• Test the language independent aspect
  – No resources required other than the text itself
• Summarization of 100 articles in the TeMario data set

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Undirected</th>
<th>Dir.forward</th>
<th>Dir.backward</th>
</tr>
</thead>
<tbody>
<tr>
<td>HITS_A^w</td>
<td>0.4814</td>
<td>0.4834</td>
<td><strong>0.5002</strong></td>
</tr>
<tr>
<td>HITS_H^w</td>
<td>0.4814</td>
<td>0.5002</td>
<td>0.4834</td>
</tr>
<tr>
<td>PR_w</td>
<td>0.4939</td>
<td>0.4574</td>
<td><strong>0.5121</strong></td>
</tr>
</tbody>
</table>

• Baseline: 0.4963
Multiple Document Summarization

- Cascaded summarization ("meta" summarizer)
  - Use best single document summarization algorithms
    - PageRank (Undirected / Directed Backward)
    - HITS_A (Undirected / Directed Backward)
  - 100-word single document summaries
  - 100-word "summary of summaries"

- Avoid sentence redundancy:
  - set max threshold on sentence similarity (0.5)

- Evaluation:
  - build summaries for 59 clusters of ~10 documents
  - compare with top 5 performing systems at DUC 2002
  - baseline: first sentence in each document
## Evaluation

- Multi-doc summaries for 59 clusters (DUC 2002)

<table>
<thead>
<tr>
<th>Algorithm – Graph</th>
<th>PageRank-U</th>
<th>PageRank-DB</th>
<th>HITS $A$-U</th>
<th>HITS $A$-DB</th>
</tr>
</thead>
<tbody>
<tr>
<td>PageRank-U</td>
<td><strong>0.3552</strong></td>
<td>0.3499</td>
<td>0.3456</td>
<td>0.3465</td>
</tr>
<tr>
<td>PageRank-DB</td>
<td>0.3502</td>
<td>0.3448</td>
<td>0.3519</td>
<td>0.3439</td>
</tr>
<tr>
<td>HITS $A$-U</td>
<td>0.3368</td>
<td>0.3259</td>
<td>0.3212</td>
<td>0.3423</td>
</tr>
<tr>
<td>HITS $A$-DB</td>
<td><strong>0.3572</strong></td>
<td>0.3520</td>
<td>0.3462</td>
<td>0.3473</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Top 5 systems (DUC 2002)</th>
<th>S26</th>
<th>S19</th>
<th>S29</th>
<th>S25</th>
<th>S20</th>
<th>Baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.3578</td>
<td>0.3447</td>
<td>0.3264</td>
<td>0.3056</td>
<td>0.3047</td>
<td>0.2932</td>
</tr>
</tbody>
</table>
TextRank – Keyword Extraction

• Identify important words in a text
• Keywords useful for
  – Automatic indexing
  – Terminology extraction
  – Within other applications: Information Retrieval, Text Summarization, Word Sense Disambiguation
• Previous work
  – mostly supervised learning
  – genetic algorithms [Turney 1999], Naïve Bayes [Frank 1999], rule induction [Hulth 2003]
TextRank – Keyword Extraction

- Store words in vertices
- Use co-occurrence to draw edges
- Rank graph vertices across the entire text
- Pick top N as keywords

Variations:
- rank all open class words
- rank only nouns
- rank only nouns + adjectives
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 of systems and systems of mixed types.

Keywords by TextRank: linear constraints, linear diophantine equations, natural numbers, non-strict inequations, strict inequations, upper bounds

Keywords by human annotators: linear constraints, linear diophantine equations, non-strict inequations, set of natural numbers, strict inequations, upper bounds
Evaluation

• Evaluation:
  – 500 INSPEC abstracts
  – collection previously used in keyphrase extraction [Hulth 2003]
• Various settings. Here:
  – nouns and adjectives
  – select top N/3
• Previous work
  – [Hulth 2003]
  – training/development/test : 1000/500/500 abstracts

<table>
<thead>
<tr>
<th>Method</th>
<th>Assigned</th>
<th>Correct</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total</td>
<td>Mean</td>
</tr>
<tr>
<td>TextRank</td>
<td>6,784</td>
<td>13.7</td>
</tr>
<tr>
<td>Ngram with tag</td>
<td>7,815</td>
<td>15.6</td>
</tr>
<tr>
<td>NP-chunks with tag</td>
<td>4,788</td>
<td>9.6</td>
</tr>
<tr>
<td>Pattern with tag</td>
<td>7,012</td>
<td>14.0</td>
</tr>
</tbody>
</table>
TextRank on Semantic Networks

- Goal: build a semantic graph that represents the meaning of the text
- Input: Any open text
- Output: Graph of meanings (synsets)
  - “importance” scores attached to each synset
  - relations that connect them

- Models text cohesion
  - *(Halliday and Hasan 1979)*
  - From a given concept, follow “links” to semantically related concepts
- Graph-based ranking identifies the most recommended concepts
Two U.S. soldiers and an unknown number of civilian contractors are unaccounted for after a fuel convoy was attacked near the Baghdad International Airport today, a senior Pentagon official said. One U.S. soldier and an Iraqi driver were killed in the incident.
Main Steps

• **Step 1**: Preprocessing
  – SGML parsing, text tokenization, part of speech tagging, lemmatization

• **Step 2**: Assume any possible meaning of a word in a text is potentially correct
  – Insert all corresponding synsets into the graph

• **Step 3**: Draw connections (edges) between vertices

• **Step 4**: Apply the graph-based ranking algorithm
  – PageRank, HITS
Semantic Relations

- Main relations provided by WordNet
  - ISA (hyponym/hypernym)
  - PART-OF (meronym/holonym)
  - causality
  - attribute
  - nominalizations
  - domain links
- Derived relations
  - coord: synsets with common hypernym
- Edges (connections)
  - directed (direction?) / undirected
  - Best results with undirected graphs
- Output: Graph of concepts (synsets) identified in the text
  - “importance” scores attached to each synset
  - relations that connect them
Word Sense Disambiguation

• Rank the synsets/meanings attached to each word
• Unsupervised method for semantic ambiguity resolution of all words in unrestricted text (*Mihalcea et al. 2004*)
• Related algorithms:
  – Lesk
  – Baseline (most frequent sense / random)
• Hybrid:
  – Graph-based ranking + Lesk
  – Graph-based ranking + Most frequent sense
• Evaluation
  – “Informed” (with sense ordering)
  – “Uninformed” (no sense ordering)
• Data
  – Senseval-2 all words data (three texts, average size 600)
  – SemCor subset (five texts: law, sports, debates, education, entertainment)
## Evaluation

<table>
<thead>
<tr>
<th>Size(words)</th>
<th>Random</th>
<th>Lesk</th>
<th>TextRank</th>
<th>TextRank+Lesk</th>
</tr>
</thead>
<tbody>
<tr>
<td>SemCor</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>law</td>
<td>825</td>
<td>37.12%</td>
<td>39.62%</td>
<td>46.42%</td>
</tr>
<tr>
<td>sports</td>
<td>808</td>
<td>29.95%</td>
<td>33.00%</td>
<td>40.59%</td>
</tr>
<tr>
<td>education</td>
<td>898</td>
<td>37.63%</td>
<td>41.33%</td>
<td>46.88%</td>
</tr>
<tr>
<td>debates</td>
<td>799</td>
<td>40.17%</td>
<td>42.38%</td>
<td>47.80%</td>
</tr>
<tr>
<td>entertainment</td>
<td>802</td>
<td>39.27%</td>
<td>43.05%</td>
<td>43.89%</td>
</tr>
<tr>
<td>Avg.</td>
<td>826</td>
<td>36.82%</td>
<td>39.87%</td>
<td>45.11%</td>
</tr>
</tbody>
</table>

| Senseval-2   |        |      |          |              |
| d00          | 471    | 28.97% | 43.94% | 43.94% | 47.77% |
| d01          | 784    | 45.47% | 52.65% | 54.46% | 57.39% |
| d02          | 514    | 39.24% | 49.61% | 54.28% | 56.42% |
| Avg.         | 590    | 37.89% | 48.73% | 50.89% | 53.86% |
| Average (all)| 740    | 37.22% | 43.19% | 47.27% | 51.16% |

“uninformed” (no sense order)
## Evaluation

<table>
<thead>
<tr>
<th>Size(words)</th>
<th>MFS</th>
<th>Lesk</th>
<th>TextRank</th>
<th>TextRank+Lesk</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SemCor</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>law</td>
<td>825</td>
<td>69.09%</td>
<td>72.65%</td>
<td>73.21%</td>
</tr>
<tr>
<td>sports</td>
<td>808</td>
<td>57.30%</td>
<td>64.21%</td>
<td>68.31%</td>
</tr>
<tr>
<td>education</td>
<td>898</td>
<td>64.03%</td>
<td>69.33%</td>
<td>71.65%</td>
</tr>
<tr>
<td>debates</td>
<td>799</td>
<td>66.33%</td>
<td>70.07%</td>
<td>71.14%</td>
</tr>
<tr>
<td>entertainment</td>
<td>802</td>
<td>59.72%</td>
<td>64.98%</td>
<td>66.02%</td>
</tr>
<tr>
<td><strong>Avg.</strong></td>
<td>826</td>
<td>63.24%</td>
<td>68.24%</td>
<td>70.06%</td>
</tr>
<tr>
<td><strong>Senseval-2</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d00</td>
<td>471</td>
<td>51.70%</td>
<td>53.07%</td>
<td>58.17%</td>
</tr>
<tr>
<td>d01</td>
<td>784</td>
<td>60.80%</td>
<td>64.28%</td>
<td>67.85%</td>
</tr>
<tr>
<td>d02</td>
<td>514</td>
<td>55.97%</td>
<td>62.84%</td>
<td>63.81%</td>
</tr>
<tr>
<td><strong>Avg.</strong></td>
<td>590</td>
<td>56.15%</td>
<td>60.06%</td>
<td>63.27%</td>
</tr>
<tr>
<td><strong>Average (all)</strong></td>
<td>740</td>
<td>60.58%</td>
<td>65.17%</td>
<td>67.51%</td>
</tr>
</tbody>
</table>

“informed” (sense order integrated)
Till Now

• Graph-based **ranking** algorithm
• Smarter IR
• NLP - TextRank, LexRank
  – Text summarization
  – Keyword extraction
  – Word Sense Disambiguation
Other graph-based algorithms for NLP

• Find entities that satisfy certain structural properties defined with respect to other entities
• Find globally optimal solutions given relations between entities
• Min-Cut Algorithm
Subjectivity Analysis for Sentiment Classification

• The objective is to detect **subjective expressions** in text (opinions against facts)

• Use this information to improve the **polarity classification** (positive vs. negative)
  – E.g. Movie reviews (see: [www.rottentomatoes.com](http://www.rottentomatoes.com))

• Sentiment analysis can be considered as a **document classification problem**, with target classes focusing on the authors' sentiments, rather than topic-based categories
  – Standard machine learning classification techniques can be applied
Subjectivity Extraction

n-sentence review

subjective sentence?

m-sentence extract (m<=n)

positive or negative review?

default polarity classifier

subjectivity extraction
Subjectivity Detection/Extraction

• Detecting the subjective sentences in a text may be useful in filtering out the objective sentences creating a **subjective extract**

• Subjective extracts facilitate the **polarity analysis** of the text (increased accuracy at reduced input size)

• Subjectivity detection can use local and contextual features:
  – Contextual: uses context information, such as e.g. sentences occurring near each other tend to share the same subjectivity status (coherence)
  – Local: relies on individual sentence classifications using standard machine learning techniques (SVM, Naïve Bayes, etc) trained on an annotated data set

• *(Pang and Lee, 2004)*
Cut-based Subjectivity Classification

- Standard classification techniques usually consider only individual features (classify one sentence at a time).
- Cut-based classification takes into account both individual and contextual (structural) features.
Min-Cut definition

- Graph cut: partitioning the graph in two disjoint sets of nodes
- Graph cut weight:
  \[ \text{cut}(A, B) = \sum_{u \in A, v \in B} w(u, v) \]
  - i.e., sum of crossing edge weights
- Minimum cut: the cut that minimizes the cross-partition similarity
Modeling Individual Features

Diagram showing relationships between variables Y, M, N, and their entries.
Modeling Contextual Features
Collective Classification

• Suppose we have $n$ items $x_1, \ldots, x_n$ to divide in two classes: $C_1$ and $C_2$.

• Individual scores: $\text{ind}_j(x_i)$ - non-negative estimates of each $x_i$ being in $C_j$ based on the features of $x_i$ alone.

• Association scores: $\text{assoc}(x_i, x_k)$ - non-negative estimates of how important it is that $x_i$ and $x_k$ be in the same class.
Collective Classification

• Maximize each item’s assignment score (individual score for the class it is assigned to, minus its individual score for the other class), while penalize the assignment of different classes to highly associated items

• Formulated as an optimization problem: assign the $x_i$ items to classes $C_1$ and $C_2$ so as to minimize the partition cost:

$$\sum_{x \in C_1} ind_2(x) + \sum_{x \in C_2} ind_1(x) + \sum_{x_i \in C_1, x_k \in C_2} assoc(x_i, x_k)$$
Cut-based Algorithm

• There are $2^n$ possible binary partitions of the $n$ elements, we need an efficient algorithm to solve the optimization problem

• Build an undirected graph $G$ with vertices \( \{v_1, \ldots, v_n, s, t\} \) and edges:
  - \((s, v_i)\) with weights \(\text{ind}_1(x_i)\)
  - \((v_i, t)\) with weights \(\text{ind}_2(x_i)\)
  - \((v_i, v_k)\) with weights \(\text{assoc}(x_i, x_k)\)
Cut-based Algorithm (cont.)

• Cut: a partition of the vertices in two sets \((S, T)\)
• The cost is the sum of the weights of all edges crossing from \(S\) to \(T\)
• A minimum cut is a cut with the minimal cost
• A minimum cut can be found using maximum-flow algorithms, with polynomial asymptotic running times
• Use the min-cut / max-flow algorithm
Notice that without the structural information we would be undecided about the assignment of node $M$. 
Subjectivity Extraction

• Assign every individual sentence a subjectivity score
  – e.g. the probability of a sentence being subjective, as assigned by a Naïve Bayes classifier, etc

• Assign every sentence pair a proximity or similarity score
  – e.g. physical proximity = the inverse of the number of sentences between the two entities

• Use the min-cut algorithm to classify the sentences into objective/subjective
Subjectivity Extraction with Min-Cut

n-sentence review

\[ s_1 \\
\| \\
\| \\
\| \\
\| \\
\| s_n \]

construct graph

\[ P_{Pr_{sub}^{NB}}(s_l) \]

\[ 1 - P_{Pr_{sub}^{NB}}(s_l) \]

compute min. cut

individual subjectivity-probability link

proximity link

compute min. cut

create extract

\[ \times \text{ edge crossing the cut} \]

\[ \text{m-sentence extract} \]

\[ (m <= n) \]

\[ s_1 \\
\| \\
\| \\
\| \\
\| \vdots \\
\| s_4 \]
Results

• 2000 movie reviews (1000 positive / 1000 negative)

• The use of subjective extracts improves or maintains the accuracy of the polarity analysis while reducing the input data size