Discourse and Summarization

LING 573 — Systems and Applications
April 12th, 2018
Begin Recording!
Miscellanea
What is a Centroid?

- Way to define the “middle” of a cluster
- In document clustering setting, centroid often:
  - Vector representation of “model” document
  - highest similarity to the most other documents in the cluster
- Can also be a “pseudo-document”
  - Words picked from all documents rather than single document
What Does a MEAD Centroid Look Like?

- As computed by CIDR clustering algorithm
- **R code on Github**

- Single-pass clustering
  - Filter words based on their tf*idf
  - N best +/- above word threshold
  - Join with new cluster if above cluster similarity threshold

<table>
<thead>
<tr>
<th>word</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td>suharto</td>
<td>2.48</td>
</tr>
<tr>
<td>jakarta</td>
<td>0.58</td>
</tr>
<tr>
<td>habibie</td>
<td>0.47</td>
</tr>
<tr>
<td>students</td>
<td>0.45</td>
</tr>
<tr>
<td>student</td>
<td>0.22</td>
</tr>
<tr>
<td>protesters</td>
<td>0.20</td>
</tr>
<tr>
<td>asean</td>
<td>0.11</td>
</tr>
<tr>
<td>campuses</td>
<td>0.05</td>
</tr>
<tr>
<td>geertz</td>
<td>0.04</td>
</tr>
<tr>
<td>medan</td>
<td>0.04</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>word</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td>suharto</td>
<td>2.61</td>
</tr>
<tr>
<td>jakarta</td>
<td>0.58</td>
</tr>
<tr>
<td>habibie</td>
<td>0.53</td>
</tr>
<tr>
<td>students</td>
<td>0.43</td>
</tr>
<tr>
<td>student</td>
<td>0.21</td>
</tr>
<tr>
<td>protesters</td>
<td>0.19</td>
</tr>
<tr>
<td>asean</td>
<td>0.10</td>
</tr>
<tr>
<td>campuses</td>
<td>0.04</td>
</tr>
<tr>
<td>geertz</td>
<td>0.04</td>
</tr>
<tr>
<td>medan</td>
<td>0.04</td>
</tr>
</tbody>
</table>

Figure 1: Centroid for cluster 44 (the two scores are after 10,000 (left) and all 22,443 documents (right).
LexRank Revisited

- Begin by computing cosine similarity matrix between sentences in cluster

```
<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.00</td>
<td>0.45</td>
<td>0.02</td>
<td>0.17</td>
<td>0.03</td>
<td>0.22</td>
<td>0.03</td>
<td>0.28</td>
<td>0.06</td>
<td>0.06</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>0.45</td>
<td>1.00</td>
<td>0.16</td>
<td>0.27</td>
<td>0.03</td>
<td>0.19</td>
<td>0.03</td>
<td>0.21</td>
<td>0.03</td>
<td>0.15</td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td>0.02</td>
<td>0.16</td>
<td>1.00</td>
<td>0.03</td>
<td>0.00</td>
<td>0.01</td>
<td>0.03</td>
<td>0.04</td>
<td>0.00</td>
<td>0.01</td>
<td>0.00</td>
</tr>
<tr>
<td>4</td>
<td>0.17</td>
<td>0.27</td>
<td>0.03</td>
<td>1.00</td>
<td>0.01</td>
<td>0.16</td>
<td>0.28</td>
<td>0.17</td>
<td>0.00</td>
<td>0.09</td>
<td>0.01</td>
</tr>
<tr>
<td>5</td>
<td>0.03</td>
<td>0.03</td>
<td>0.00</td>
<td>0.01</td>
<td>1.00</td>
<td>0.29</td>
<td>0.05</td>
<td>0.15</td>
<td>0.20</td>
<td>0.04</td>
<td>0.18</td>
</tr>
<tr>
<td>6</td>
<td>0.22</td>
<td>0.19</td>
<td>0.01</td>
<td>0.16</td>
<td>0.29</td>
<td>1.00</td>
<td>0.05</td>
<td>0.29</td>
<td>0.04</td>
<td>0.20</td>
<td>0.03</td>
</tr>
<tr>
<td>7</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.28</td>
<td>0.05</td>
<td>0.05</td>
<td>1.00</td>
<td>0.06</td>
<td>0.00</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>8</td>
<td>0.28</td>
<td>0.21</td>
<td>0.04</td>
<td>0.17</td>
<td>0.15</td>
<td>0.29</td>
<td>0.06</td>
<td>1.00</td>
<td>0.25</td>
<td>0.20</td>
<td>0.17</td>
</tr>
<tr>
<td>9</td>
<td>0.06</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
<td>0.20</td>
<td>0.04</td>
<td>0.00</td>
<td>0.25</td>
<td>1.00</td>
<td>0.26</td>
<td>0.38</td>
</tr>
<tr>
<td>10</td>
<td>0.06</td>
<td>0.15</td>
<td>0.01</td>
<td>0.09</td>
<td>0.04</td>
<td>0.20</td>
<td>0.00</td>
<td>0.20</td>
<td>0.26</td>
<td>1.00</td>
<td>0.12</td>
</tr>
<tr>
<td>11</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.01</td>
<td>0.18</td>
<td>0.03</td>
<td>0.01</td>
<td>0.17</td>
<td>0.38</td>
<td>0.12</td>
<td>1.00</td>
</tr>
</tbody>
</table>
```
LexRank Revisited

- Use these initial weights to build a graph between sentences
- Cosine similarity sets weights of edges
LexRank Revisited

- Next step: compute node ranks:
  - What we want is ultimately a vector, where each element is the score for our node
  - This is the eigenvector of our weight matrix
  - Represents stable distribution of markov chain
LexRank Revisited

- Use Power Method: series of matrix transformations:
  - Start with initial guess for eigenvector $x$
  - Calculate $w = Ax$ [$w$ is new matrix]
  - Largest magnitude column in $w$ is estimate of eigenvalue
  - Re-scale $w$ by eigenvalue to get next guess for eigenvector $x$
  - Repeat until convergence
Example of power method converging toward approximation

<table>
<thead>
<tr>
<th>$A_z^{(4)}$</th>
<th>2 8 10</th>
<th>0.9243</th>
<th>17.513</th>
<th>0.9181</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8 3 4</td>
<td>0.7080</td>
<td>13.519</td>
<td>0.7087</td>
</tr>
<tr>
<td></td>
<td>10 4 7</td>
<td>1.0</td>
<td>19.075</td>
<td>1.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$A_z^{(5)}$</th>
<th>2 8 10</th>
<th>0.9181</th>
<th>17.506</th>
<th>0.9206</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8 3 4</td>
<td>0.7087</td>
<td>13.471</td>
<td>0.7084</td>
</tr>
<tr>
<td></td>
<td>10 4 7</td>
<td>1.0</td>
<td>19.016</td>
<td>1.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$A_z^{(6)}$</th>
<th>2 8 10</th>
<th>0.9206</th>
<th>17.508</th>
<th>0.9196</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8 3 4</td>
<td>0.7084</td>
<td>13.490</td>
<td>0.7085</td>
</tr>
<tr>
<td></td>
<td>10 4 7</td>
<td>1.0</td>
<td>19.040</td>
<td>1.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$A_z^{(7)}$</th>
<th>2 8 10</th>
<th>0.9196</th>
<th>17.507</th>
<th>0.9200</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8 3 4</td>
<td>0.7085</td>
<td>13.482</td>
<td>0.7085</td>
</tr>
<tr>
<td></td>
<td>10 4 7</td>
<td>1.0</td>
<td>19.030</td>
<td>1.0</td>
</tr>
</tbody>
</table>
LexRank Revisited

- Example of power method converging toward approximation

\[
\begin{bmatrix}
2 & 8 & 10 \\
8 & 3 & 4 \\
10 & 4 & 7 \\
\end{bmatrix}
\begin{bmatrix}
0.9243 \\
0.7080 \\
1.0 \\
\end{bmatrix}
= 
\begin{bmatrix}
17.513 \\
13.519 \\
19.075 \\
\end{bmatrix}
= (19.075) \\
\begin{bmatrix}
0.9181 \\
0.7087 \\
1.0 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
2 & 8 & 10 \\
8 & 3 & 4 \\
10 & 4 & 7 \\
\end{bmatrix}
\begin{bmatrix}
0.9181 \\
0.7087 \\
1.0 \\
\end{bmatrix}
= 
\begin{bmatrix}
17.506 \\
13.471 \\
19.016 \\
\end{bmatrix}
= (19.016) \\
\begin{bmatrix}
0.9206 \\
0.7084 \\
1.0 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
2 & 8 & 10 \\
8 & 3 & 4 \\
10 & 4 & 7 \\
\end{bmatrix}
\begin{bmatrix}
0.9206 \\
0.7084 \\
1.0 \\
\end{bmatrix}
= 
\begin{bmatrix}
17.508 \\
13.490 \\
19.040 \\
\end{bmatrix}
= (19.040) \\
\begin{bmatrix}
0.9196 \\
0.7085 \\
1.0 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
2 & 8 & 10 \\
8 & 3 & 4 \\
10 & 4 & 7 \\
\end{bmatrix}
\begin{bmatrix}
0.9196 \\
0.7085 \\
1.0 \\
\end{bmatrix}
= 
\begin{bmatrix}
17.507 \\
13.482 \\
19.030 \\
\end{bmatrix}
= (19.030) \\
\begin{bmatrix}
0.9200 \\
0.7085 \\
1.0 \\
\end{bmatrix}
\]
LexRank Revisited

- Don’t worry, we don’t expect you to have linear algebra nailed!
LexRank Demo

- Link to LexRank Demo: http://clair.si.umich.edu/demos/lexrank/
Experimental Setup

- Design different features, both *discourse-related* and *non-discourse*
  - Using model summaries (human-generated)
    - Perform statistical significance tests on included vs. non-included sentences
      - $\chi^2$ (categorical) t-test (continuous)
  - Use features in logistic regression classifier (MaxEnt)
    - Use to select sentences for extraction
- Evaluation:
  - $F_1$ against model sentences
  - ROUGE over summary sentences
Experimental Setup

- Caveat:
  - Experimental approach is using human-created discourse analyses
  - Authors do not attempt using automatic discourse parsers for analyses
  - Purely a study of how well discourse features correlate in an idealized setting
How Would This be Applied?

- Learn and apply classifiers for segmentation and parsing of discourse
- Assign coherence relations between spans
- Create a representation over whole text \( \rightarrow \) parse
- Use parsed representations as features in classifier for content selection
Discourse (RST) Structure Example

1. [Mr. Watkins said]
2. [volume on Interprovincial’s system is down about 2% since January]
3. [and is expected to fall further,]
4. [making expansion unnecessary until perhaps the mid-1990s.]
Discourse Structure Features

- **Satellite penalty**
  - For each EDU — number of satellite nodes between EDU and root
  - 1 satellite in tree: one step to root: penalty = 1
  - **Intuition**: Helpful summary content will be closely related to nucleus.
Discourse Structure Features

- **Promotion set:**
  - Nuclear units at some level of tree
  - At leaves, EDUs are themselves nuclear
  - Intuition:
    - The more times a unit is promoted in the tree, the more necessary its concepts to understanding the whole discourse
Discourse Structure Features

- **Depth score:**
  - Distance from lowest tree level to EDU’s highest rank
  - 2, 3, 4: score = 4
  - 1: score = 3

- **Promotion score:**
  - # of levels span is promoted
  - 1: score = 0
  - 4: score = 2
  - 2, 3: score = 3
Converting to Sentence Level

- Each feature has:
  - Raw score
  - Normalized score: \( \frac{\text{raw score}}{\text{sentence length}} \)

- Sentence score for a feature:
  - Maximum value over all EDUs in sentence
“Semantic” Features

- Represent sentences purely in terms of their discourse relationships

- **Binary features:**
  - Implicit vs. Explicit
  - `sentence_in_{RELATION_NAME}`
  - `sentence_contains_{ARG1|ARG2}_of_{RELATION_NAME}` (multi-sentential)
  - `sentence_expresses_{RELATION_NAME}` (both args in single sent)

- **Real-valued features:**
  - Number of relations
  - Distance between arguments within sentence
Example 1

- *In addition, its machines are easier to operate, so customers require less assistance from software.*

- Is there an explicit discourse marker?
  - Yes, “so”

- Discourse relation?
  - Contingency
Example 11

● (1) Wednesday’s dominant issue was Yasuda & Marine Insurance, which continued to surge on rumors of speculative buying. (2) It ended the day up 80 yen to 1880 yen.

● Is there an explicit discourse marker?
  ● No

● Is there a relation?
  ● Yes, Implicit.

● What relation?
  ● Expansion. (More specifically, restatement).
Non-Discourse Features

- Sentence length
- Sentence position
- Probabilities of words in sentence
  - mean, sum, product
- # of signature words (LLR)
Statistical Analysis
Statistical Analysis

- Used model summaries to analyze whether features were predictive
  - for a given feature-sent pair in docset…
  - How likely was that sentence to appear in summary?
Significant Features: Summary Sentences

- **Structure:**
  - depth score
  - promotion score

- **Semantic:**
  - Arg1 of Explicit Expansion
  - Arg1 of Implicit Contingency
  - Arg1 of Implicit Expansion
  - Distance to other Argument

- **Non-discourse:**
  - length
  - 1st sent in article
  - 1st sent in paragraph
  - offset from paragraph end
  - # signature terms
  - mean content word probabilities
  - sum content word probabilities

All VERY small p-values
Significant Features: Non-Summary Sentences

- **Structure:**
  - satellite penalty

- **Semantic:**
  - expresses explicit expansion
  - expresses explicit contingency
  - Arg2 of implicit temporal
  - Arg2 of implicit expansion
  - Arg2 of implicit contingency
  - # of shared implicit relations
  - total shared relations

- **Non-discourse:**
  - offset from paragraph start
  - offset from article start
  - sentence probability
Observations

- Non-discourse features good cues to summary
- Structural features match intuition
- Semantic features
  - Relatively few useful features for selecting summaries
  - Most features associated with non-summary… but most sentences are non-summary
Evaluation

- Structural is best, both alone and in combination
- Best overall combines all types
- Both $F_1$ and ROUGE-1

<table>
<thead>
<tr>
<th>Features used</th>
<th>Acc</th>
<th>P</th>
<th>R</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>structural</td>
<td>78.11</td>
<td>63.38</td>
<td>22.77</td>
<td>33.50</td>
</tr>
<tr>
<td>semantic</td>
<td>75.53</td>
<td>44.31</td>
<td>5.04</td>
<td>9.05</td>
</tr>
<tr>
<td>non-discourse (ND)</td>
<td>77.25</td>
<td>67.48</td>
<td>11.02</td>
<td>18.95</td>
</tr>
<tr>
<td>ND + semantic</td>
<td>77.38</td>
<td>59.38</td>
<td>20.62</td>
<td>30.61</td>
</tr>
<tr>
<td>ND + structural</td>
<td>78.51</td>
<td>63.49</td>
<td>26.05</td>
<td>36.94</td>
</tr>
<tr>
<td>semantic + structural</td>
<td>77.94</td>
<td>58.39</td>
<td>30.47</td>
<td>40.04</td>
</tr>
<tr>
<td>structural + semantic + ND</td>
<td>78.93</td>
<td>61.85</td>
<td>34.42</td>
<td>44.23</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Features</th>
<th>ROUGE</th>
<th>Features</th>
<th>ROUGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>structural + semantic + ND</td>
<td>0.479</td>
<td>ND</td>
<td>0.432</td>
</tr>
<tr>
<td>structural + ND</td>
<td>0.468</td>
<td>LEAD</td>
<td>0.411</td>
</tr>
<tr>
<td>structural + semantic</td>
<td>0.453</td>
<td>semantic</td>
<td>0.369</td>
</tr>
<tr>
<td>semantic + ND</td>
<td>0.444</td>
<td>TS</td>
<td>0.338</td>
</tr>
<tr>
<td>structural</td>
<td>0.433</td>
<td>*TS = “topic signature”</td>
<td></td>
</tr>
</tbody>
</table>
Graph-Based Comparison

- Page-Rank Based Centrality Computed Over
  - RST Link Structure
  - Graphbank Link Structure
  - LexRank (sentence cosine similarity)

- Quite similar, but:
  - $F_1$: LR > GB > RST
  - ROUGE: RST > LR > GB

<table>
<thead>
<tr>
<th></th>
<th>Acc</th>
<th>P</th>
<th>R</th>
<th>F</th>
<th>ROUGE-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>RST-struct</td>
<td>81.61</td>
<td>63.00</td>
<td>31.56</td>
<td>42.05</td>
<td>0.569</td>
</tr>
<tr>
<td>GB-struct</td>
<td>82.58</td>
<td>62.50</td>
<td>39.16</td>
<td>48.15</td>
<td>0.508</td>
</tr>
<tr>
<td>LEX-struct</td>
<td>83.23</td>
<td>75.17</td>
<td>41.14</td>
<td>53.18</td>
<td>0.557</td>
</tr>
</tbody>
</table>
Notes

- Single document, short (100 word) summaries
  - What about multi-document? Longer?

- Structure relatively better

- Manually labeled discourse structure, relations
  - Some automatic systems available, but not perfect
  - Better at getting the structure than the exact relation
    - Especially implicit
Topic Orientation & Optimization
Topic-Focused Summarization

- “Query-focused” or “Guided”
- Extrinsic task vs. generic:
  - Why are we creating this summary?
  - Viewed as complex question answering (vs. factoid)
- High variation in human summaries
  - Depending on perspective, different content is focused
Topic-Focused Summarization: Key Idea

- Target response to specific question, topic in documents
- Later TACs identify topic categories and aspects
  - e.g. Natural disasters: who, what, where, when
When treated as a factoid/sentence selection problem:

- Mean Rank Reciprocal (MRR)
  - Inverse of rank of correct answer
- Total Reciprocal Document Rank (TRDR)
  - Total of all reciprocal ranks of all answers system suggests
  - (Usually taken as average)
Query-Focused LexRank

Otterbacher et al (2005)

- Focus on sentences relevant to query
- Rather than computing similarity of sentences to all other sentences

How do we measure relevance?

- tf*idf-like measure over sentences & query
- Compute sentence-level “idf<sub>w</sub>”
- \( N = \# \) of sentences in cluster
- \( idf<sub>w</sub> = \log \left( \frac{N + 1}{0.5 + sf<sub>w</sub>} \right) \)
- \( sf<sub>w</sub> = \# \) of sentences with \( w \)
Query-Focused LexRank

Otterbacher et al (2005)

\[ rel(s | q) = \sum_{w \in q} \log(tf_{w,s} + 1) \cdot \log(tf_{w,q} + 1) \cdot idf_w \]

- Relevance of sentence \( s \) given query \( q \)
- Log Sum (Product) of:
  - term frequency for word \( w \) in sentence
  - term frequency for word \( w \) in query
  - \( idf_w \) for word across all sentences
Updated LexRank Model

- Combines original similarity weighting with query
- Mixture model of query relevance, sentence similarity (LexRank)

\[ p(s \mid q) = d \frac{\text{rel}(s \mid q)}{\sum_{z \in C} \text{rel}(z \mid q)} + (1 - d) \left( \sum_{v \in C} \frac{\text{sim}(s, v)}{\sum_{z \in C} \text{sim}(z, v)} \cdot p(v \mid q) \right) \]

- \( d \) controls “bias”: i.e. relative weighting toward query relevance
Tuning & Assessment

- Parameters:
  - **Similarity threshold**: filters adjacency matrix
  - **Question bias**: Weights emphasis on question focus

- Empirical results:
  - Best similarity threshold: 0.14–0.2
  - Best question bias: high: 0.8–0.95
  - Higher question bias in LexRank improves MRR
Other Strategies

- Methods depend on base system design
  - All aim to incorporate similarity with query/topic

- CLASSY HMM (Conroy et al., 2005):
  - Add question overlap feature to HMM vector — $\log(#\_query\_tokens\_in\_sentence + 1)$
  - Query tokens: filtered to NN, VB, JJ, RB, or NNP

- FastSum (Schilder & Kondadadi, 2008):
  - SVM regression on sentences
  - Adds topic title frequency feature:
    - Proportion of words in sent which appear in title

- Others: require minimum number of topic words
Overview

- Many similar strategies:
  - Features, weighting, ranking: overlap based

- Actual evaluation impact:
  - Not necessarily very large (e.g. 0.003 ROUGE)
    - But can be useful

- Aggressive approaches can have large negative impact
  - i.e. explicitly adding NER spans
Optimization Approaches to Reducing Redundancy
Optimization Approaches to Reducing Redundancy

- DPP: Determinantal Point Processes [python GH] (Kulesza & Taskar 2012)
  - Set models balancing information importance w/diversity

  - Optimizes coverage of key bigrams weighted by document frequency

- OCCAMS_V (Davis et al, 2012)
  - Uses LSA (Latent Semantic Analysis) to weight terms
  - Sentence selection via optimization problems:
    - Budgeted maximal coverage; knapsack
ICSISumm

Key ideas:
- Cast summarization as optimization problem
- Identify important “concepts” to incorporate
- Build best such summary
- Implemented as Integer Linear Programming
Integer Linear Programming

- Aka ILP

- An integer linear program specifies
  - A single linear maximization term
  - Subject to linear equality/inequality constraints
  - Involving integer valued variables

- For summarization:
  - Map summary requirements to ILP elements
Summarization as ILP

- **Summary goal:**
  - “Best” summary

- **Summary requirements:**
  - Minimize redundancy
  - Within desired length

- **Maximization term:**
  \[
  \sum_i W_i C_i
  \]

- **Implicit:**
  - Length Constraint
    \[
    \sum_j l_j s_j < L
    \]
  - Coverage Constraint
    \[
    \sum_j s_j o_{ij} \geq c_i \forall i
    \]
    \[
    s_j o_{ij} \leq c_i \forall i, j
    \]

Weight
Representing Concepts

- Concepts = Bigrams
  - Stemmed
  - No stopword-only bigrams
  - Occuring in at least 3 documents

- Weights
  - Document frequency
  - # Of Documents (from cluster) for bigram

- Selected sentences must contain ≥ 2 query terms
Results

- After using open source solver

- 2009 results:
  - 2nd best pyramid, ROUGE-2
  - Best ROUGE-3, ROUGE-4