Content Selection: Supervision & Discourse

Ling573
Systems & Applications
April 11, 2017
Roadmap

- Content selection
  - Supervised content selection
    - Analysis & Regression with rich features
  - “CLASSY”: HMM methods
- Discourse structure
  - Models of discourse structure
  - Structure and relations for summarization
Supervised Word Selection

- RegSumm:
  - Improving the Estimation of Word Importance for News Multi-Document Summarization (Hong & Nenkova, ’14)

- Key ideas:
  - Supervised method for word selection
  - Diverse, rich feature set: unsupervised measures, POS, NER, position, etc
  - Identification of common “important” words via side corpus of news articles and human summaries
Basic Approach

- Learn keyword importance
  - Contrasts with unsupervised selection, learning sentences

- Train regression over large number of possible features
  - Supervision over *words*
    - Did document word appear in summary or not?

- Greedy sentence selection:
  - Highest scoring sentences: average word weight
  - Do not add if >= 0.5 cosine similarity w/any curr sents
Features I

- Unsupervised measures:
  - Used as binary features given some threshold
  - Word probability: \( \frac{\text{count}(w)}{N} \)
    - Computed over input cluster
  - Log likelihood ratio: Gigaword as background corpus

- Markov Random Walk (MRW):
  - Graphical model approach similar to LexRank
  - Nodes: words
  - Edges: \# syntactic dependencies b/t wds in sentences
  - Weights via PageRank algorithm
“Global” word importance:

Question: Are there words which are intrinsically likely to show up in (news) summaries?

Approach:

- Build language models on NYT corpus of articles+summs
  - One model on articles, one model on summaries
  - Measures: $Pr_A(w)$, $Pr_A(w) - Pr_G(w)$, $Pr_A(w)/Pr_G(w)$
    - $KL(A||G) = Pr_A(w) \times \ln \left( \frac{Pr_A(w)}{Pr_G(w)} \right)$
    - $KL(G||A) = Pr_G(w) \times \ln \left( \frac{Pr_G(w)}{Pr_A(w)} \right)$
  - Binary features: top-k or bottom-k features
Features III

- Adaptations of common features:
  - Word position as proportion of document $[0,1]$
    - Earliest first, latest last, average, average first
  - Word type: POS, NER
    - Emphasizes NNS, NN, capitalization; ORG, PERS, LOC

- MPQA and LIWC features:
  - MPQA: sentiment, subjectivity terms
    - Strong sentiment likely or not? NOT
  - LIWC: words for 64 categories: +: death, anger, money
    - Neg: pron, neg, fn words, swear, adverbs, etc
Assessment: Words

- Select N highest ranked keywords via regression
- Compute F-measure over words in summaries
  - \( G_i \): \( i = \# \) of summaries in which word appears

<table>
<thead>
<tr>
<th>( G_i )</th>
<th>#words</th>
<th>PROB</th>
<th>LLR</th>
<th>MRW</th>
<th>REGBASIC</th>
<th>REGSUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>( G_1 )</td>
<td>80</td>
<td>43.6</td>
<td>37.9</td>
<td>38.9</td>
<td>39.9</td>
<td>45.7</td>
</tr>
<tr>
<td>( G_1 )</td>
<td>100</td>
<td>44.3</td>
<td>38.7</td>
<td>39.2</td>
<td>41.0</td>
<td>46.5</td>
</tr>
<tr>
<td>( G_1 )</td>
<td>120</td>
<td>44.6</td>
<td>38.5</td>
<td>39.2</td>
<td>40.9</td>
<td>46.4</td>
</tr>
<tr>
<td>( G_2 )</td>
<td>30</td>
<td>47.8</td>
<td>44.0</td>
<td>42.4</td>
<td>47.4</td>
<td>50.2</td>
</tr>
<tr>
<td>( G_2 )</td>
<td>35</td>
<td>47.1</td>
<td>43.3</td>
<td>42.1</td>
<td>47.0</td>
<td>49.5</td>
</tr>
<tr>
<td>( G_2 )</td>
<td>40</td>
<td>46.5</td>
<td>42.4</td>
<td>41.8</td>
<td>46.4</td>
<td>49.2</td>
</tr>
</tbody>
</table>
Assessment: Summaries

- Compare summarization w/ ROUGE-1,2,4

<table>
<thead>
<tr>
<th>System</th>
<th>R-1</th>
<th>R-2</th>
<th>R-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROB</td>
<td>35.14</td>
<td>8.17</td>
<td>1.06</td>
</tr>
<tr>
<td>LLR</td>
<td>34.60</td>
<td>7.56</td>
<td>0.83</td>
</tr>
<tr>
<td>MRW</td>
<td>35.78</td>
<td>8.15</td>
<td>0.99</td>
</tr>
<tr>
<td>REGBASIC</td>
<td>37.56</td>
<td>9.28</td>
<td>1.49</td>
</tr>
<tr>
<td>KL</td>
<td>37.97</td>
<td>8.53</td>
<td>1.26</td>
</tr>
<tr>
<td>PEER-65</td>
<td>37.62</td>
<td>8.96</td>
<td>1.51</td>
</tr>
<tr>
<td>SUBMOD</td>
<td>39.18</td>
<td>9.35</td>
<td>1.39</td>
</tr>
<tr>
<td>DPP</td>
<td>39.79</td>
<td>9.62</td>
<td>1.57</td>
</tr>
<tr>
<td>REGSUM</td>
<td>38.57</td>
<td>9.75</td>
<td>1.60</td>
</tr>
</tbody>
</table>

Basic Systems

State of The Art Systems
CLASSY

- “Clustering, Linguistics and Statistics for Summarization Yield”
  - Conroy et al. 2000-2011

Highlights:
- High performing system
  - Often rank 1 in DUC/TAC, commonly used comparison
- Topic signature-type system (LLR)
- HMM-based content selection
- Redundancy handling
Using LLR for Weighting

- Compute weight for all cluster terms
  - \( \text{weight}(w_i) = 1 \) if \(-2\log \lambda > 10\), 0 o.w.

- Use that to compute sentence weights

\[
\text{weight}(s_i) = \sum_{w \in s_i} \frac{\text{weight}(w)}{|\{w | w \in s_i\}|}
\]

- How do we use the weights?
  - One option: directly rank sentences for extraction

- LLR-based systems historically perform well
  - Better than tf*idf generally
HMM Sentence Selection

- CLASSY strategy: Use LLR as feature in HMM
- How does HMM map to summarization?
  - Key idea:
    - Two classes of states: summary, non-summary
    - Feature(s)?: log(#sig+1) (tried: length, position,..)
    - Lower cased, white-space tokenized (a-z), stopped
  - Topology:

- Select sentences with highest posterior (in “summary”)
Matrix-based Selection

- Redundancy minimizing selection
- Create term x sentence matrix
  - If term in sentence, weight is nonzero
- Loop:
  - Select highest scoring sentence
    - Based on Euclidean norm
  - Subtract those components from remaining sentences
  - Until enough sentences
- Effect: selects highly ranked but different sentences
  - Relatively insensitive to weighting schemes
Combining Approaches

- Both HMM and Matrix method select sentences
- Can combine to further improve

**Approach:**
- Use HMM method to compute sentence scores
  - (e.g. rather than just weight based)
    - Incorporates context information, prior states
- Loop:
  - Select highest scoring sentence
  - Update matrix scores
    - Exclude those with too low matrix scores
  - Until enough sentences are found
Other Linguistic Processing

- **Sentence manipulation (before selection):**
  - Remove uninteresting phrases based on POS tagging
    - Gerund clauses, restr. rel. appos, attrib, lead adverbs

- **Coreference handling (Serif system):**
  - Created coref chains initially
  - Replace all mentions with longest mention (# caps)
  - Used only for sentence selection
Outcomes

- HMM, Matrix: both effective, better combined

- Linguistic pre-processing improves
  - Best ROUGE-1, ROUGE-2 in DUC

- Coref handling improves:
  - Best ROUGE-3, ROUGE-4; 2nd ROUGE-2
Notes

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

- Structure relatively better, all contribute

- Manually labeled discourse structure, relations
  - Some automatic systems, but not perfect
    - However, better at structure than relation ID
      - Esp. implicit