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: 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

Features II

- "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: $Pr_A(w)$, $Pr_A(w)-Pr_G(w)$, $Pr_A(w)/Pr_G(w)$
        - $KL(A||G) = Pr_A(w)*\ln(Pr_A(w)/Pr_G(w))$
        - $KL(G||A) = Pr_G(w)*\ln(Pr_G(w)/Pr_A(w))$
    - 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

G_i	#words	PROB	LLR	MRW	REGBASIC	**REGSUM**
G_1	80	43.6	37.9	38.9	39.9	45.7
G_1	100	44.3	38.7	39.2	41.0	46.5
G_1	120	44.6	38.5	39.2	40.9	46.4
G_2	30	47.8	44.0	42.4	47.4	50.2
G_2	35	47.1	43.3	42.1	47.0	49.5
G_2	40	46.5	42.4	41.8	46.4	49.2

Assessment: Summaries

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

Basic Systems

State of The Art Systems

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

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

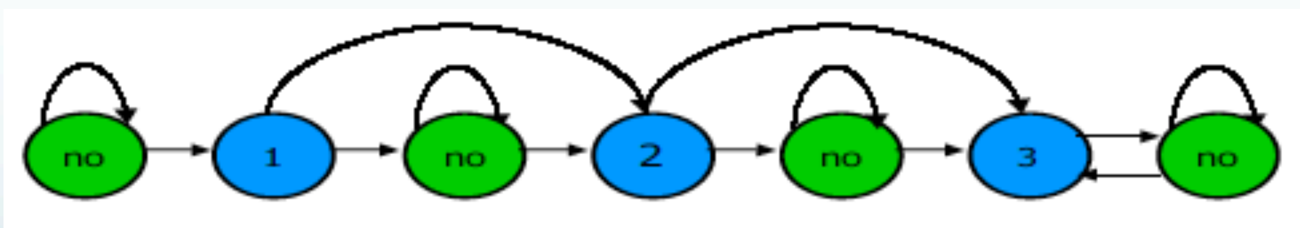
- Compute weight for all cluster terms
  - $weight(w_i) = 1$ if $-2\log\lambda > 10$, 0 o.w.
- Use that to compute sentence weights

$$weight(s_i) = \sum_{w \in s_i} \frac{weight(w)}{|\{w | w \in s_i\}|}$$

- 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; 2^{nd} 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