Entity- & Topic-Based Information Ordering

Ling 573
Systems and Applications
May 7, 2015

Roadmap

- Entity-based cohesion model:
  - Model entity based transitions
- Topic-based cohesion model:
  - Models sequence of topic transitions
- Ordering as optimization

Entity Grid

- Need compact representation of:
  - Mentions, grammatical roles, transitions
    - Across sentences
- Entity grid model:
  - Rows: sentences
  - Columns: entities
  - Values: grammatical role of mention in sentence
    - Roles: (S)ubject, (O)bject, X (other), __ (no mention)
    - Multiple mentions: ? Take highest

	Department	Trial	Microsoft	Evidence	Competitors	Markets	Products	Brands	Case	Netscape	Software	Tactics	Government	Suit	Earnings	
1	s	o	s	x	o	–	–	–	–	–	–	–	–	–	–	1
2	–	–	o	–	–	x	s	o	–	–	–	–	–	–	–	2
3	–	–	s	o	–	–	–	–	s	o	o	–	–	–	–	3
4	–	–	s	–	–	–	–	–	–	–	–	s	–	–	–	4
5	–	–	–	–	–	–	–	–	–	–	–	–	s	o	–	5
6	–	x	s	–	–	–	–	–	–	–	–	–	–	–	o	6

1 [The Justice Department]$_s$ is conducting an [anti-trust trial]$_o$ against [Microsoft Corp.]$_x$ with [evidence]$_x$ that [the company]$_s$ is increasingly attempting to crush [competitors]$_o$.
2 [Microsoft]$_o$ is accused of trying to forcefully buy into [markets]$_x$ where [its own products]$_s$ are not competitive enough to unseat [established brands]$_o$.
3 [The case]$_s$ revolves around [evidence]$_o$ of [Microsoft]$_s$ aggressively pressuring [Netscape]$_o$ into merging [browser software]$_o$.
4 [Microsoft]$_s$ claims [its tactics]$_s$ are commonplace and good economically.
5 [The government]$_s$ may file [a civil suit]$_o$ ruling that [conspiracy]$_s$ to curb [competition]$_o$ through [collusion]$_x$ is [a violation of the Sherman Act]$_o$.
6 [Microsoft]$_s$ continues to show [increased earnings]$_o$ despite [the trial]$_x$.

Grids → Features

- Intuitions:
  - Some columns dense: focus of text (e.g. MS)
    - Likely to take certain roles: e.g. S, O
  - Others sparse: likely other roles (x)
  - Local transitions reflect structure, topic shifts

Grids → Features

- Intuitions:
  - Some columns dense: focus of text (e.g. MS)
    - Likely to take certain roles: e.g. S, O
  - Others sparse: likely other roles (x)
  - Local transitions reflect structure, topic shifts
- Local entity transitions: $\{s,o,x,\_\}^n$
  - Continuous column subsequences (role n-grams?)
  - Compute probability of sequence over grid:
    - # occurrences of that type/# of occurrences of that len

Vector Representation

- Document vector:
  - Length

Vector Representation

- Document vector:
  - Length: # of transition types
  - Values:

Vector Representation

- Document vector:
  - Length: # of transition types
  - Values: Probabilities of each transition type

	S S	S O	S X	S –	O S	O O	O X	O –	X S	X O	X X	X –	– S	– O	– X	– –
d_1	.01	.01	0	.08	.01	0	0	.09	0	0	0	.03	.05	.07	.03	.59
d_2	.02	.01	.01	.02	0	.07	0	.02	.14	.14	.06	.04	.03	.07	0.1	.36
d_3	.02	0	0	.03	.09	0	.09	.06	0	0	0	.05	.03	.07	.17	.39

- Can vary by transition types:
  - E.g. most frequent; all transitions of some length, etc

Dependencies & Comparisons

- Tools needed:

Dependencies & Comparisons

- Tools needed:
  - Coreference: Link mentions
    - Full automatic coref system vs

Dependencies & Comparisons

- Tools needed:
  - Coreference: Link mentions
    - Full automatic coref system vs
    - Noun clusters based on lexical match
  - Grammatical role:
    - Extraction based on dependency parse (+passive rule) vs

Dependencies & Comparisons

- Tools needed:
  - Coreference: Link mentions
    - Full automatic coref system vs
    - Noun clusters based on lexical match

  - Grammatical role:
    - Extraction based on dependency parse (+passive rule) vs
    - Simple present vs absent (X, _)

Dependencies & Comparisons

- Tools needed:
  - Coreference: Link mentions
    - Full automatic coref system vs
    - Noun clusters based on lexical match
  - Grammatical role:
    - Extraction based on dependency parse (+passive rule) vs
    - Simple present vs absent (X, _)
- Salience:
  - Distinguish focused vs not:? By frequency
  - Build different transition models by saliency group

Experiments & Analysis

- Trained SVM:
  - Salient: >= 2 occurrences; Transition length: 2
  - Train/Test: Is higher manual score set higher by system?
- Feature comparison: DUC summaries

Model	Accuracy
Coreference+Syntax+Salience+	80.0
Coreference+Syntax+Salience−	75.0
Coreference+Syntax−Salience+	78.8
Coreference−Syntax+Salience+	**83.8**
Coreference+Syntax−Salience−	71.3*
Coreference−Syntax+Salience−	78.8
Coreference−Syntax−Salience+	77.5
Coreference−Syntax−Salience−	73.8*

Discussion

- Best results:
  - Use richer syntax and salience models
    - But **NOT** coreference (though not significant)
      - Why

Discussion

- Best results:
  - Use richer syntax and salience models
    - But **NOT** coreference (though not significant)
      - Why? Automatic summaries in training, unreliable coref
- Worst results:
  - Significantly worse with both simple syntax, no salience
    - Extracted sentences still parse reliably
  - Still not horrible: 74% vs 84%

Discussion

- Best results:
  - Use richer syntax and salience models
    - But **NOT** coreference (though not significant)
      - Why? Automatic summaries in training, unreliable coref
- Worst results:
  - Significantly worse with both simple syntax, no salience
    - Extracted sentences still parse reliably
  - Still not horrible: 74% vs 84%
    - Much better than LSA model (52.5%)
- Learning curve shows 80-100 pairs good enough

State-of-the-Art Comparisons

- Two comparison systems:
  - Latent Semantic Analysis (LSA)
  - Barzilay & Lee (2004)

Comparison I

- LSA model:
  - Motivation: Lexical gaps

Comparison

- LSA model:
  - Motivation: Lexical gaps
    - Pure surface word match misses similarity

Comparison

- LSA model:
  - Motivation: Lexical gaps
    - Pure surface word match misses similarity
    - Discover underlying concept representation
      - Based on distributional patterns

Comparison

- LSA model:
  - Motivation: Lexical gaps
    - Pure surface word match misses similarity
    - Discover underlying concept representation
      - Based on distributional patterns

  - Create term x document matrix over large news corpus

Comparison

- LSA model:
  - Motivation: Lexical gaps
    - Pure surface word match misses similarity
    - Discover underlying concept representation
      - Based on distributional patterns

  - Create term x document matrix over large news corpus
  - Perform SVD to create 100-dimensional dense matrix

Comparison

- LSA model:
  - Motivation: Lexical gaps
    - Pure surface word match misses similarity
    - Discover underlying concept representation
      - Based on distributional patterns
  - Create term x document matrix over large news corpus
  - Perform SVD to create 100-dimensional dense matrix
- Score summary as:
  - Sentence represented as mean of its word vectors
  - Average of cosine similarity scores of adjacent sents
    - Local “concept” similarity score

“Catching the Drift”

- Barzilay and Lee, 2004 (NAACL best paper)

- Intuition:
  - Stories:
    - Composed of topics/subtopics
    - Unfold in systematic sequential way
    - Can represent ordering as sequence modeling over topics

"Catching the Drift"

- Barzilay and Lee, 2004 (NAACL best paper)

- Intuition:
  - Stories:
    - Composed of topics/subtopics
    - Unfold in systematic sequential way
    - Can represent ordering as sequence modeling over topics
- Approach: HMM over topics

Strategy

- Lightly supervised approach:
  - Learn topics in unsupervised way from data
    - Assign sentences to topics

Strategy

- Lightly supervised approach:
  - Learn topics in unsupervised way from data
    - Assign sentences to topics
  - Learn sequences from document structure
    - Given clusters, learn sequence model over them

Strategy

- Lightly supervised approach:
  - Learn topics in unsupervised way from data
    - Assign sentences to topics
  - Learn sequences from document structure
    - Given clusters, learn sequence model over them
  - No explicit topic labeling, no hand-labeling of sequence

Topic Induction

- How can we induce a set of topics from doc set?
  - Assume we have multiple documents in a domain

Topic Induction

- How can we induce a set of topics from doc set?
  - Assume we have multiple documents in a domain
- Unsupervised approach:?

Topic Induction

- How can we induce a set of topics from doc set?
  - Assume we have multiple documents in a domain

- Unsupervised approach:? Clustering
  - Similarity measure?

Topic Induction

- How can we induce a set of topics from doc set?
  - Assume we have multiple documents in a domain
- Unsupervised approach:? Clustering
  - Similarity measure?
    - Cosine similarity over word bigrams
  - Assume some irrelevant/off-topic sentences
    - Merge clusters with few members into “etcetera” cluster

Topic Induction

- How can we induce a set of topics from doc set?
  - Assume we have multiple documents in a domain
- Unsupervised approach:? Clustering
  - Similarity measure?
    - Cosine similarity over word bigrams
  - Assume some irrelevant/off-topic sentences
    - Merge clusters with few members into "etcetera" cluster
- Result: *m* topics, defined by clusters

Sequence Modeling

- Hidden Markov Model
  - States

Sequence Modeling

- Hidden Markov Model
  - States = Topics
    - State m: special insertion state
  - Transition probabilities:
    - Evidence for ordering?

Sequence Modeling

- Hidden Markov Model
  - States = Topics
    - State m: special insertion state
  - Transition probabilities:
    - Evidence for ordering?
      - Document ordering
        - Sentence from topic *a* appears before sentence from topic *b*

Sequence Modeling

- Hidden Markov Model
  - States = Topics
    - State m: special insertion state
  - Transition probabilities:
    - Evidence for ordering?
      - Document ordering
        - Sentence from topic *a* appears before sentence from topic *b*

$$p(s_j \mid s_i) = \frac{D(c_i, c_j) + \delta_2}{D(c_i) + \delta_2 m}$$

Sequence Modeling II

- Emission probabilities:
  - Standard topic state:
    - Probability of observation given state (topic)

Sequence Modeling II

- Emission probabilities:
  - Standard topic state:
    - Probability of observation given state (topic)
      - Probability of sentence under topic-specific bigram LM
      - Bigram probabilities

Sequence Modeling II

- Emission probabilities:
  - Standard topic state:
    - Probability of observation given state (topic)
      - Probability of sentence under topic-specific bigram LM
      - Bigram probabilities

$$p_{s_i}(w' \mid w) = \frac{f_{c_i}(ww') + \delta_1}{f_{c_i}(w) + |V|}$$

Sequence Modeling II

- Emission probabilities:
  - Standard topic state:
    - Probability of observation given state (topic)
      - Probability of sentence under topic-specific bigram LM
      - Bigram probabilities

$$p_{s_i}(w'|w) = \frac{f_{c_i}(ww') + \delta_1}{f_{c_i}(w) + |V|}$$

  - Etcetera state:
      - Forced complementary to other states

$$p_{s_m} = \frac{1 - \max_{i:i<m} p_{s_i}(w'|w)}{\sum_{u \in V}(1 - \max_{i:i<m} p_{s_i}(u|w))}$$

Sequence Modeling III

- Viterbi re-estimation:
  - Intuition: Refine clusters, etc based on sequence info

Sequence Modeling III

- Viterbi re-estimation:
  - Intuition: Refine clusters, etc based on sequence info
  - Iterate:
    - Run Viterbi decoding over original documents
    - Assign each sentence to cluster most likely to generate it
    - Use new clustering to recompute transition/emission

Sequence Modeling III

- Viterbi re-estimation:
  - Intuition: Refine clusters, etc based on sequence info
  - Iterate:
    - Run Viterbi decoding over original documents
    - Assign each sentence to cluster most likely to generate it
    - Use new clustering to recompute transition/emission
  - Until stable (or fixed iterations)

Sentence Ordering Comparison

- Restricted domain text:
  - Separate collections of earthquake, aviation accidents
  - LSA predictions:

Sentence Ordering Comparison

- Restricted domain text:
  - Separate collections of earthquake, aviation accidents
  - LSA predictions: which order has higher score
  - Topic/content model:

Sentence Ordering Comparison

- Restricted domain text:
  - Separate collections of earthquake, aviation accidents
  - LSA predictions: which order has higher score
  - Topic/content model: highest probability under HMM

Model	Earthquakes	Accidents
Coreference+Syntax+Salience+	87.2	90.4
Coreference+Syntax+Salience−	88.3	90.1
Coreference+Syntax−Salience+	86.6	88.4**
Coreference−Syntax+Salience+	83.0**	89.9
Coreference+Syntax−Salience−	86.1	89.2
Coreference−Syntax+Salience−	82.3**	88.6*
Coreference−Syntax−Salience+	83.0**	86.5**
Coreference−Syntax−Salience−	81.4**	86.0**
HMM-based Content Models	88.0	75.8**
Latent Semantic Analysis	81.0**	87.3**

Summary Coherence Scoring Comparison

- Domain independent:
  - Too little data per domain to estimate topic-content model
- Train: 144 pairwise summary rankings
- Test: 80 pairwise summary rankings

Summary Coherence Scoring Comparison

- Domain independent:
    - Too little data per domain to estimate topic-content model
  - Train: 144 pairwise summary rankings
  - Test: 80 pairwise summary rankings
  - Entity grid model (best): 83.8%
  - LSA model: 52.5%
- Likely issue:

Summary Coherence Scoring Comparison

- Domain independent:
  - Too little data per domain to estimate topic-content model
  - Train: 144 pairwise summary rankings
  - Test: 80 pairwise summary rankings

  - Entity grid model (best): 83.8%

  - LSA model: 52.5%

- Likely issue:
  - Bad auto summaries highly repetitive ➔

Summary Coherence Scoring Comparison

- Domain independent:
  - Too little data per domain to estimate topic-content model
- Train: 144 pairwise summary rankings
- Test: 80 pairwise summary rankings

- Entity grid model (best): 83.8%

- LSA model: 52.5%

- Likely issue:
- Bad auto summaries highly repetitive →
  - High inter-sentence similarity

Ordering as Optimization

- Given a set of sentences to order
- Define a local pairwise coherence score b/t sentences
- Compute a total order optimizing local distances
- Can we do this efficiently?

Ordering as Optimization

- Given a set of sentences to order
- Define a local pairwise coherence score b/t sentences
- Compute a total order optimizing local distances
- Can we do this efficiently?
  - Optimal ordering of this type is equivalent to TSP
    - Traveling Salesperson Problem: Given a list of cities and distances between cities, find the shortest route that visits each city exactly once and returns to the origin city.

Ordering as Optimization

- Given a set of sentences to order
- Define a local pairwise coherence score b/t sentences
- Compute a total order optimizing local distances
- Can we do this efficiently?
  - Optimal ordering of this type is equivalent to TSP
    - Traveling Salesperson Problem: Given a list of cities and distances between cities, find the shortest route that visits each city exactly once and returns to the origin city.
    - TSP is NP-complete (NP-hard)

Ordering as TSP

- Can we do this practically?
  - Summaries are 100 words, so 6-10 sentences
    - 10 sentences have how many possible orders

Ordering as TSP

- Can we do this practically?
  - Summaries are 100 words, so 6-10 sentences
    - 10 sentences have how many possible orders? O(n!)
    - Not impossible
  - Alternatively,

Ordering as TSP

- Can we do this practically?
  - Summaries are 100 words, so 6-10 sentences
    - 10 sentences have how many possible orders? O(n!)
    - Not impossible
  - Alternatively,
    - Use an approximation methods
    - Take the best of a sample

CLASSY 2006

- Formulates ordering as TSP
- Requires pairwise sentence distance measure

CLASSY 2006

- Formulates ordering as TSP
- Requires pairwise sentence distance measure
  - Term-based similarity: # of overlapping terms

CLASSY 2006

- Formulates ordering as TSP
- Requires pairwise sentence distance measure
  - Term-based similarity: # of overlapping terms
  - Document similarity:
    - Multiply by a weight if in the same document (there, 1.6)

CLASSY 2006

- Formulates ordering as TSP
- Requires pairwise sentence distance measure
  - Term-based similarity: # of overlapping terms
  - Document similarity:
    - Multiply by a weight if in the same document (there, 1.6)
  - Normalize to between 0 and 1 (sqrt of product of selfsim)
    - Make distance: subtract from 1

Practicalities of Ordering

- Brute force: $O(n!)$

Practicalities of Ordering

- Brute force: O(n!)
  - “there are **only** 3,628,800 ways to order 10 sentences plus a lead sentence, so exhaustive search is feasible.“ (Conroy)

Practicalities of Ordering

- Brute force: O(n!)
  - "there are **only** 3,628,800 ways to order 10 sentences plus a lead sentence, so exhaustive search is feasible." (Conroy)
- Still,..
  - Used sample set to pick best
    - Candidates:
      - Random
      - Single-swap changes from good candidates

Practicalities of Ordering

- Brute force: O(n!)
  - “there are **only** 3,628,800 ways to order 10 sentences plus a lead sentence, so exhaustive search is feasible.“ (Conroy)

- Still,..
  - Used sample set to pick best
    - Candidates:
      - Random
      - Single-swap changes from good candidates
  - 50K enough to consistently generate minimum cost order

Conclusions

- Many cues to ordering:
  - Temporal, coherence, cohesion
    - Chronology, topic structure, entity transitions, similarity
- Strategies:
  - Heuristic, machine learned; supervised, unsupervised
  - Incremental build-up versus generate & rank
- Issues:
  - Domain independence, semantic similarity, reference