Information Ordering

Ling 573
Systems and Applications
May 5, 2015

Roadmap

- Ordering models:
  - Chronology and topic structure

  - Mixture of experts
    - Preference ranking:
      - Chronology, topic similarity, succession/precedence

  - Entity-based cohesion
    - Entity transitions
    - Coreference, syntax, and salience

Framework

- Build on existing Multigen system
- Motivated by issues of similarity and difference
  - Managing redundancy and contradiction in docs
- Analysis groups sentences into “themes”
  - Text units from diff’t docs with repeated information
  - Roughly clusters of sentences with similar content
  - Intersection of their information is summarized
- Ordering is done on this selected content

Chronological Orderings I

- Two basic strategies explored:
  - CO:
    - Need to assign dates to **themes** for ordering

Chronological Orderings I

- Two basic strategies explored:
  - CO:
    - Need to assign dates to **themes** for ordering
      - Theme sentences from multiple docs, lots of dup content
    - Temporal relation extraction

Chronological Orderings I

- Two basic strategies explored:
  - CO:
    - Need to assign dates to themes for ordering
      - Theme sentences from multiple docs, lots of dup content
    - Temporal relation extraction is hard, try simple sub.
      - Doc publication date: what about duplicates?

Chronological Orderings I

- Two basic strategies explored:
  - CO:
    - Need to assign dates to t**hemes** for ordering
      - Theme sentences from multiple docs, lots of dup content
    - Temporal relation extraction is hard, try simple sub.
      - Doc publication date: what about duplicates?
    - **Theme** date: earliest pub date for theme sentence
  - Order **themes** by date
  - If different themes have same date?

Chronological Orderings I

- Two basic strategies explored:
  - CO:
    - Need to assign dates to **themes** for ordering
      - Theme sentences from multiple docs, lots of dup content
    - Temporal relation extraction is hard, try simple sub.
      - Doc publication date: what about duplicates?
    - **Theme** date: earliest pub date for theme sentence
  - Order **themes** by date
  - If different themes have same date?
    - Same article, so use article order
- Slightly more sophisticated than simplest model

Chronological Orderings II

- MO (Majority Ordering):
  - Alternative approach to ordering themes
    - Order the whole themes relative to each other
      - i.e. Th1 precedes Th2
    - How?

Chronological Orderings II

- MO (Majority Ordering):
  - Alternative approach ordering themes
    - Order the whole themes relative to each other
      - i.e. Th1 precedes Th2
    - How? If all sentences in Th1 before all sentences in Th2?

Chronological Orderings II

- MO (Majority Ordering):
  - Alternative approach ordering themes
    - Order the whole themes relative to each other
      - i.e. Th1 precedes Th2
    - How? If all sentences in Th1 before all sentences in Th2?
      - Easy: Th1 b/f Th2
      - If not?

Chronological Orderings II

- MO (Majority Ordering):
  - Alternative approach ordering themes
    - Order the whole themes relative to each other
      - i.e. Th1 precedes Th2
    - How? If all sentences in Th1 before all sentences in Th2?
      - Easy: Th1 b/f Th2
      - If not? Majority rule
        - Problematic b/c not guaranteed transitive
    - Create an ordering by modified topological sort over graph

Chronological Orderings II

- MO (Majority Ordering):
  - Alternative approach ordering themes
    - Order the whole themes relative to each other
      - i.e. Th1 precedes Th2
    - How? If all sentences in Th1 before all sentences in Th2?
      - Easy: Th1 b/f Th2
      - If not? Majority rule
        - Problematic b/c not guaranteed transitive
    - Create an ordering by modified topological sort over graph
      - Nodes are themes:
        - Weight: sum of outgoing edges minus sum of incoming edges
      - Edges E(x,y): precedence, weighted by # texts
        - where sentences in x precede those in y

Chronological Orderings II

- MO (Majority Ordering):
  - Alternative approach ordering themes
    - Order the whole themes relative to each other
      - i.e. Th1 precedes Th2
    - How? If all sentences in Th1 before all sentences in Th2?
      - Easy: Th1 b/f Th2
      - If not? Majority rule
        - Problematic b/c not guaranteed transitive
    - Create an ordering by modified topological sort over graph
      - Nodes are themes:
        - Weight: sum of outgoing edges minus sum of incoming edges
      - Edges E(x,y): precedence, weighted by # texts
        - where sentences in x precede those in y

CO vs MO

	Poor	Fair	Good
MO	3	14	8
CO	10	8	7

CO vs MO

- Neither of these is particularly good:

	Poor	Fair	Good
MO	3	14	8
CO	10	8	7

- MO works when presentation order consistent
  - When inconsistent, produces own brand new order

CO vs MO

- Neither of these is particularly good:

	Poor	Fair	Good
MO	3	14	8
CO	10	8	7

- MO works when presentation order consistent
  - When inconsistent, produces own brand new order
- CO problematic on:
  - Themes that aren't tied to document order
    - E.g. quotes about reactions to events
  - Multiple topics not constrained by chronology

New Approach

- Experiments on sentence ordering by subjects
  - Many possible orderings but far from random
    - Blocks of sentences group together (cohere)

New Approach

- Experiments on sentence ordering by subjects
  - Many possible orderings but far from random
    - Blocks of sentences group together (cohere)
- Combine chronology with cohesion
  - Order chronologically, but group similar themes

New Approach

- Experiments on sentence ordering by subjects
  - Many possible orderings but far from random
    - Blocks of sentences group together (cohere)
- Combine chronology with cohesion
  - Order chronologically, but group similar themes
- Perform topic segmentation on original texts
- Themes “related” if,

New Approach

- Experiments on sentence ordering by subjects
  - Many possible orderings but far from random
    - Blocks of sentences group together (cohere)
- Combine chronology with cohesion
  - Order chronologically, but group similar themes
- Perform topic segmentation on original texts
- Themes "related" if, when two themes appear in same text, they frequently appear in same segment (threshold)

New Approach

- Experiments on sentence ordering by subjects
  - Many possible orderings but far from random
    - Blocks of sentences group together (cohere)
- Combine chronology with cohesion
  - Order chronologically, but group similar themes
- Perform topic segmentation on original texts
- Themes "related" if, when two themes appear in same text, they frequently appear in same segment (threshold)
- Order over groups of themes by CO,
  - Then order within groups by CO
- Significantly better!

Before and After

Thousands of people have attended a ceremony in Nairobi commemorating the first anniversary of the deadly bombings attacks against U.S. Embassies in Kenya and Tanzania.
Saudi dissident Osama bin Laden, accused of masterminding the attacks, and nine others are still at large.
President Clinton said, "The intended victims of this vicious crime stood for everything that is right about our country and the world".
U.S. federal prosecutors have charged 17 people in the bombings.
Albright said that the mourning continues.
Kenyans are observing a national day of mourning in honor of the 215 people who died there.

Deliverable #3

- Requirements:
  - Information ordering:
    - Do something non-stub for information ordering
  - Improve content selection component:
    - Incorporate some topic-orientation
    - Build on what you've learned in D#2
      - Alternative, more sophisticated strategies
- Code due May 15, report 18th

Integrating Ordering Preferences

- Learning Ordering Preferences
  - (Bollegala et al, 2012)
- Key idea:
  - Information ordering involves multiple influences
    - Can be viewed as soft preferences
  - Combine via multiple experts:
    - Chronology
    - Sequence probability
    - Topicality
    - Precedence/Succession

Basic Framework

- Combination of experts
- Build one expert for each of diff't preferences
  - Take a pair of sentences (a,b) and partial summary
    - Score > 0.5 if prefer a before b
    - Score < 0.5 if prefer b before a
- Learn weights for linear combination
- Use greedy algorithm to produce final order

Chronology Expert

- Implements the simple chronology model
  - If sentences from two different docs w/diff't times
    - Order by document timestamp
  - If sentences from same document
    - Order by document order
  - Otherwise, no preference

Topicality Expert

- Same motivation as Barzilay 2002
- Example:
  - The earthquake crushed cars, damaged hundreds of houses, and terrified people for hundreds of kilometers around.
  - A major earthquake measuring 7.7 on the Richter scale rocked north Chile Wednesday.
  - Authorities said two women, one aged 88 and the other 54, died when they were crushed under the collapsing walls.
- 2 > 1 > 3

Topicality Expert

- Idea: Prefer sentence about the “current” topic
- Implementation:?
  - Prefer sentence with highest similarity to sentence in summary so far
  - Similarity computation:?
    - Cosine similarity b/t current & summary sentence
    - Stopwords removed; nouns, verbs lemmatized; binary

Precedence/Succession Experts

- Idea: Does current sentence look like blocks preceding/ following current summary sentences in their original documents?
- Implementation:
  - For each summary sentence, compute similarity of current sentence w/most similar pre/post in original doc
    - Similarity?: cosine
- $PREF_{pre}(u,v,Q)$= 0.5 if [Q=v] or [pre(u)=pre(v)]
- 1.0 if [Q!=null] and [pre(u)>pre(v)]
- 0 otherwise
      - Symmetrically for post

Sketch

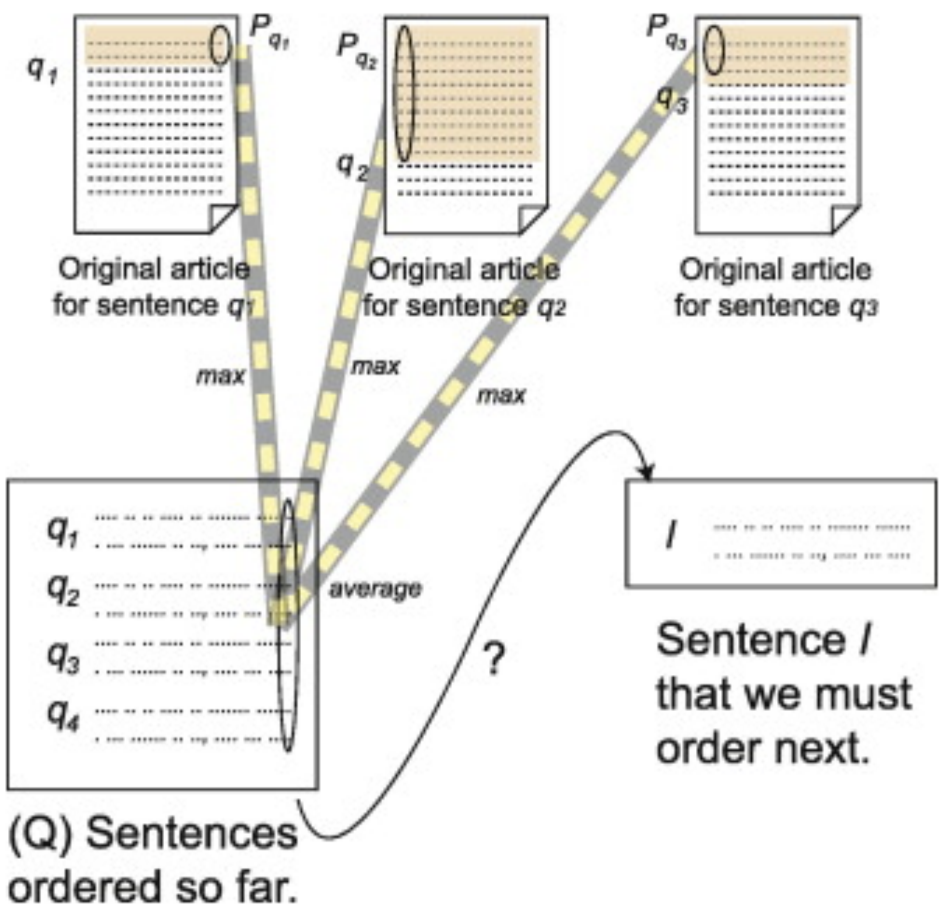

Fig. 4.

Precedence expert.

Probabilistic Sequence

- Intuition:
  - Probability of summary is the probability of sequence of sentences in it, assumed Markov
  - P(summary)= Π $P(S_i | S_{I-1})$
- Issue:
  - Sparsity: will we actually see identical pairs in training?
- Repeatedly backoff:
  - To N, V pairs in ordered sentences
  - To backoff smoothing + Katz

Results & Weights

- Trained weighting using a boosting method
- Combined:
  - Learning approach significantly outperforms random, prob
  - Somewhat better that raw chronology

Expert	Weight
Succession	0.44
Chronology	0.33
Precedence	0.20
Topic	0.016
Prob. Seq.	0.00004

Observations

- Nice ideas:
  - Combining multiple sources of ordering preference
  - Weight-based integration
- Issues:
  - Sparseness everywhere
    - Ubiquitous word-level cosine similarity
    - Probabilistic models
  - Score handling

Entity-Centric Cohesion

- Continuing to talk about same thing(s) lends cohesion to discourse
- Incorporated variously in discourse models
  - Lexical chains: Link mentions across sentences
    - Fewer lexical chains crossing → shift in topic
  - Salience hierarchies, information structure
    - Subject > Object > Indirect > Oblique > ....
  - Centering model of coreference
    - Combines grammatical role preference with
    - Preference for types of reference/focus transitions

Entity-Based Ordering

- Idea:
  - Leverage patterns of entity (re)mentions
- Intuition:
  - Captures local relations b/t sentences, entities
  - Models cohesion of evolving story
- Pros:
  - Largely delexicalized
    - Less sensitive to domain/topic than other models
  - Can exploit state-of-the-art syntax, coreference tools

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
- 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: # 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:
  - 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*

Comparison

- LSA model:
  - 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

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