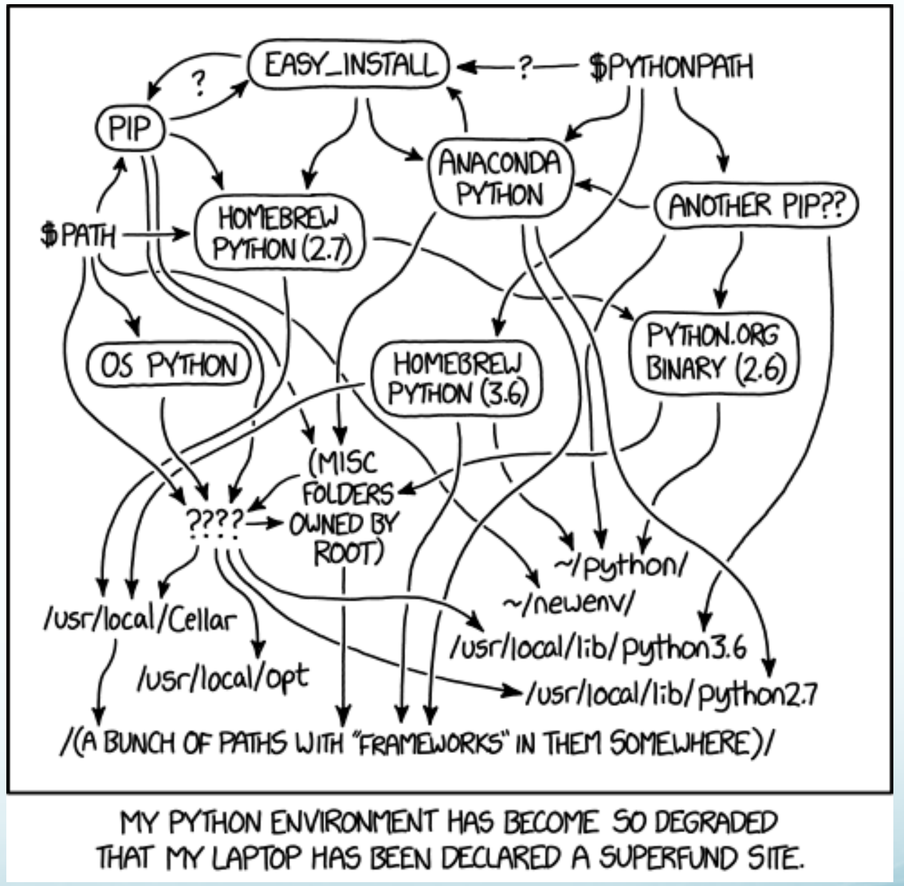
Information Ordering

LING 573 — NLP Systems & Applications
4/30/2018

Using External Python Packages

- If you want to use additional python packages on patas:
  - **`python3 -m venv $SOME_PATH`**
  - create a **`requirements.txt`** file for the packages you use in your root dir
  - use **`$SOME_PATH/bin/python3`** to run code
    - …you can also add **`$SOME_PATH/bin`** to your **`$PATH`** variable



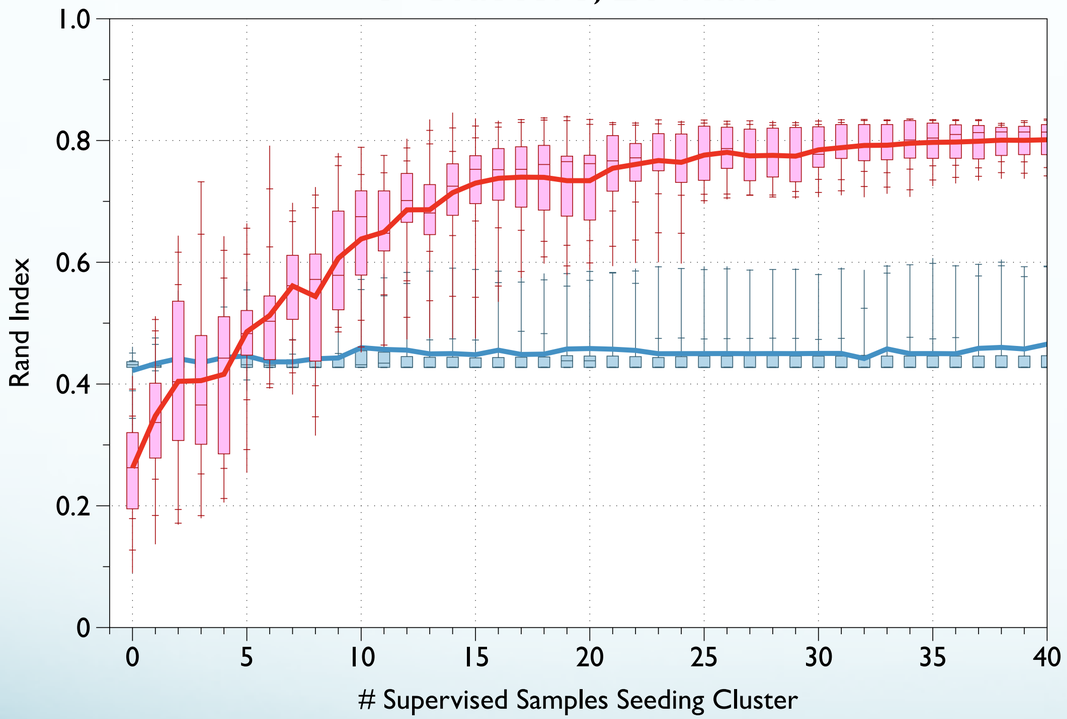
MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

Using External Python Packages

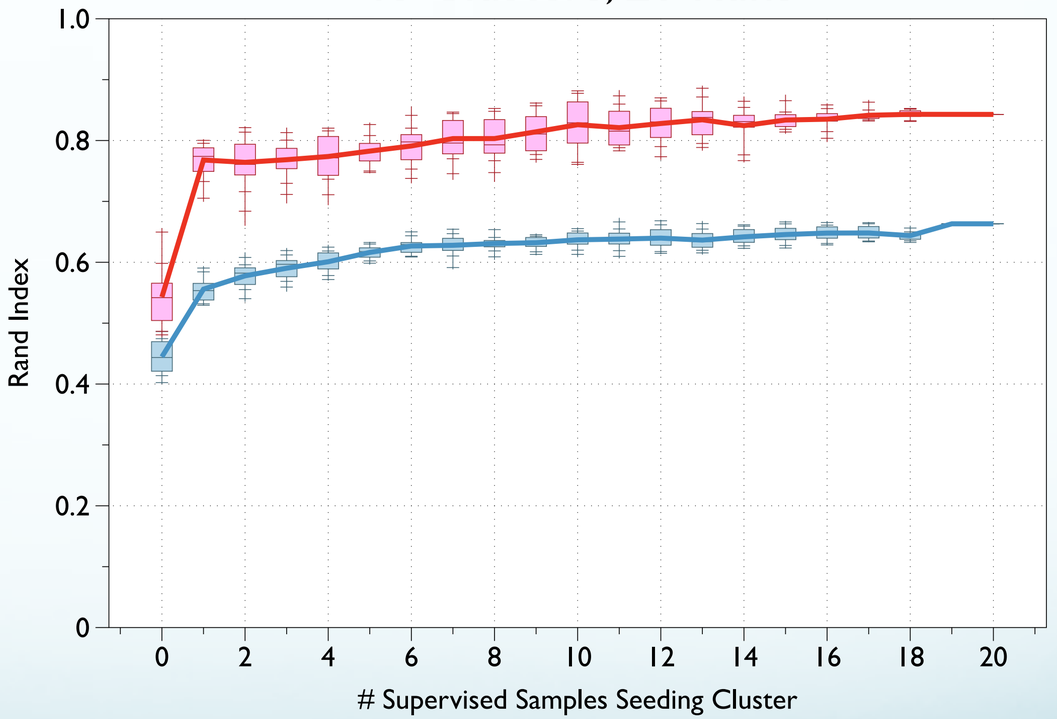
- This will allow:
  - local installation of packages from PyPy
  - Other user to quickly install dependencies:
    - **`pip install -r requirements.txt`**
- Standard way of supporting dependencies in python packaging

Semi-Supervised Document Clustering

KMeans Clustering on TAC Categories
5 Clusters, 20 Runs



KMeans Clustering on TAC Topics
46 Clusters, 20 Runs



= tf•idf via sklearn = GloVe via spacy

Semi-Supervised Document Clustering

- Code on github:
  - github.com/rgeorgi/tac-clusters

Information Ordering: Combining Experts

Integrating Ordering Preferences

Bollegala et al, 2012

- Key idea:
  - Information ordering is a combination of factors.
  - Consider multiple "experts" that model different factors
  - Combine in a linear combination to determine ordering

Training "Experts"

1. Chronological Expert
2. Probabilistic Expert
3. Topical-closeness Expert
4. Precedence Expert
5. Succession Expert

Basic Framework

- Build one expert for each preference
  - Iterate through pairs of sentences (u, v) and partial summary Q
    - prefer u before v if score > 0.5
    - prefer v before u if score < 0.5
- Learn weights for linear combination
- Use greedy algorithm to produce final order

$$PREF_{total}(u,v,Q) = \sum_{e \in E} w_e PREF_e(u,v,Q)$$

Chronological Expert

- If sentences from two different documents with different times
  - Order by document timestamp
- If sentences from same document
  - Order by order within document
- Otherwise, no preference

$$PREF_{chro}(u,v,Q) = \begin{cases} 1 & T(u) < T(v) \\ 1 & \left[D(u) = D(v)\right] \wedge \left[N(u) < N(v)\right] \\ 0.5 & \left[T(u) = T(v)\right] \wedge \left[D(u) \neq D(v)\right] \\ 0 & otherwise \end{cases}$$

Probabilistic Expert

- Based on Lapata (2003)
- Model the probability of ***u*** preceding ***v*** in summary — features:
  - POS tags
  - Dependency Structures
  - Lemmas (Smoothed)
- 0.5 is returned for equally likely outcomes, ***u*** preferred if > 0.5

$$PREF_{prob}(u,v) = \frac{1 + P(v \mid u) - P(u \mid v)}{2}$$

Topical-closeness Expert

- Same motivation as Barzilay, 2002 (Clustering sentences into "themes")
  1. *The earthquake crushed cars, damaged hundreds of houses, and terrified people for hundreds of kilometers around.*
  2. *A major earthquake measuring 7.7 on the Richter scale rocked north Chile Wednesday*
  3. *Authorities said two women, one aged 88 and the other 54, died when they were crushed under the collapsing walls*
- **1** and **3** discuss theme of **impact**
- **2** describes magnitude and **location**
- Better order — (**2**), (**1**, **3**)

Topical-closeness Expert

$$topic(l) = \max_{q \in Q} sim(l, q)$$

- Q = sentences ordered thus far, $q \in Q$
- Look at candidate sentences u, v
  - Pick one with closest similarity to already ordered sentence q
  - 0.5 if similarity is identical

$$PREF_{topic}(u, v, Q) = \begin{cases} 0.5 & [Q = \varnothing] \vee [topic(u) = topic(v)] \\ 1 & [Q \neq \varnothing] \vee [topic(u) > topic(v)] \\ 0 & otherwise \end{cases}$$

Topical-closeness Expert

$$topic(l) = \max_{q \in Q} sim(l, q)$$

- Bollegala et al use cosine similarity
  - Could use any similarity measure, suggest WordNet as an alternative

Precedence Expert

- With following example:

(a) Honduran death estimates grew from 32 to 231 in the first two days, to 6,076 with 4,621 missing.

(b) Honduras braced as category 5 Hurricane Mitch approached.

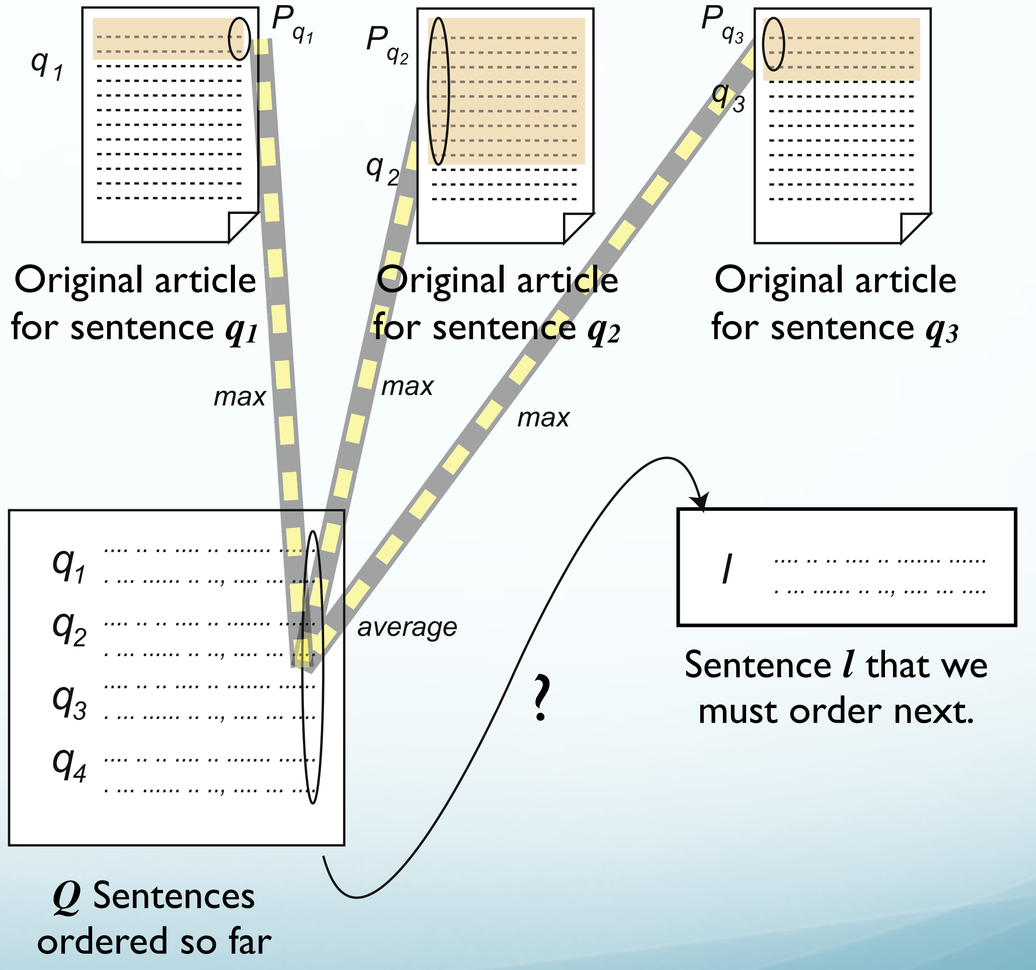
(c) The EU approved 6.4 million in aid to Mitch's victims.

- *(b)* introduces event that is needed to understand *(a)*, *(c)*
  - *(a)* and *(c)* contain information preceded by *(b)*

Precedence Expert

$$pre(l) = \frac{1}{|Q|} \sum_{q \in Q} \max_{p \in P_q} sim(p, l)$$

- For candidate sentence l:
  - For every already ordered sentence q:
    - Find max similarity of any sentence p preceding q in q's original document to l
  - Average this for all q.
- **Idea:**
  - Sentences with maximum similarity to sentences preceding those in Q should (similarly) come first.



Precedence Expert

$$PREF_{pre}(u,v,Q) = \begin{cases} 0.5 & [Q = \varnothing] \vee [pre(u) = pre(v)] \\ 1 & [Q \neq \varnothing] \wedge [pre(u) > pre(v)] \\ 0 & otherwise \end{cases}$$

- No preference if there are no sentences already in Q
- If precedence of ***u*** more than ***v***, prefer ***u***
- Otherwise prefer ***v***

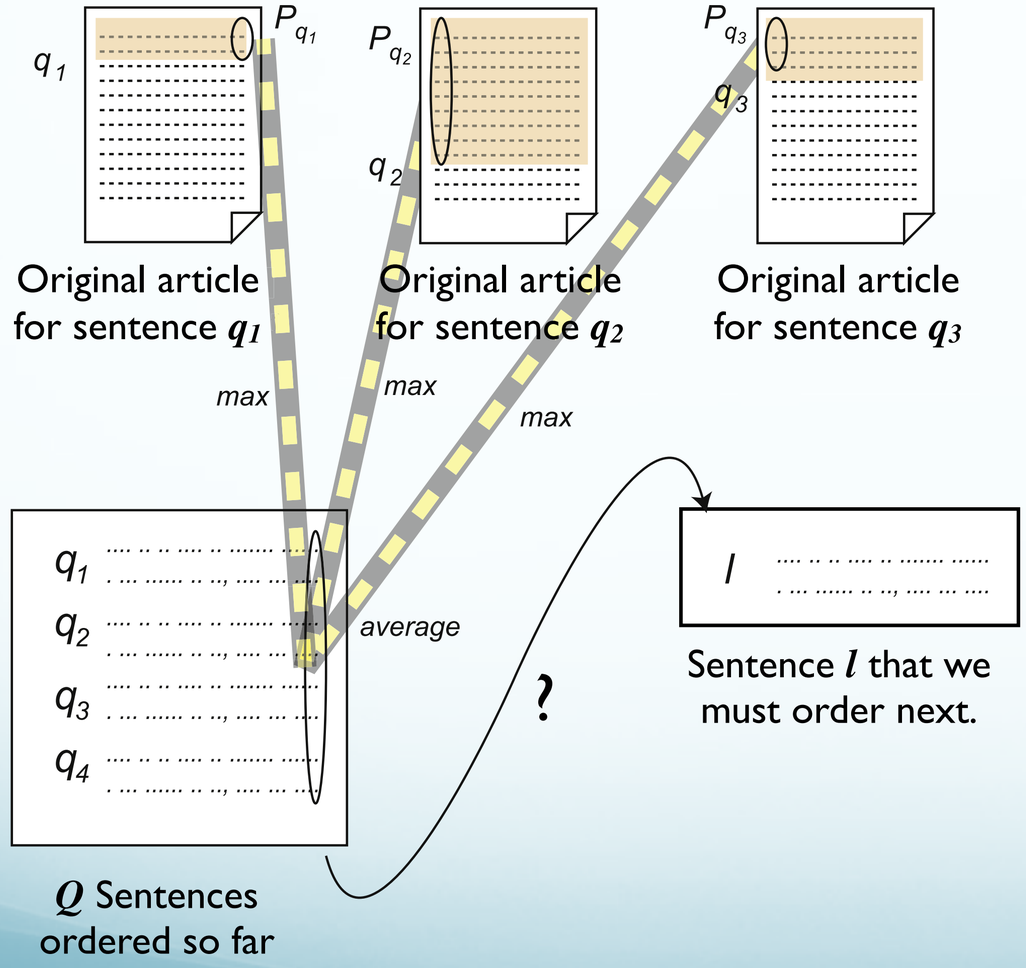
Succession Expert

$$succ(l) = \frac{1}{|Q|}\sum_{q \in Q} \max_{s \in S_q} sim(s, l)$$

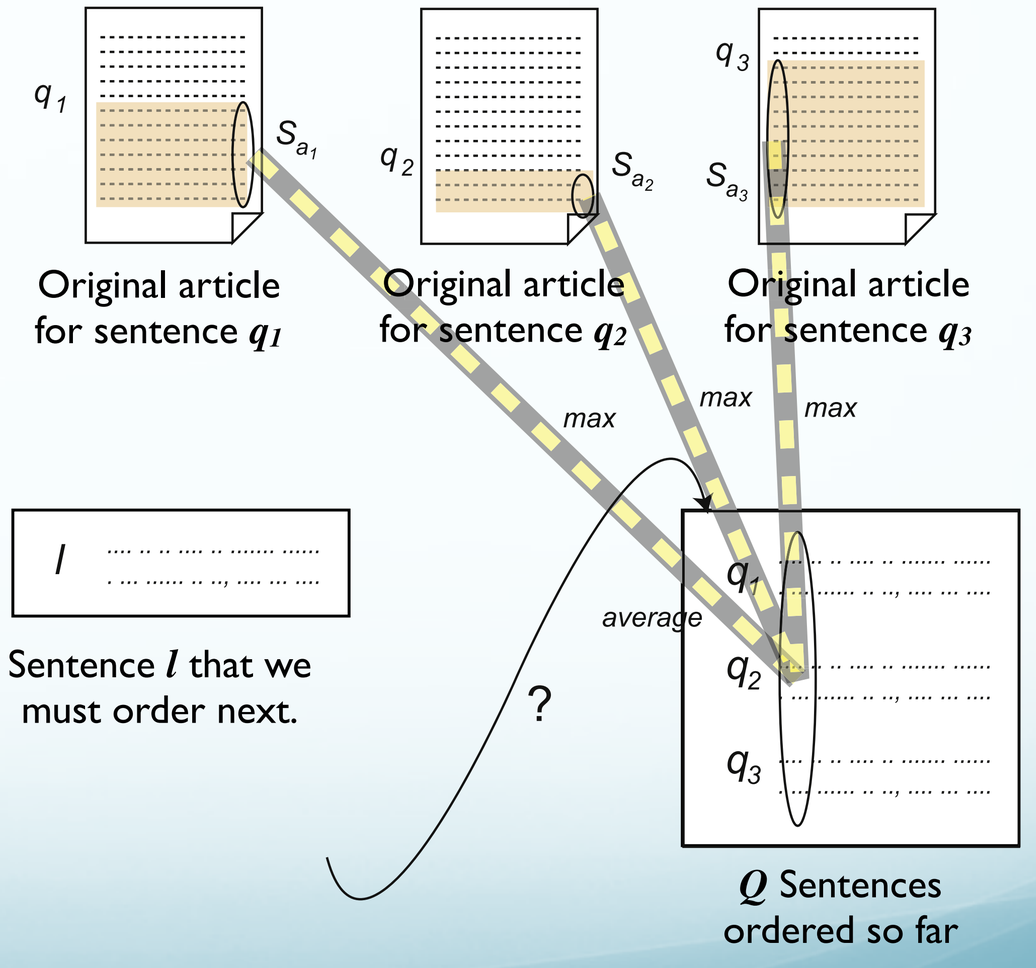
- Inverse of precedence
  - Calculate similarity of candidate with information that succeeds Q in original docs

$$PREF_{succ}(u, v, Q) = \begin{cases} 0.5 & [Q = \varnothing] \vee [succ(u) = succ(v)] \\ 1 & [Q \neq \varnothing] \wedge [succ(u) > succ(v)] \\ 0 & otherwise \end{cases}$$

Precedence Expert

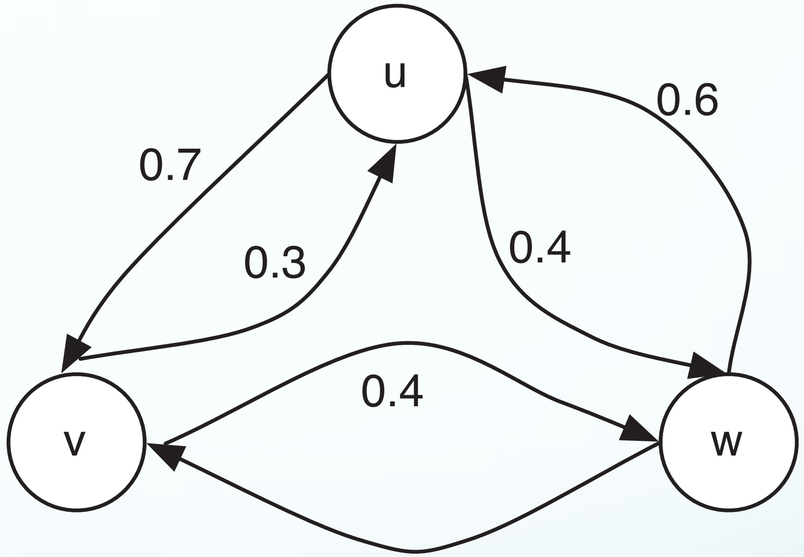


Succession Expert



Learning Algorithm

- Use the same algorithm to find optimal weights as Barzilay (2002)
  - Namely, Cohen et. al (1999)
- Use model summaries to train
  - Learn optimal weights for experts given training data



Learned Weights

- Optimal learned weights:

Expert	Weight
Chronological	0.327947
Probabilistic	0.000039
Topical-closeness	0.016287
Precedent	0.196562
Succession	0.444102

Results

- Correlated scores of various approaches against human judgments

Method		Spearman
Random Ordering	(RO)	-0.267
Probabilistic Ordering	(PO)	0.062
Chronological Ordering	(CO)	0.774
Proposed Method	(LO)	0.783

- Probabilistic ordering is rubbish
- Chronological actually does pretty well
- Combined model with learned weights better than Chronological alone

Observations

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

Entity-Based Ordering

Entity-Based Ordering

Barzilay & Lapata (2005, 2008)

- Continuing to talk about same thing(s) lends cohesion to discourse
- Incorporated variously in discourse models
  - Lexical chains: Link mentions across sentences
    - Fewer lexical chain crossings → fewer shifts in topics
  - Salience hierarchies, information structure
    - Subject > Object > Indirect > Oblique …
  - Centering model of Coreference
    - Combine grammatical role preference with
    - Preference for types of references/focus transitions

Entity-Based Ordering

Barzilay & Lapata (2005, 2008)

- Idea:
  - Leverage patterns of entity (re)mentions
- Intuition:
  - Capture local relations between sentences, entities
  - Model cohesion of evolving story

Entity-Based Ordering

Barzilay & Lapata (2005, 2008)

- Pros:
  - Largely delexicalized
    - Less sensitive to domain/topic than other models
  - Can exploit state-of-the-art syntax, coreference tools

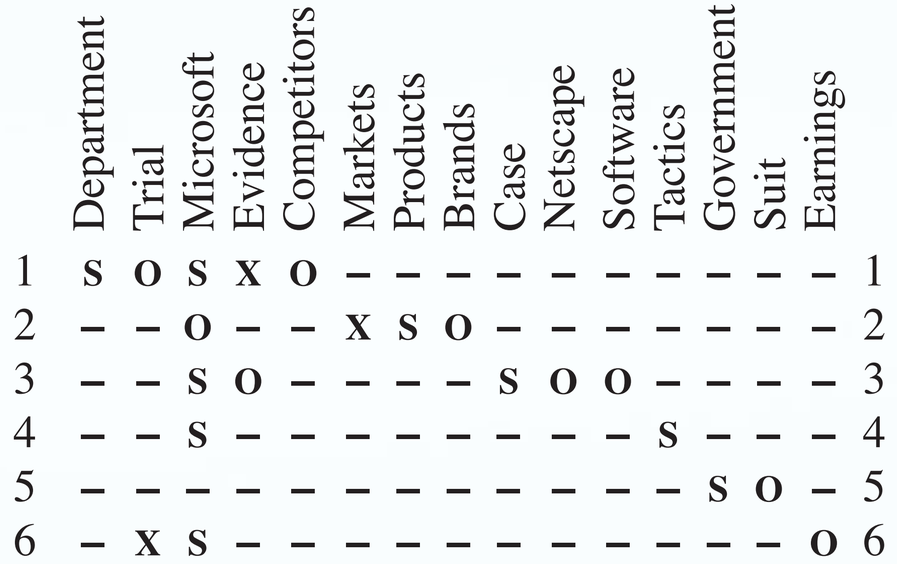
Entity Grid

- Compact representation across sentences of:
  - Mentions
  - Grammatical Roles
  - Transitions

The Entity Grid

- Rows = sentences
- Columns = discourse entities
- Values = grammatical role of mention in sentence
  - (S)ubject
  - (O)bject
  - X (other)
  - _ (no mention)
  - Multiple mentions — Take highest grammatical ranking (S > O > X)

	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**.

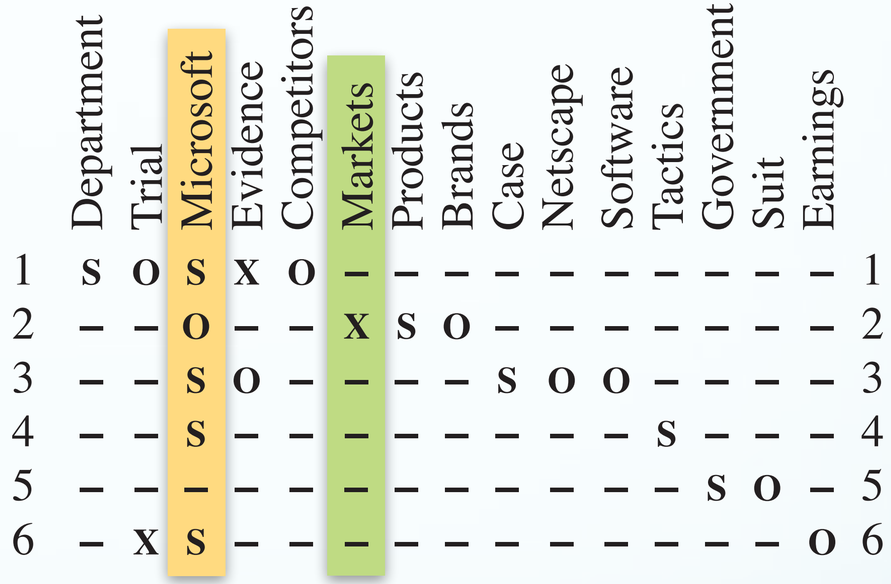
4 [Microsoft]**s** claims [its tactics]**s** are commonplace and good economically

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. Microsoft)
    - 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:
    - # of occurrences of that type/# of occurrences of that length



Vector Representation

	S S	S O	S X	S I	O S	O O	O X	O I	X S	X O	X X	X I	I S	I O	I X	I I
d_1	0	0	0	.03	0	0	0	.02	.07	0	0	.12	.02	.02	.05	.25
d_2	0	0	0	.02	0	.07	0	.02	0	0	.06	.04	0	0	0	.36
d_3	.02	0	0	.03	0	0	0	.06	0	0	0	.05	.03	.07	.07	.29

- Document vector:
  - Length = # of transition types
  - Values = Probabilities of each transition type
- Can vary by transition types
  - e.g. most frequent; all transition of some length, etc.