Discourse and Summarization

LING 573 — Systems and Applications
April 12th, 2018

UNIVERSITY OF WASHINGTON

PROFESSIONAL MASTER'S IN COMPUTATIONAL LINGUISTICS

Begin Recording!

Miscellanea

What is a Centroid?

- Way to define the "middle" of a cluster
- In document clustering setting, centroid often:
  - Vector representation of "model" document
  - highest similarity to the most other documents in the cluster
- Can also be a "pseudo-document"
  - Words picked from all documents rather than single document

What Does a MEAD Centroid Look Like?

- As computed by CIDR clustering algorithm
  - Radev et al (1999) — Paper
  - R code on Github
- Single-pass clustering
  - Filter words based on their tf*idf
    - N best +/- above word threshold
  - Join with new cluster if above cluster similarity threshold

word	score
suharto	2.48
jakarta	0.58
habibie	0.47
students	0.45
student	0.22
protesters	0.20
asean	0.11
campuses	0.05
geertz	0.04
medan	0.04

word	score
suharto	2.61
jakarta	0.58
habibie	0.53
students	0.43
student	0.21
protesters	0.19
asean	0.10
campuses	0.04
geertz	0.04
medan	0.04

Figure 1: Centroid for cluster 44 (the two scores are after 10,000 (left) and all 22,443 documents (right).

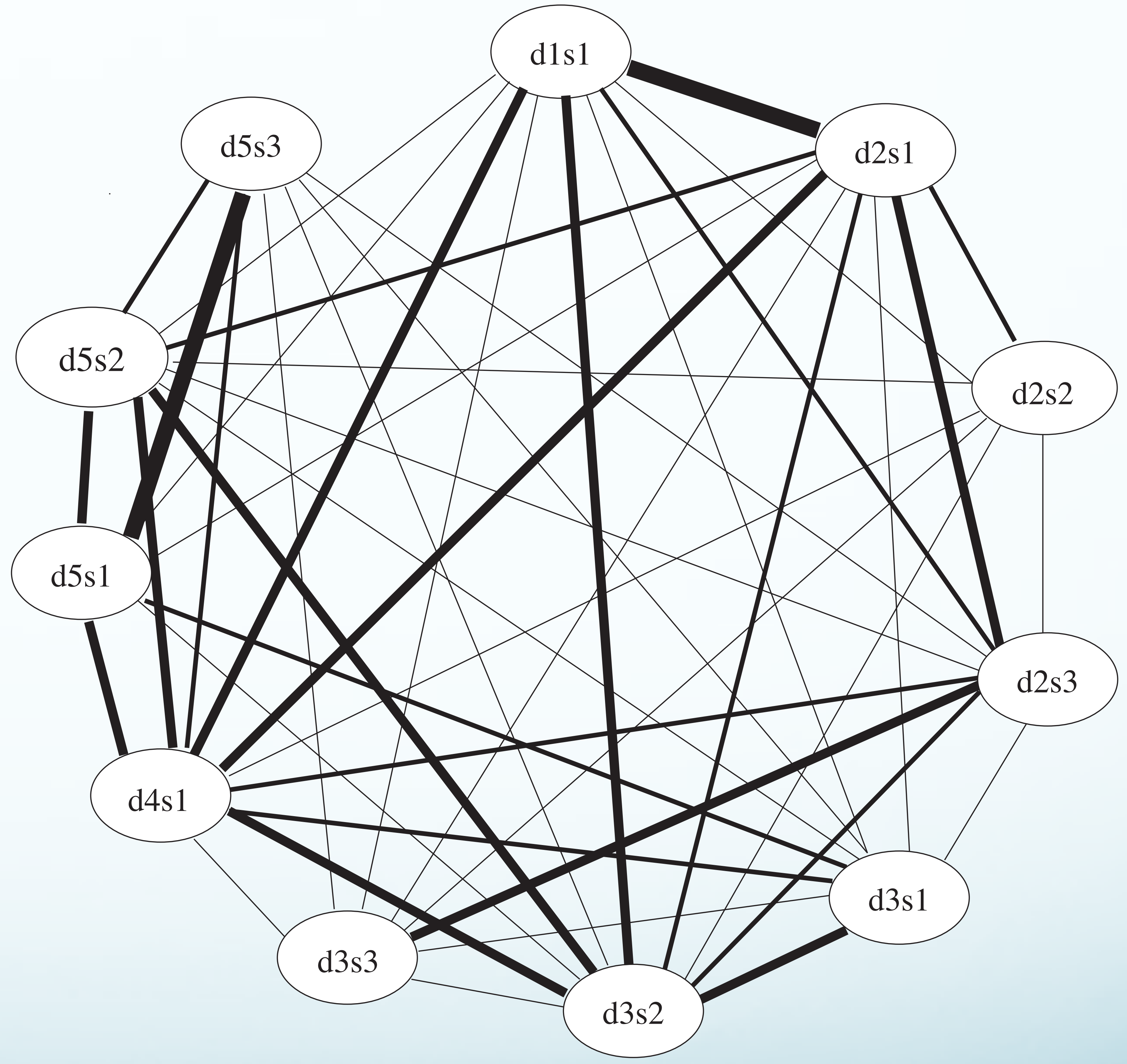
LexRank Revisited

- Begin by computing cosine similarity matrix between sentences in cluster

	1	2	3	4	5	6	7	8	9	10	11
1	**1.00**	0.45	0.02	0.17	0.03	0.22	0.03	0.28	0.06	0.06	0.00
2	0.45	**1.00**	0.16	0.27	0.03	0.19	0.03	0.21	0.03	0.15	0.00
3	0.02	0.16	**1.00**	0.03	0.00	0.01	0.03	0.04	0.00	0.01	0.00
4	0.17	0.27	0.03	**1.00**	0.01	0.16	0.28	0.17	0.00	0.09	0.01
5	0.03	0.03	0.00	0.01	**1.00**	0.29	0.05	0.15	0.20	0.04	0.18
6	0.22	0.19	0.01	0.16	0.29	**1.00**	0.05	0.29	0.04	0.20	0.03
7	0.03	0.03	0.03	0.28	0.05	0.05	**1.00**	0.06	0.00	0.00	0.01
8	0.28	0.21	0.04	0.17	0.15	0.29	0.06	**1.00**	0.25	0.20	0.17
9	0.06	0.03	0.00	0.00	0.20	0.04	0.00	0.25	**1.00**	0.26	0.38
10	0.06	0.15	0.01	0.09	0.04	0.20	0.00	0.20	0.26	**1.00**	0.12
11	0.00	0.00	0.00	0.01	0.18	0.03	0.01	0.17	0.38	0.12	**1.00**

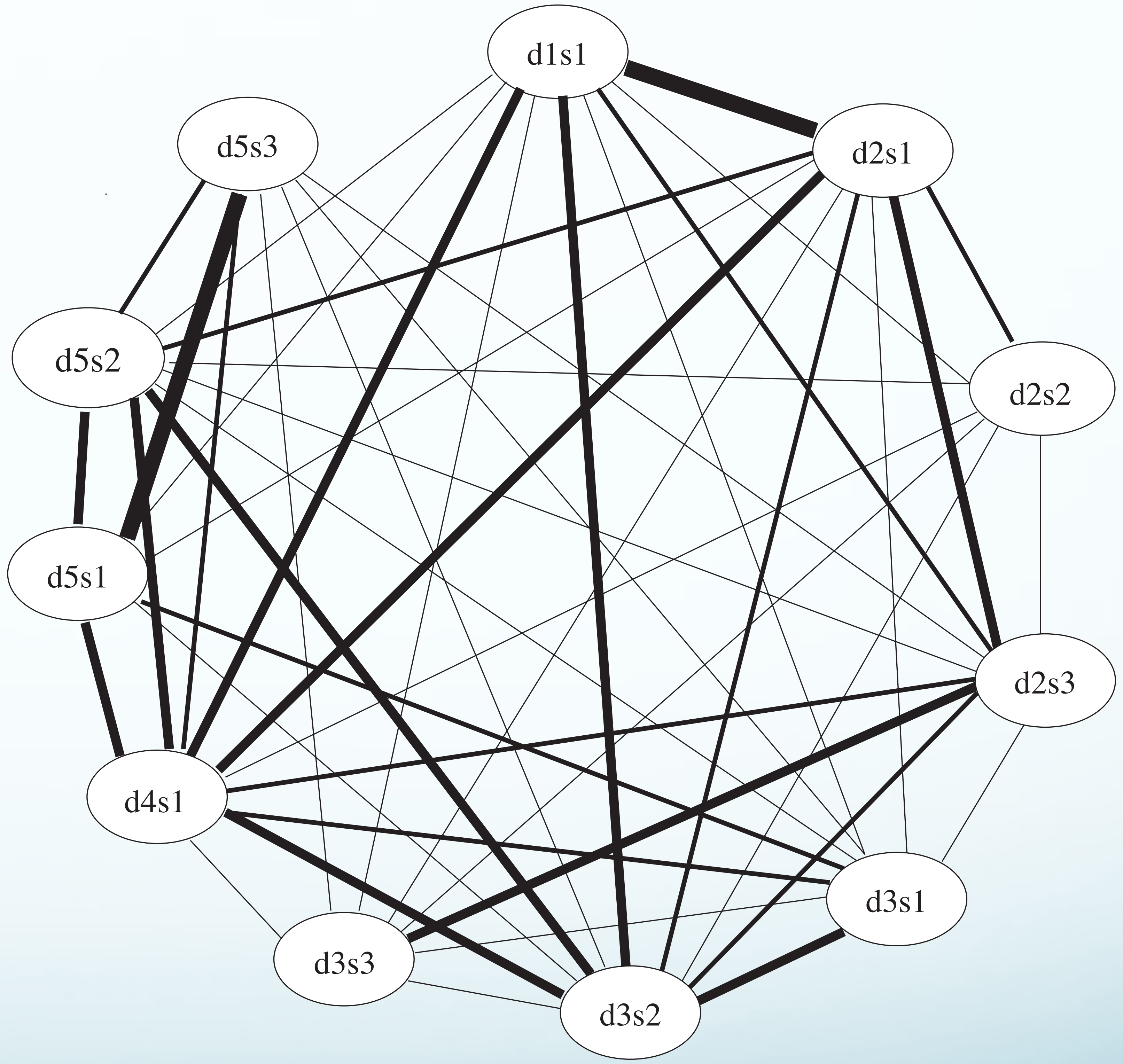
LexRank Revisited

- Use these initial weights to build a graph between sentences
- Cosine similarity sets weights of edges



LexRank Revisited

- Next step: compute node ranks:
  - What we want is ultimately a vector, where each element is the score for our node
  - This is the **eigenvector** of our weight matrix
    - Represents **stable distribution** of markov chain



LexRank Revisited

- Use Power Method: series of matrix transformations:
  - Start with initial guess for **eigenvector** x
  - Calculate $w=Ax$ [w is new matrix]
  - Largest magnitude column in w is estimate of **eigenvalue**
  - Re-scale w by eigenvalue to get next guess for eigenvector x
  - Repeat until convergence

LexRank Revisited

- Example of power method converging toward approximation

$$A_z^{(4)} = \begin{bmatrix} 2 & 8 & 10 \\ 8 & 3 & 4 \\ 10 & 4 & 7 \end{bmatrix} \begin{Bmatrix} 0.9243 \\ 0.7080 \\ 1.0 \end{Bmatrix} = \begin{Bmatrix} 17.513 \\ 13.519 \\ 19.075 \end{Bmatrix} = (19.075) \begin{Bmatrix} 0.9181 \\ 0.7087 \\ 1.0 \end{Bmatrix}$$

$$A_z^{(5)} = \begin{bmatrix} 2 & 8 & 10 \\ 8 & 3 & 4 \\ 10 & 4 & 7 \end{bmatrix} \begin{Bmatrix} 0.9181 \\ 0.7087 \\ 1.0 \end{Bmatrix} = \begin{Bmatrix} 17.506 \\ 13.471 \\ 19.016 \end{Bmatrix} = (19.016) \begin{Bmatrix} 0.9206 \\ 0.7084 \\ 1.0 \end{Bmatrix}$$

$$A_z^{(6)} = \begin{bmatrix} 2 & 8 & 10 \\ 8 & 3 & 4 \\ 10 & 4 & 7 \end{bmatrix} \begin{Bmatrix} 0.9206 \\ 0.7084 \\ 1.0 \end{Bmatrix} = \begin{Bmatrix} 17.508 \\ 13.490 \\ 19.040 \end{Bmatrix} = (19.040) \begin{Bmatrix} 0.9196 \\ 0.7085 \\ 1.0 \end{Bmatrix}$$

$$A_z^{(7)} = \begin{bmatrix} 2 & 8 & 10 \\ 8 & 3 & 4 \\ 10 & 4 & 7 \end{bmatrix} \begin{Bmatrix} 0.9196 \\ 0.7085 \\ 1.0 \end{Bmatrix} = \begin{Bmatrix} 17.507 \\ 13.482 \\ 19.030 \end{Bmatrix} = (19.030) \begin{Bmatrix} 0.9200 \\ 0.7085 \\ 1.0 \end{Bmatrix}$$

LexRank Revisited

- Example of power method converging toward approximation

$$A_z^{(4)} = \begin{bmatrix} 2 & 8 & 10 \\ 8 & 3 & 4 \\ 10 & 4 & 7 \end{bmatrix} \begin{Bmatrix} 0.9243 \\ 0.7080 \\ 1.0 \end{Bmatrix} = \begin{Bmatrix} 17.513 \\ 13.519 \\ 19.075 \end{Bmatrix} = (19.075) \begin{Bmatrix} 0.9181 \\ 0.7087 \\ 1.0 \end{Bmatrix}$$

$$A_z^{(5)} = \begin{bmatrix} 2 & 8 & 10 \\ 8 & 3 & 4 \\ 10 & 4 & 7 \end{bmatrix} \begin{Bmatrix} 0.9181 \\ 0.7087 \\ 1.0 \end{Bmatrix} = \begin{Bmatrix} 17.506 \\ 13.471 \\ 19.016 \end{Bmatrix} = (19.016) \begin{Bmatrix} 0.9206 \\ 0.7084 \\ 1.0 \end{Bmatrix}$$

$$A_z^{(6)} = \begin{bmatrix} 2 & 8 & 10 \\ 8 & 3 & 4 \\ 10 & 4 & 7 \end{bmatrix} \begin{Bmatrix} 0.9206 \\ 0.7084 \\ 1.0 \end{Bmatrix} = \begin{Bmatrix} 17.508 \\ 13.490 \\ 19.040 \end{Bmatrix} = (19.040) \begin{Bmatrix} 0.9196 \\ 0.7085 \\ 1.0 \end{Bmatrix}$$

$$A_z^{(7)} = \begin{bmatrix} 2 & 8 & 10 \\ 8 & 3 & 4 \\ 10 & 4 & 7 \end{bmatrix} \begin{Bmatrix} 0.9196 \\ 0.7085 \\ 1.0 \end{Bmatrix} = \begin{Bmatrix} 17.507 \\ 13.482 \\ 19.030 \end{Bmatrix} = (19.030) \begin{Bmatrix} 0.9200 \\ 0.7085 \\ 1.0 \end{Bmatrix}$$

LexRank Revisited

- Don’t worry, we don’t expect you to have linear algebra nailed!

LexRank Demo

- *Link to LexRank Demo:* http://clair.si.umich.edu/demos/lexrank/

Analyzing Discourse Features:

Louis et al (2014)

Experimental Setup

- Design different features, both **discourse-related** and **non-discourse**
  - Using model summaries (human-generated)
    - Perform statistical significance tests on included vs. non-included sentences
    - χ^2 (categorical) t-test (continuous)
- Use features in logistic regression classifier (MaxEnt)
  - Use to select sentences for extraction
- Evaluation:
  - F_1 against model sentences
  - ROUGE over summary sentences

Experimental Setup

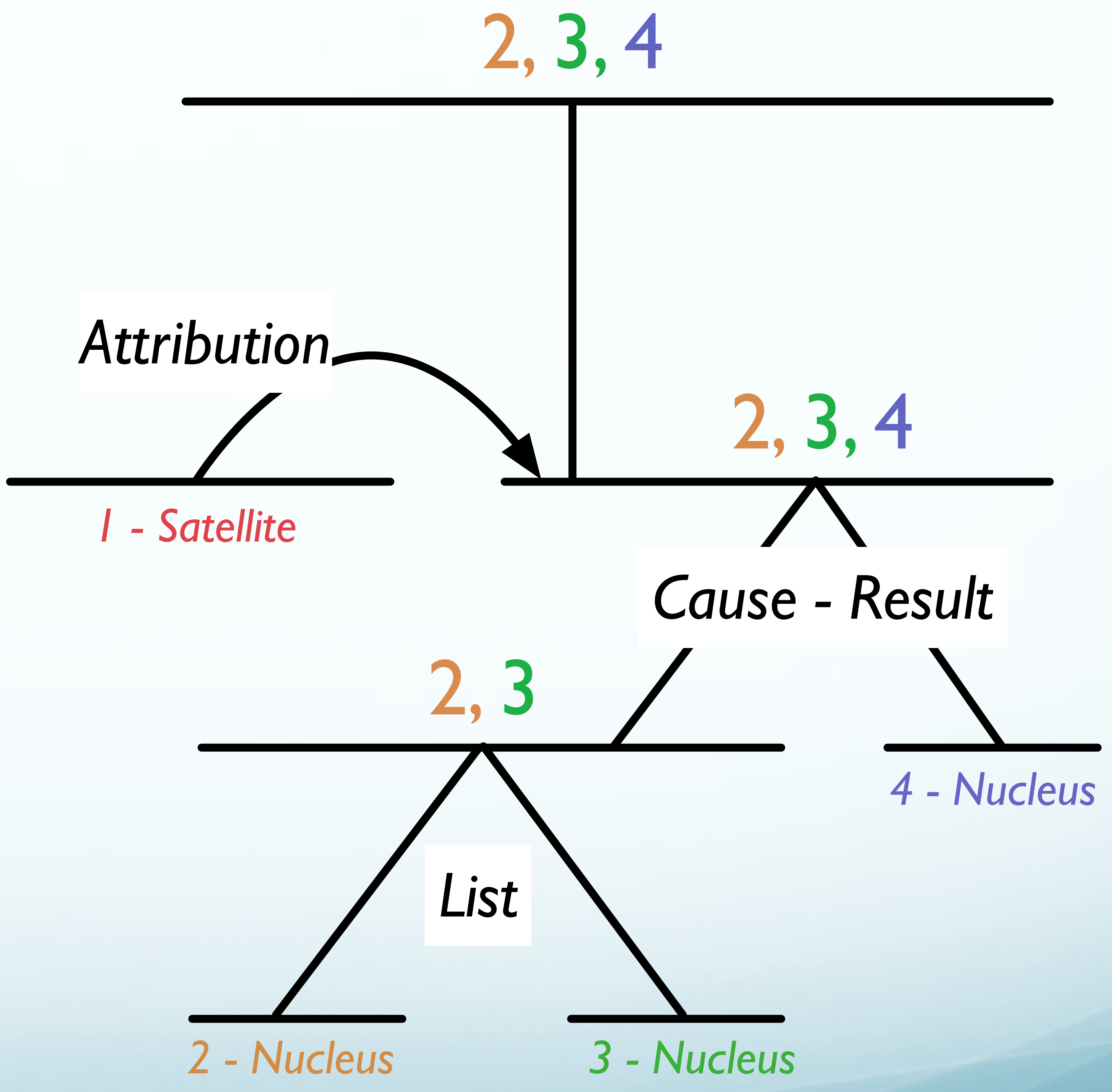
- Caveat:
  - Experimental approach is using human-created discourse analyses
  - Authors do not attempt using automatic discourse parsers for analyses
  - Purely a study of how well discourse features correlate in an idealized setting

How Would This be Applied?

- Learn and apply classifiers for segmentation and parsing of discourse
- Assign coherence relations between spans
- Create a representation over whole text → parse
- Use parsed representations as features in classifier for content selection

Discourse (RST) Structure Example

- 1. [Mr. Watkins said]
- 2. [volume on Interprovincial's system is down about 2% since January]
- 3. [and is expected to fall further,]
- 4. [making expansion unnecessary until perhaps the mid-1990s.]

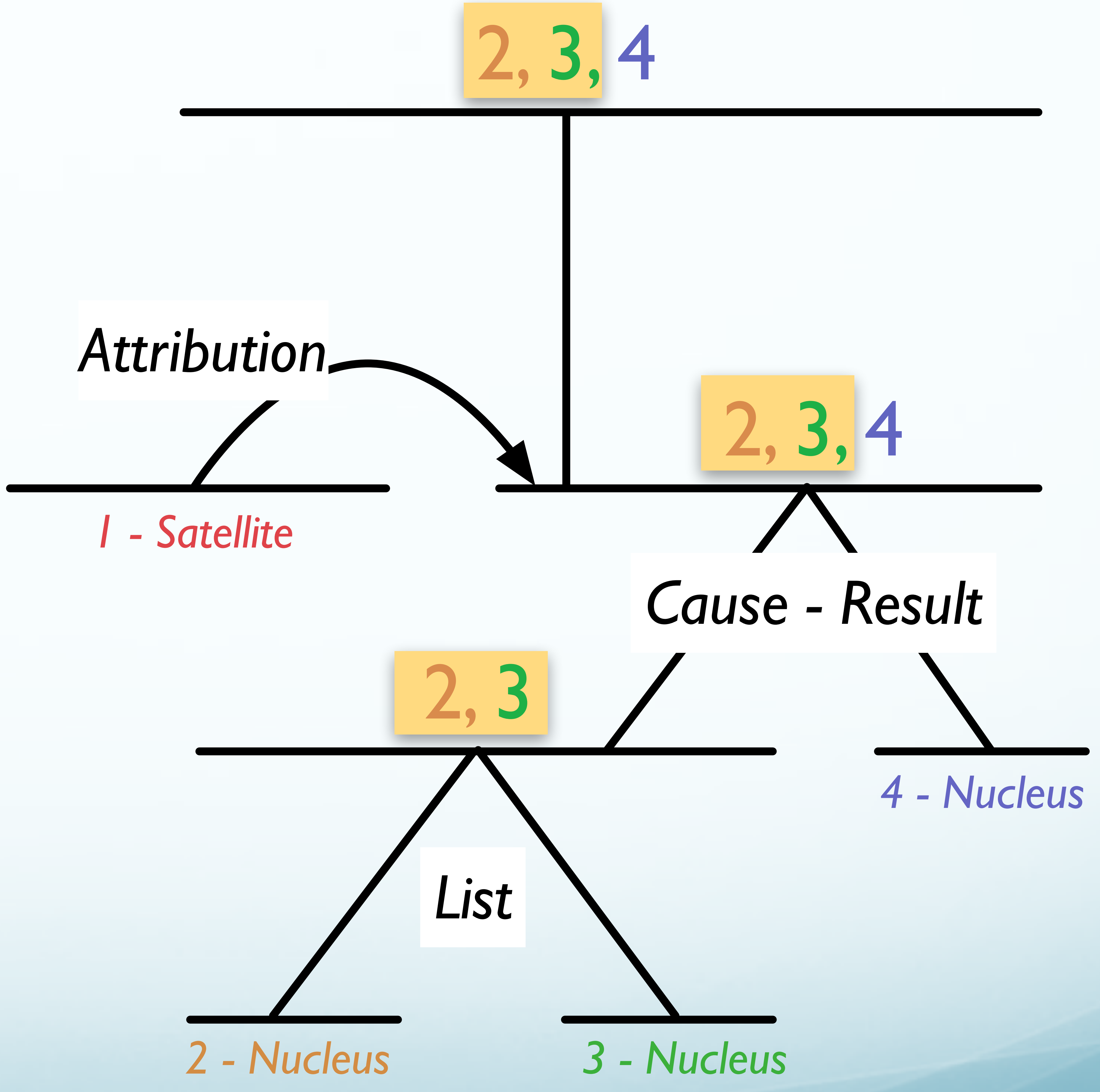


Discourse Structure Features

- **Satellite penalty**
  - For each EDU — number of satellite nodes between EDU and root
    - 1 satellite in tree: one step to root: penalty = 1
  - **Intuition**: Helpful summary content will be closely related to nucleus.

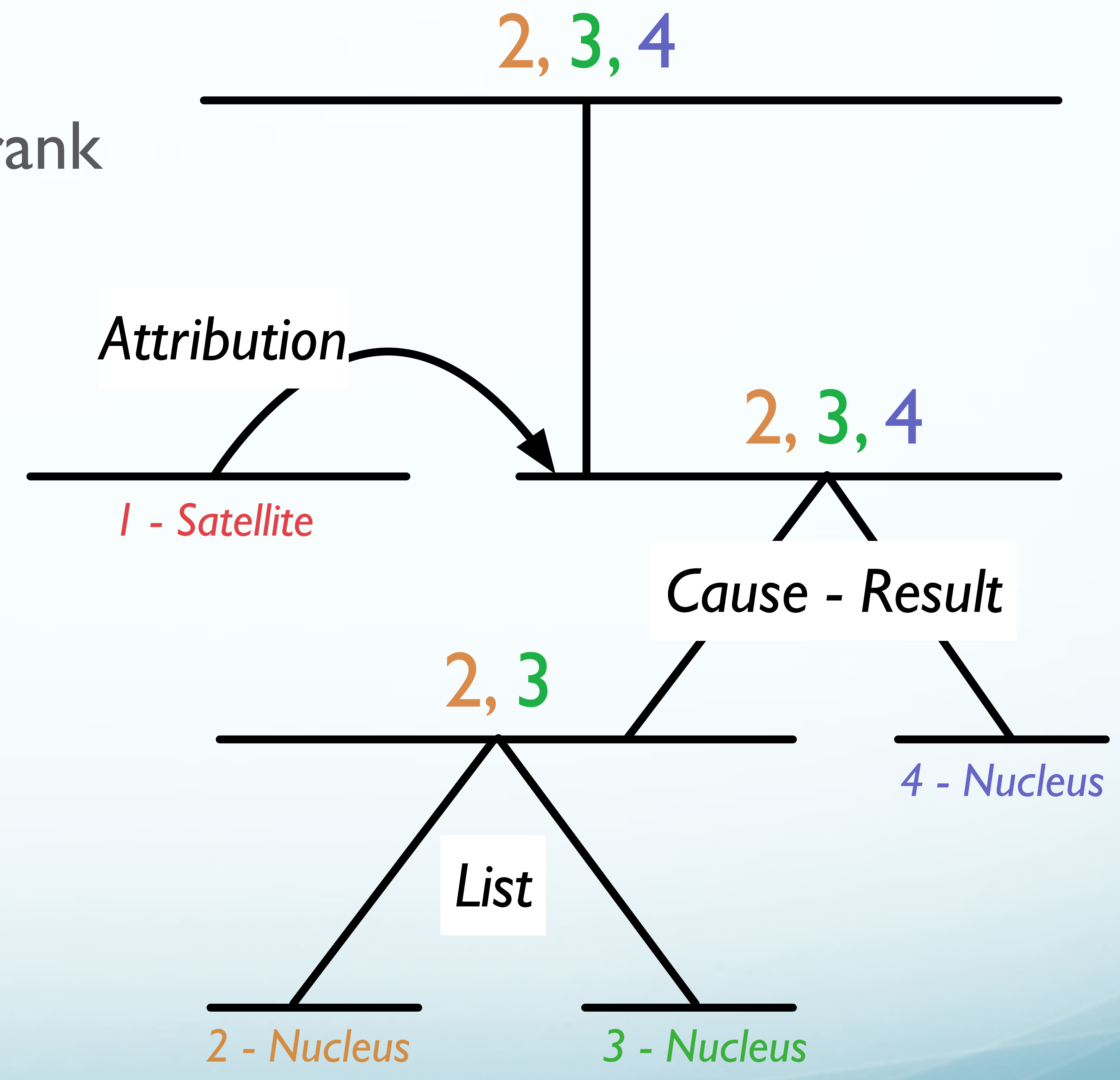
Discourse Structure Features

- **Promotion set:**
  - Nuclear units at some level of tree
    - At leaves, EDUs are themselves nuclear
  - Intuition:
    - The more times a unit is promoted in the tree, the more necessary its concepts to understanding the whole discourse



Discourse Structure Features

- **Depth score:**
  - Distance from lowest tree level to EDU's highest rank
    - **2,3,4**: score=4
    - **1**: score=3
- **Promotion score:**
  - # of levels span is promoted
    - **1**: score = 0
    - **4**: score = 2
    - **2,3**: score = 3



Converting to Sentence Level

- Each feature has:
  - Raw score
  - Normalized score: $\dfrac{\text{raw score}}{\text{sentence length}}$
- Sentence score for a feature:
  - Maximum value over all EDUs in sentence

“Semantic” Features

- Represent sentences purely in terms of their discourse relationships
- **Binary features:**
  - Implicit vs. Explicit
  - sentence_in_{RELATION_NAME}
  - sentence_**contains**_{ARG_1|ARG_2}_of_{RELATION_NAME} **(multi-sentential)**
  - sentence_**expresses**_{RELATION_NAME} **(both args in single sent)**
- **Real-valued features:**
  - Number of relations
  - Distance between arguments within sentence

Example 1

- *In addition, its machines are easier to operate, so customers require less assistance from software.*

- Is there an explicit discourse marker?
  - Yes, "**so**"

- Discourse relation?
  - ***Contingency***

Example II

- *(1) Wednesday's dominant issue was Yasuda & Marine Insurance, which continued to surge on rumors of speculative buying. (2) It ended the day up 80 yen to 1880 yen.*

- Is there an explicit discourse marker?
  - **No**

- Is there a relation?
  - **Yes, Implicit**.

- What relation?
  - **Expansion**. (More specifically, restatement).

Non-Discourse Features

- Sentence length
- Sentence position
- Probabilities of words in sentence
  - mean, sum, product
- # of signature words (LLR)

Statistical Analysis

Statistical Analysis

- Used model summaries to analyze whether features were predictive
  - for a given feature-sent pair in docset…
  - How likely was that sentence to appear in summary?

Significant Features: Summary Sentences

- Structure:
  - depth score
  - promotion score
- Semantic:
  - Arg1 of Explicit Expansion
  - Arg1 of Implicit Contingency
  - Arg1 of Implicit Expansion
  - Distance to other Argument
- Non-discourse:
  - length
  - 1st sent in article
  - 1st sent in paragraph
  - offset from paragraph end
  - # signature terms
  - mean content word probabilities
  - sum content word probabilities

All VERY small p-values

Significant Features: Non-Summary Sentences

- Structure:
  - satellite penalty
- Semantic:
  - expresses explicit expansion
  - expresses explicit contingency
  - Arg2 of implicit temporal
  - Arg2 of implicit expansion
  - Arg2 of implicit contingency
  - # of shared implicit relations
  - total shared relations
- Non-discourse:
  - offset from paragraph start
  - offset from article start
  - sentence probability

Observations

- Non-discourse features good cues to summary
- Structural features match intuition
- Semantic features
  - Relatively few useful features for selecting summaries
  - Most features associated with non-summary… but most sentences are non-summary

Evaluation

- Structural is best, both alone and in combination
- Best overall combines all types
- Both F_1 and ROUGE-1

Features used	Acc	P	R	F
structural	78.11	63.38	22.77	33.50
semantic	75.53	44.31	5.04	9.05
non-discourse (ND)	77.25	67.48	11.02	18.95
ND + semantic	77.38	59.38	20.62	30.61
ND + structural	78.51	63.49	26.05	36.94
semantic + structural	77.94	58.39	30.47	40.04
structural + semantic + ND	78.93	61.85	34.42	44.23

Features	ROUGE	Features	ROUGE
structural + semantic + ND	0.479	ND	0.432
structural + ND	0.468	LEAD	0.411
structural + semantic	0.453	semantic	0.369
semantic + ND	0.444	TS	0.338
structural	0.433		

*TS = "topic signature"

Graph-Based Comparison

- Page-Rank Based Centrality Computed Over
  - RST Link Structure
  - Graphbank Link Structure
  - LexRank (sentence cosine similarity)
- Quite similar, but:
  - F_1: LR > GB > RST
  - ROUGE: RST > LR > GB

	Acc	P	R	F	ROUGE-1
RST-struct	81.61	63.00	31.56	42.05	0.569
GB-struct	82.58	62.50	39.16	48.15	0.508
LEX-struct	83.23	75.17	41.14	53.18	0.557

Notes

- Single document, short (100 word) summaries
  - What about multi-document? Longer?
- Structure relatively better
- Manually labeled discourse structure, relations
  - Some automatic systems available, but not perfect
  - Better at getting the structure than the exact relation
    - Especially implicit

Topic Orientation & Optimization

Topic-Focused Summarization

- "Query-focused" or "Guided"
- Extrinsic task vs. generic:
  - Why are we creating this summary?
  - Viewed as complex question answering (vs. factoid)
- High variation in human summaries
  - Depending on perspective, different content is focused

Topic-Focused Summarization: Key Idea

- Target response to specific question, topic in documents
- Later TACs identify topic categories and aspects
  - e.g. Natural disasters: who, what, where, when

Topic-Focused Summarization: Evaluation

- When treated as a factoid/sentence selection problem:
  - Mean Rank Reciprocal (MRR)
    - Inverse of rank of correct answer
  - Total Reciprocal Document Rank (TRDR)
    - Total of all reciprocal ranks of all answers system suggests
      - (Usually taken as average)

Query-Focused LexRank

Otterbacher et al (2005)

- Focus on sentences relevant to query
  - Rather than computing similarity of sentences to all other sentences
- How do we measure relevance?
  - tf*idf-like measure over sentences & query
  - Compute sentence-level "idf_w"
  - N = # of sentences in cluster
  - sf_w = # of sentences with w

$$idf_w = \log\left(\frac{N+1}{0.5+sf_w}\right)$$

Query-Focused LexRank

Otterbacher et al (2005)

$$rel(s \mid q) = \sum_{w \in q} \log(tf_{w,s} + 1) \cdot \log(tf_{w,q} + 1) \cdot idf_w$$

- Relevance of sentence ***s*** given query ***q***
  - Log Sum (Product) of:
    - term frequency for word ***w*** in sentence
    - term frequency for word ***w*** in query
    - idf_w for word across all sentences

Updated LexRank Model

- Combines original similarity weighting with query
  - Mixture model of query relevance, sentence similarity (LexRank)

$$p(s \mid q) = d \frac{rel(s \mid q)}{\sum_{z \in C} rel(z \mid q)} + (1-d) \sum_{v \in C} \frac{sim(s,v)}{\sum_{z \in C} sim(z,v)} p(v \mid q)$$

  - ***d*** controls "bias": i.e. relative weighting toward query relvance

Tuning & Assessment

- Parameters:
  - **Similarity threshold**: filters adjacency matrix
  - **Question bias**: Weights emphasis on question focus
- Empirical results:
  - Best similarity threshold: 0.14–0.2
  - Best question bias: high: 0.8–0.95
- Higher question bias in LexRank improves MRR

Other Strategies

- Methods depend on base system design
  - All aim to incorporate similarity with query/topic
- **CLASSY HMM** (Conroy et al, 2005):
  - Add question overlap feature to HMM vector — $\mathbf{log}(\#\_query\_tokens\_in\_sentence + 1)$
  - Query tokens: filtered to NN, VB, JJ, RB, or NNP
- **FastSum** (Schilder & Kondadadi, 2008):
  - SVM regression on sentences
  - Adds topic title frequency feature:
    - Proportion of words in sent which appear in title
- Others: require minimum number of topic words

Overview

- Many similar strategies:
  - Features, weighting, ranking: overlap based
- Actual evaluation impact:
  - Not necessarily very large (e.g. 0.003 ROUGE)
    - But can be useful
- Aggressive approaches can have large negative impact
  - i.e. explicitly adding NER spans

Optimization Approaches to Reducing Redundancy

Optimization Approaches to Reducing Redundancy

- DPP: Determinantal Point Processes [python GH] (Kulesza & Taskar 2012)
  - Set models balancing information importance w/diversity
- ICSISumm: Uses Integer Linear Programming frame [code] (Gillick et al, 2008)
  - Optimizes coverage of key bigrams weighted by document frequency
- OCCAMS_V (Davis et al, 2012)
  - Uses LSA (Latent Semantic Analysis) to weight terms
  - Sentence selection via optimization problems:
    - Budgeted maximal coverage; knapsack

ICSISumm

- Key ideas:
  - Cast summarization as optimization problem
  - Identify important "concepts" to incorporate
  - Build best such summary
  - Implemented as Integer Linear Programming

Integer Linear Programming

- Aka ILP
- An integer linear program specifies
  - A single linear maximization term
  - Subject to linear equality/inequality constraints
  - Involving integer valued variables
- **For summarization:**
  - Map summary requirements to ILP elements

Summarization as ILP

- **Summary goal:**
  - "Best" summary
- **Summary requirements:**
  - Minimize redundancy
  - Within desired length

- Maximization term: $\sum_i w_i c_i$
- Implicit:
  - Length Constraint $\sum_j l_j s_j < L$
  - Coverage Constraint $\sum_j s_j o_{ij} \geq c_i \forall i$

$$s_j o_{ij} \leq c_i \forall i, j$$

Weight

Representing Concepts

- Concepts = Bigrams
  - Stemmed
  - No stopword-only bigrams
  - Occuring in at least 3 documents
- Weights
  - Document frequency
  - # Of Documents (from cluster) for bigram
- Selected sentences must contain ≥ 2 query terms

Results

- After using open source solver
- 2009 results:
  - 2nd best pyramid, ROUGE-2
  - Best ROUGE-3, ROUGE-4