Computational Models of Discourse: Segmentation

Regina Barzilay

MIT

October, 2005

Today

- Finish leftovers
  - Learning PCFGs
- Computational models of discourse

Learning PCFGs

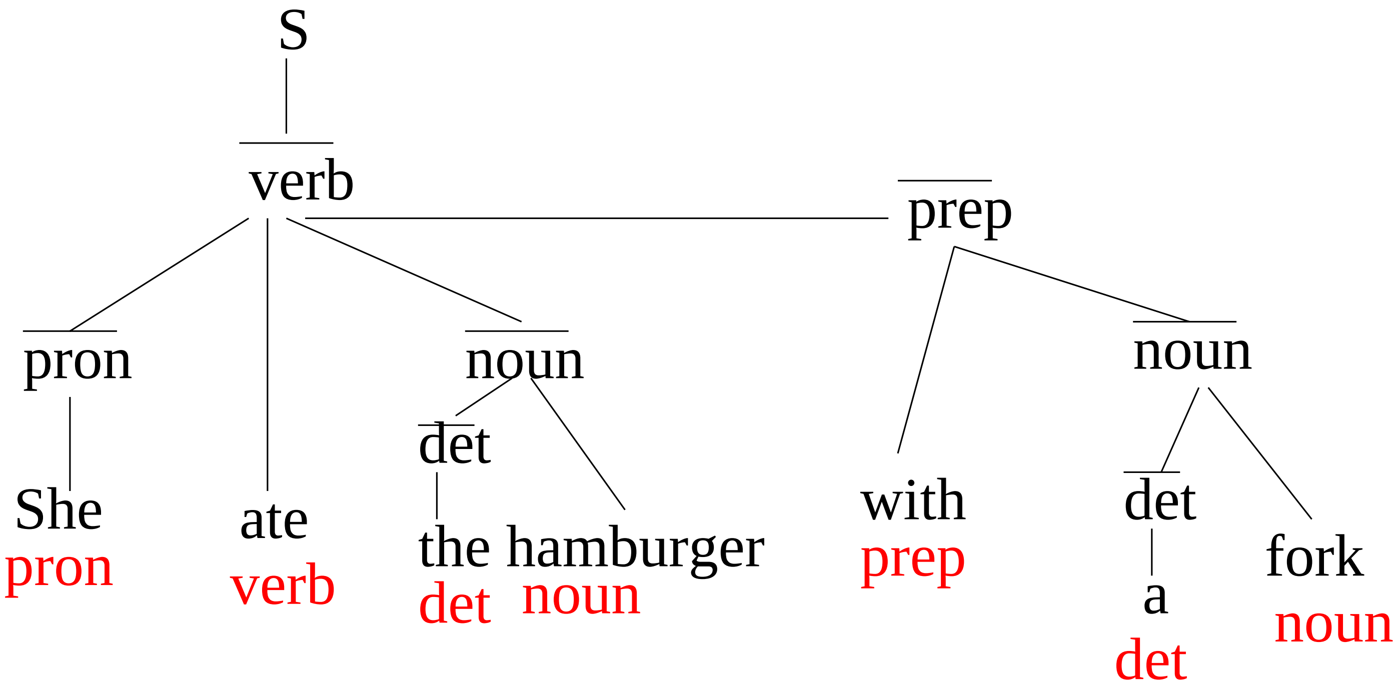
(Carroll&Charniak, 1992)

Goal: Learning grammars for natural language

- Divide the corpus into two parts: the rule corpus and the training corpus.
- For all the sentences in the rule corpus, generate all rules which might be used to parse the sentence, subject to constraints which we will specify later.
- Estimate the probabilities for the rules.
- Using the training corpus, improve our estimate of probabilities.
- Delete all rules with probability $\leq \delta$ for some small δ.

Rule Generation: Dependency Format

Informally, a dependency grammar produces a set of terminals connected by a set of directed arcs — one arc for every terminal except the root terminal



Dependency Grammar

- Target: a dependency grammar $< S, N, R >$

  S is the start symbol

  N is a set of terminals

  R is a set of rewrite rules, where

  $R \subseteq \{S \rightarrow \bar{n} | n \in N\} \cup \{\bar{n} \rightarrow \alpha n \beta | n \in N, \alpha, \beta \in \Gamma\}$,

  Γ is a set of strings of zero or more $\bar{a}$, for $a \in N$

- Assumption: POS tags are provided

- Theorem: A sentence of length n, consisting of all distinct terminals will have $n(2^{n-1} + 1)$ dependency grammar rules to confirm to it

Example

Induce PCFG, given the following corpus:

“noun verb”

“verb noun”

“verb”

“det noun verb”

“verb det noun”

Rule	1 ITER	6 ITER	20 ITER
$S \longrightarrow \bar{det}$	0.181818	0.0	0.0
$S \longrightarrow \bar{noun}$	0.363636	0.0	0.0
$S \longrightarrow \bar{verb}$	0.454545	1.0	1.0
$\bar{det} \longrightarrow det$	0.250000	1.0	1.0
$\bar{det} \longrightarrow det\ \bar{noun}$	0.250000	0.0	0.0
$\bar{det} \longrightarrow det\ \bar{verb}$	0.125	0.0	0.0
$\bar{det} \longrightarrow verb\ \bar{det}$	0.125	0.0	0.0
$\bar{det} \longrightarrow verb\ \bar{det}\ \bar{noun}$	0.125	0.0	0.0
$\bar{noun} \longrightarrow noun$	0.333333	0.781317	0.998847
$\bar{noun} \longrightarrow \bar{det}\ noun$	0.166667	0.218683	0.01153
$\bar{verb} \longrightarrow \bar{noun}\ verb$	0.153846	0.286749	0.200461
$\bar{verb} \longrightarrow verb\ \bar{noun}$	0.153846	0.288197	0.200461

Rule Generation

We have to prune rule space!

- Order sentences by length and generate rules incrementally
- Do not consider rules that were discarded on previous stages
- Limit the number of symbols on the right-hand side of the rule

Algorithm

```
Loop for i from 2 until $i >$ sentence-length-stopping point
    Add rules required for the sentences with length $i$ from the rule creation subset
    Estimate the probabilities for all rules, based upon all sentences of length $\leq i$ from the rule training subset
    Remove any rules with probability $\leq \delta$ if its probability doesn’t increase
```

Reestimation

- We have sentences $S_1, \ldots, S_n$. Trees are hidden variables.

$$L(\theta) = \sum_i \log \sum_T P(S_i, T|\theta)$$

- Basic quantity needed for re-estimating with EM:

$$\theta_{\alpha\to\beta} = \frac{\sum_i Count(S_i, \alpha \to \beta)}{\sum_i \sum_{s \in R(\alpha)} Count(S_i, s)}$$

- There are efficient algorithms for calculating

$$Count(S_i, r) = \sum_T P(T|S_i, \theta^{t-1}) Count(S_i, T, r)$$

for a PCFG. See Inside-Outside algorithm (Baker, 1979)

Experiment 1

- Use grammar from the handout
- Randomly generate 1000 words for the rule corpus, and 9000 for the training corpus
- Evaluation: compare the output with the generated grammar
- Constraint: rules were required to have fewer than five symbols on their right-hand side

Results

- Successfully minimizes a cross entropy (1.245 bits/word on the training of the learned grammar vs. 1.220 bits/word of the correct grammar)
- Miserably fails to recover the correct grammar
  - 300 unsuccessful attempts

.220	$\bar{pron}$	→	$pron\ \bar{verb}$
.214	$\bar{pron}$	→	$\bar{prep}\ pron$
.139	$\bar{pron}$	→	$pron\ \bar{verb}\ \bar{det}$
.118	$\bar{pron}$	→	$\bar{verb}\ pron$

Experiment 2

Place more restrictions on the grammar

Specify what non-terminals may appear on the right-hand side of a rule with a particular non-terminal on the left

- The algorithm converges to the correct grammar

	noun	verb	pron	det	prep	adj	wh	.
noun				+	+	+	+	
verb	+		+		+			
pron		–						
det						–		

Adding Knowledge to Grammar Induction Algorithms

- Carrol&Charniak (1992): restrictions on the rule format

- Magerman&Marcus (1990): use a di-stituent grammar to eliminate undesirable rules

- Pereira&Schabes (1992): use partially bracketed corpora

Learning Constituents

Are syntactic patterns evident in a corpus? (Klein, 2005)

- Compute context for each POS

Tag	Top Context by Frequency
DT	(IN-NN), (IN-JJ), (IN-NNP), (VB-NN)
JJ	(DT-NN), (IN-NNS), (IN-NN), (JJ-NN)

- Cluster POS based on their context

Learning Constituents

The most similar POS pairs based on their context

Rank	Tag Pairs
1	(VBZ, VBD)
2	(DT, PRP$)
3	(NN, NNS)
4	(WDT, WP)
5	(VBG, VBN)

Learning Constituents

The most similar POS sequence pairs based on their context

Rank	Tag Pairs
1	(NNP NNP, NNP NNP NNP)
2	(DT JJ NN IN, DT NN IN)
3	(NNP NNP NNP NNP, NNP NNP NNP)
4	(DT NNP NNP, DT NNP)
5	(IN DT JJ NN, IN DT NN)

Learning Constituents (Clark, 2001)

- Identify frequent POS sequences in a corpus
- Cluster them based on their context
- Filter out spurious candidates
  - Based on mutual information before the candidate constituent and the symbol after — they are not independent

Grammar Induction: Summary

- Language acquisition problem
- Three unsupervised induction algorithms:
  - Vocabulary Induction
  - HMM-topology induction
  - PCFG induction

Computational Models of Discourse

Active networks and virtual machines have a long history of collaborating in this manner. The basic tenet of this solution is the refinement of Scheme. The disadvantage of this type of approach, however, is that public-private key pair and red-black trees are rarely incompatible.

SCIgen: An Automatic CS Paper Generator

- An output of a system that automatically generates scientific papers (Stribling et al., 2005):

> Active networks and virtual machines have a long history of collaborating in this manner. The basic tenet of this solution is the refinement of Scheme. The disadvantage of this type of approach, however, is that public-private key pair and red-black trees are rarely incompatible.

Courtesy of Jeremy Stribling. Used with permission.
See: http://pdos.csail.mit.edu/scigen/

- The paper was accepted to a conference (not ACL!)

Reference Resolution: Example

Text removed for copyright reasons.

Source: transcription of the Burns & Allen "Salesgirl" comedy routine.

Reference Resolution: Example

Text removed for copyright reasons.

Source: transcription of the Burns & Allen "Salesgirl" comedy routine.

Transcribed lecture example

What information is the speaker trying to convey?

I've been talking uh I I've been multiplying matrices already but certainly time for me to discuss the rules for matrix multiplication and the interesting part is the many ways you can do it and they all give the same answer so it's and they're all important so matrix multiplication and then uh come inverses so we're uh we mentioned the inverse of a matrix but there's that's a big deal lots to do about inverses and how to find them okay so I'll begin with how to multiply two matrices first way okay so suppose I have a matrix A multiplying a matrix B and giving me a result well I could call it C

After Some Editing

I’ve been talking – uh, I I’ve been multiplying matrices already, but certainly time for me to discuss the rules for matrix multiplication.

And the interesting part is the many ways you can do it, and they all give the same answer.

So it’s – and they’re all important.

So matrix multiplication, and then, uh, come inverses.

So we’re – uh, we – mentioned the inverse of a matrix, but there’s – that’s a big deal.

Lots to do about inverses and how to find them.

Okay, so I’ll begin with how to multiply two matrices.

First way, okay, so suppose I have a matrix A multiplying a matrix B and – giving me a result – well, I could call it C.

What We Really Want

The method for multiplying two matrices A and B to get $C = AB$ can be summarized as follows:

1. **Rule 8.1** To obtain the element in the r^{th} row and c^{th} column of C, multiply each element in the r^{th} row of A by the corresponding element in the c^{th} column of B, then add up all the products. $\ldots$

Adding structural information

Example 1: Graduate-level AI Class Lecture

. . . We're going to be talking about agents. This word used to mean something that acts. Way back when I started working on AI, agent meant something that took actions in the world. Now, people talk about Web agents that do things for you, there's publicity agent, etc. When I talk about agents, I mean something that acts. So, it could be anything from a robot, to a piece of software that runs in the world and gathers information and takes action based on that information, to a factory, to all the airplanes belonging to United Airlines. So, I will use that term very generically. When I talk about computational agents that behave autonomously, I'll use agent as a shorthand for that. So, how do we think about agents? How can we begin to formalize the problem of building an agent? Well, the first thing that we're going to do, which some people object to fairly violently, is to make a dichotomy between an agent and its environment. . . .

Agents

Software that gathers information about an environment and takes actions based on that information.

- a robot
- a web shopping program
- a factory
- a traffic control system

Modeling Discourse: Applications

- Coherence assessment
- Coreference resolution
- Segmentation
- Summarization
- . . .

Modeling Text Structure

Key Question: Can we identify consistent structural patterns in text?

Discourse Exhibits Structure!

- Discourse can be partitioned into segments, which can be connected in a limited number of ways
- Speakers use linguistic devices to make this structure explicit
  cue phrases, intonation, gesture
- Listeners comprehend discourse by recognizing this structure
  - Kintsch, 1974: experiments with recall
  - Haviland&Clark, 1974: reading time for given/new information

Models of Discourse Structure

- Cohesion-based
- Content-based
- Rhetorical
- . . .

Cohesion-based Model

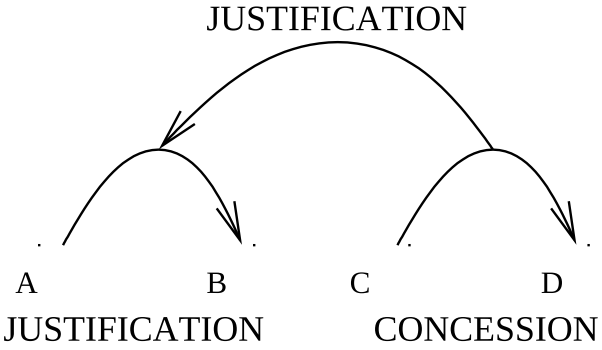
There was once a Prince who wished to marry a Princess; but then she must be a real Princess. He travelled all over the world in hopes of finding such a lady; but there was always something wrong. At last he returned to his palace quite cast down, because he wished so much to have a real Princess for his wife.

Content-based Model

- Wants to marry
- Fails miserably
- Upset
- . . .

Rhetorical Structure Model

[There was once a Prince who wished to marry a Princess]$_1$; [but then she must be a real Princess]$_2$. [He travelled all over the world in hopes of finding such a lady]$_3$.



What is Segmentation?

Segmentation: determining the positions at which topics change in a stream of text or speech.

SEGMENT 1: OKAY
tsk There's a farmer,
he looks like ay uh Chicano American,
he is picking pears.
A-nd u-m he's just picking them,
he comes off the ladder,
a-nd he- u-h puts his pears into the basket.
SEGMENT 2: U-h a number of people are going by,
and one of them is um I don't know,
I can't remember the first . . . the first person that goes by

Flow model of discourse

Chafe’76:

> “Our data … suggest that as a speaker moves from focus to focus (or from thought to thought) there are certain points at which they may be a more or less radical change in space, time, character configuration, event structure, or even world … At points where all these change in a maximal way, an episode boundary is strongly present.”

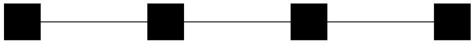
From Chafe, W. L. "The flow of thought and the flow of language." In Syntax and Semantics: Discourse and Syntax. Vol. 12. Edited by Talmy Givón. Burlington, MA: Academic Press, 1979.

Word Distribution in Text

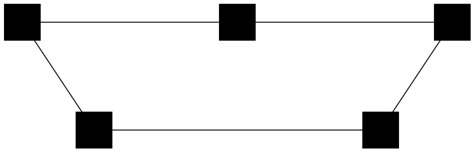
Table removed for copyright reasons.

Please see: Figure 2 in Hearst, M. "Multi-Paragraph Segmentation of Expository Text." *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL 94)*, June 1994. (http://www.sims.berkeley.edu/~hearst/papers/tiling-acl94/acl94.html)

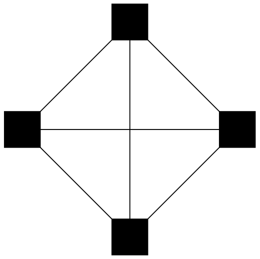
Skorochodko’s Text Types



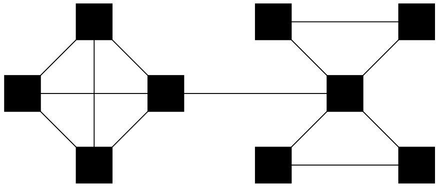
Chained



Ringed



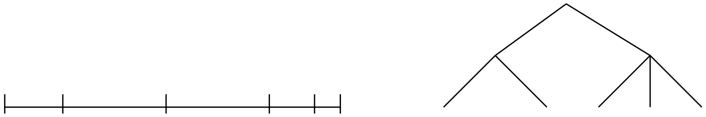
Monolith



Piecewise

Types of Structure

- Linear vs. hierarchical

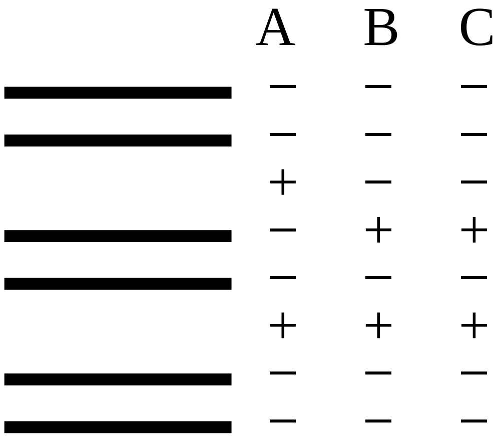


- Typed vs. untyped

Our focus: Linear untyped segmentation

Segmentation: Agreement

Percent agreement — ratio between observed agreements and possible agreements



A	B	C
−	−	−
−	−	−
+	−	−
−	+	+
−	−	−
+	+	+
−	−	−
−	−	−

$$\frac{22}{8 * 3} = 91\%$$

Results on Agreement

Grosz&Hirschbergberg'92	newspaper text	74-95%
Hearst'93	expository text	80%
Passanneau&Litman'93	monologues	82-92%

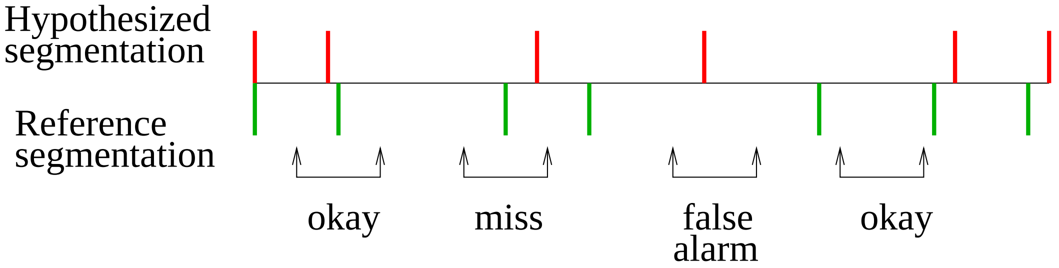
Kappa Statistics

(Siegal&Castellan, 1998; Carletta, 1999)
Kappa controls agreement $P(A)$ for chance agreement $P(E)$

$$K = \frac{P(A) - p(E)}{1 - p(E)}$$

	A	B	C
▬	−	−	−
▬	−	−	−
	+	−	−
▬	−	+	+
▬	−	−	−
	+	+	+
▬	−	−	−
▬	−	−	−

Evaluation Metric: P_k Measure



P_k: Probability that a randomly chosen pair of words k words apart is inconsistently classified (Beeferman '99)

- Set k to half of average segment length
- At each location, determine whether the two ends of the probe are in the same or different location. Increase a counter if the algorithm's segmentation disagree
- Normalize the count between 0 and 1 based on the number of measurements taken

Notes on P_k measure

- $P_k \in [0, 1]$, the lower the better
- Random segmentation: $P_k \approx 0.5$
- On synthetic corpus: $P_k \in [0.05, 0.2]$
- Beeferman reports 0.19 P_k on WSJ, 0.13 on Broadcast News

Cohesion

Key hypothesis: cohesion ties reflect text structure
Cohesion captures devices that link sentences into a text
(Halliday&Hasan)

- Lexical cohesion
- References
- Ellipsis
- Conjunctions

Example

Table removed for copyright reasons.

Please see: Figure 2 in Hearst, M. "Multi-Paragraph Segmentation of Expository Text." *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL 94)*, June 1994. (http://www.sims.berkeley.edu/~hearst/papers/tiling-acl94/acl94.html)

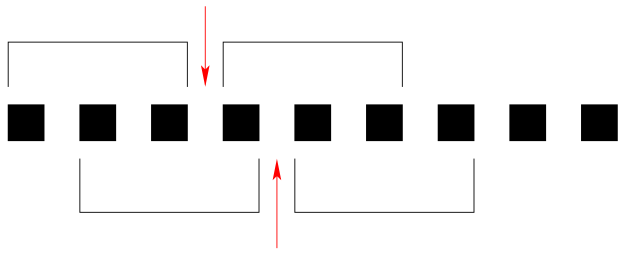
Example

Stargazers Text(from Hearst, 1994)

- Intro - the search for life in space
- The moon's chemical composition
- How early proximity of the moon shaped it
- How the moon helped life evolve on earth
- Improbability of the earth-moon system

Segmentation Algorithm of Hearst

- Preprocessing and Initial segmentation
- Similarity Computation
- Boundary Detection



Similarity Computation: Representation

Vector-Space Representation

SENTENCE$_1$: I like apples

SENTENCE$_2$: Apples are good for you

Vocabulary	Apples	Are	For	Good	I	Like	you
Sentence$_1$	1	0	0	0	1	1	0
Sentence$_2$	1	1	1	1	0	0	1

Similarity Computation: Cosine Measure

Cosine of angle between two vectors in n-dimensional space

$$sim(b_1, b_2) = \frac{\sum_t w_{y,b_1} w_{t,b_2}}{\sqrt{\sum_t w_{t,b_1}^2 \sum_{t=1}^n w_{t,b_2}^2}}$$

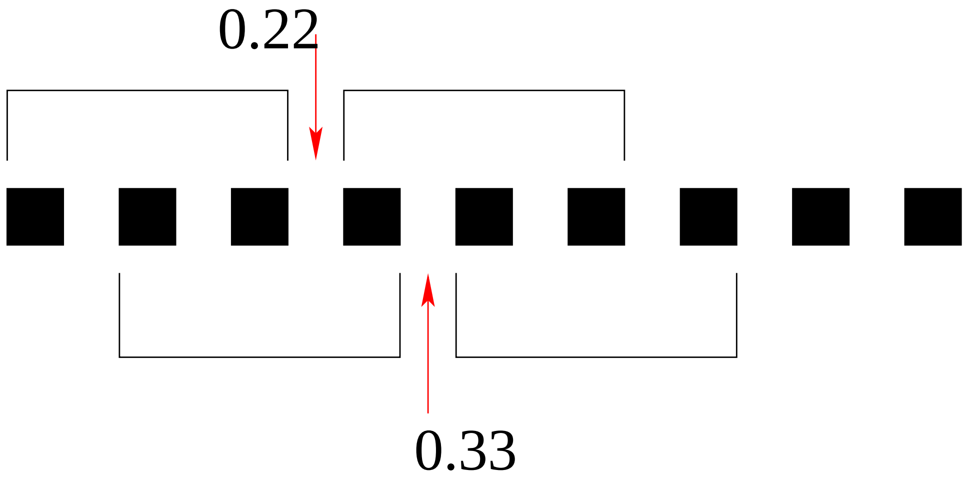
SENTENCE$_1$: 1 0 0 0 1 1 0

SENTENCE$_2$: 1 1 1 1 0 0 1

sim(S$_1$,S$_2$) =

$$\frac{1*0+0*1+0*1+0*1+1*0+1*0+0*1}{\sqrt{(1^2+0^2+0^2+0^2+1^2+1^2+0^2)*(1^2+1^2+1^2+1^2+0^2+0^2+1^2}} = 0.26$$

Similarity Computation: Output



Gap Plot

Figure of Gap Plot removed for copyright reasons.

Boundary Detection

Based on changes in sequence of similarity scores:

Depth Scores: relative depth (in comparison to the closest maximum)

Number of segments: $s - \sigma/2$

Segmentation Evaluation

Comparison with human-annotated segments(Hearst'94):

- 13 articles (1800 and 2500 words)
- 7 judges
- boundary if three judges agree on the same segmentation point

Agreement on Segmentation

Figure removed for copyright reasons.

Please see: Figure 3 in Hearst, M. "Multi-Paragraph Segmentation of Expository Text." *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL 94)*, June 1994. (http://www.sims.berkeley.edu/~hearst/papers/tiling-acl94/acl94.html)

Evaluation Results

Methods	Precision	Recall
Baseline 33%	0.44	0.37
Baseline 41%	0.43	0.42
Chains	0.64	0.58
Blocks	0.66	0.61
Judges	0.81	0.71

More Results

- High sensitivity to change in parameter values
- Thesaural information does not help
- Most of the mistakes are “close misses”

Summary

- Types of discourse models:
  - Cohesion-based
  - Content-based
  - Rhetorical
- Segmentation
  - Agreement and Evaluation
  - Hearst's segmentation algorithm