Introduction to Topic Models

Eugene Weinstein
October 21st, 2008
Machine Learning Seminar
Computer Science Department
Courant Institute of Mathematical Sciences
New York University

Topic Analysis

- ...this protest has brought out thousands of serbs calling for the end of the milosevic regime. opposition leaders are confident milosevic's days in power are numbered. on capitol hill tonight the senate approved 600,000 visas for skilled high technology workers...
- Topic analysis: label text or speech stream with topic boundaries and/or identities.

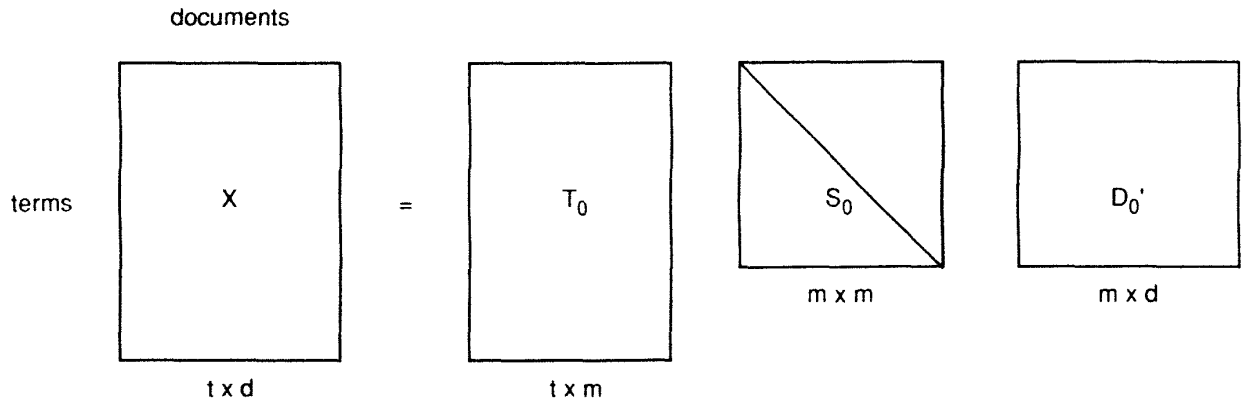
Topic Modeling

- Idea: find low-dimensional descriptions of high-dimensional text
- Topic models enable spoken/text document
  - Summarization - finding concise restatements
  - Similarity - evaluating closeness of texts
- This can help improve, e.g.,
  - Navigation quality of speech/text collections
  - Speech recognition quality (with topic-specific models)

Latent Semantic Analysis

- Idea: text is explained by mixing latent topics/factors
  - Topic models try to discover this underlying structure
- Latent Semantic Analysis/Indexing [Deerwester et al. '90]
  - Measure occurrence frequency of terms in documents
  - Write frequencies as term-document matrix
  - Analyze using Singular Value Decomposition (SVD)
  - Components: term-topic, topic-topic, and document-term matrices

Applying SVD



T_0 has orthogonal, unit-length columns ($T_0' T_0 = I$)
D_0 has orthogonal, unit-length columns ($D_0' D_0 = I$)
S_0 is the diagonal matrix of singular values
t is the number of rows of X
d is the number of columns of X
m is the rank of X ($\leq$ min(t,d))

- Discover latent factors with approximate SVD (keep k highest singular values). $X \approx \hat{X} = TSD'$

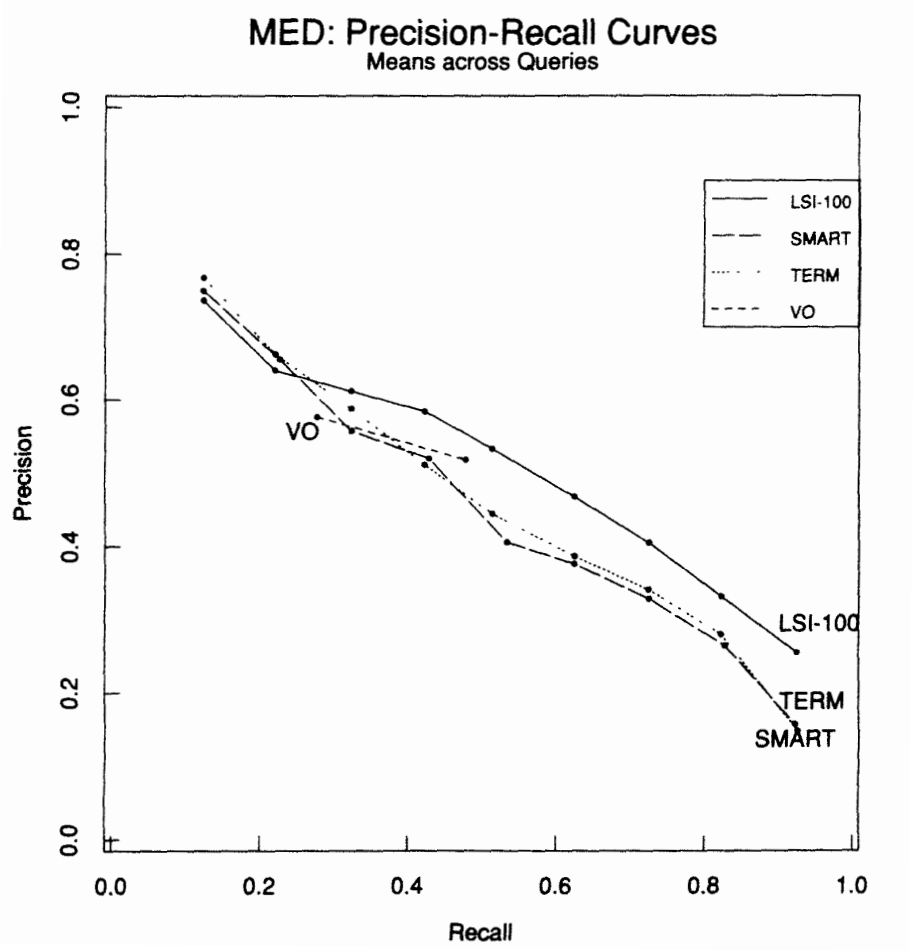
Using LSA

- Approximate SVD formulation of LSA: $X \approx \hat{X} = TSD'$
- Term similarity matrix: $\hat{X}\hat{X}' = TS^2T'$
- Document similarity matrix: $\hat{X}'\hat{X} = DS^2D'$
- Factored document-term matrix $\hat{X}$ is used for indexing
- Retrieval: compute cosines between query vector, $\hat{X}$
  - Apply threshold (determines operating point)
- Medical abstracts database: 1033 documents, 30 queries
- Compare to two state-of-the art IR systems

Indexing Results

- Good improvement on MED dataset
  - But, some concern that this is artificial
- Mixed results on other corpora
- Summary: LSA is not explicitly a topic model but is the foundation for much later work

	Term and LSI	Voorhees	SMART
Number of unique terms	5823	6927	6927
Mean number of terms per document	50.1	51.6	51.6
Mean number of terms per query	9.8	39.7[7]	10.1
Mean number of relevant documents per query	23.2	23.2	23.2



Probabilistic Topic Models

- LSA: try to discover fixed factors underlying the text
- Document-term $\hat{X} = TSD'$, singular values s_k, score:

$$M(w,d) = [\hat{X}]_{w,d} = \sum_k [D']_{k,d}[T]_{w,k}s_k$$

- Want model with solid statistical foundation
  - Based on likelihood principle, defines generative model
- PLSA [Hoffman '99]: learn a set of models for hidden topics $z_1, \ldots, z_K$

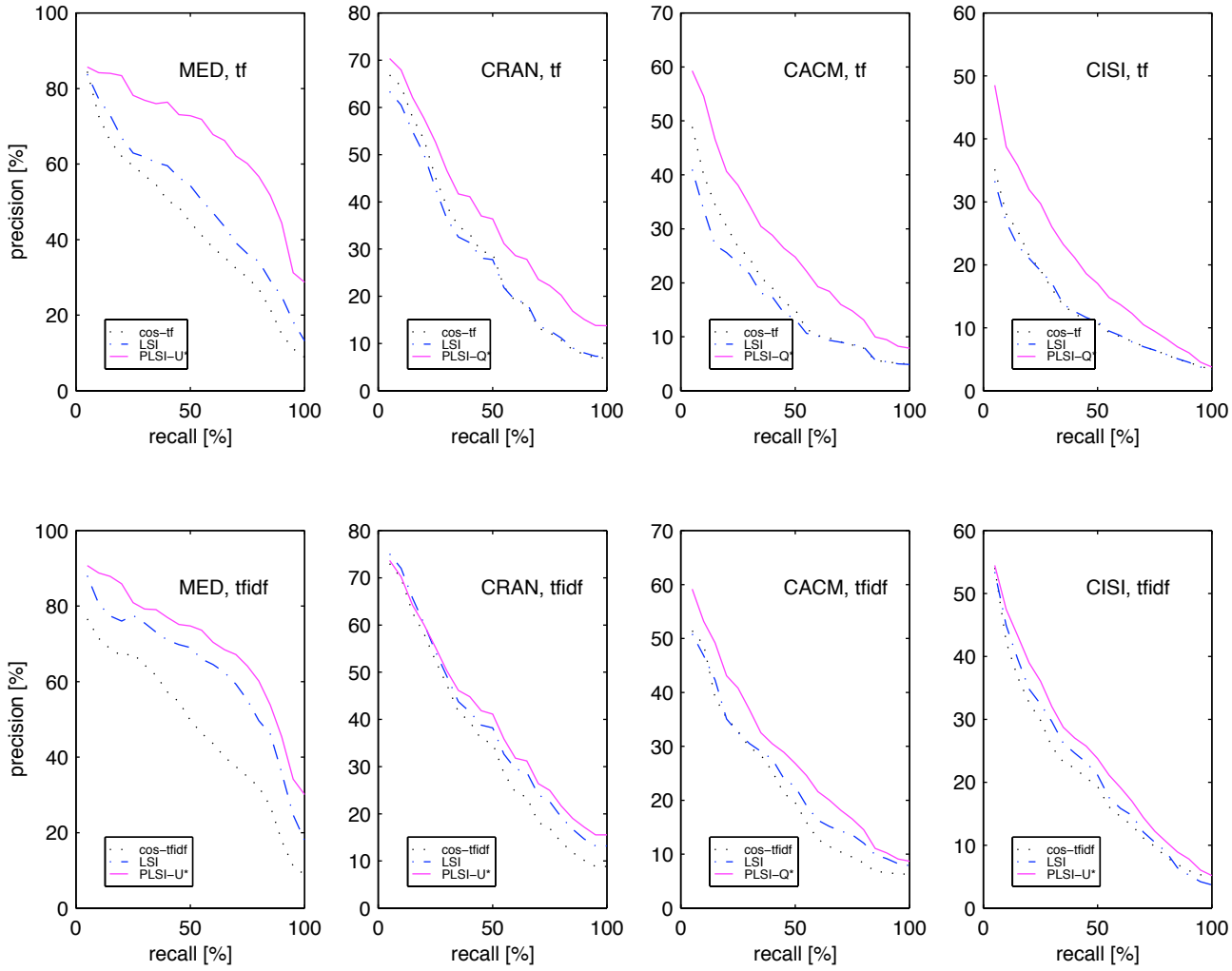
$$\Pr(w,d) = \sum_k \Pr(d|z_k)\Pr(w|z_k)\Pr(z_k)$$

PLSA

- Generative process:
  - select a document d with probability $P(d)$,
  - pick a latent class z with probability $P(z|d)$,
  - generate a word w with probability $P(w|z)$.
- Testing scenario: indexing task as with LSA
  - Given text w, calculate $\Pr(w, d)$, apply threshold

PLSA Results

- Compare LSI, PLSI, plain cosine score
- Compare term-frequency (tf) by itself and weighted by inverse document frequency (idf)



Moving Away from LSA

- Want to model topic stream underlying arbitrary text
  - So, move away from explicitly modeling documents
  - Thus, the indexing task is no longer representative
- Simple generative model:

$$\Pr(w) = \sum_k \Pr(w|z_k)\Pr(z_k)$$

- Generative interpretation: pick topic z_k from prior distribution
  - Pick words according to distribution of topic z_k

Generic Topic Models

- Any text can be explained by any topic sequence with some likelihood
- We can find the maximum *a posteriori* topic for a text

$$k = \arg\max_k \Pr(z_k|w) = \arg\max_k \Pr(w|z_k)\Pr(z_k)$$

- An aside: what do we mean by "text" w?
  - It's a generic bag-of-words, could be
    - Single word
    - Sentence
    - Speech utterance

Model Specifics

- PLSA: $\Pr(w, d) = \sum_k \Pr(d|z_k)\Pr(w|z_k)\Pr(z_k)$
- Generic topic models $\Pr(w) = \sum_k \Pr(w|z_k)\Pr(z_k)$
- What kind of distributions to use?
- Simple choice: unigram/Naïve Bayes: $\Pr(x) = \text{Count}(x)$
  - e.g., $\Pr(w|z_k) = \phi_{w,k} =$ normalized number of times topic z_k is assigned to text w in the training data
- Can do smoothing here, many options, e.g., add-one:

$$\phi_{w,k} = \frac{\text{Count}(w, z_k) + 1}{N + M} \quad N = \sum_{w,k} \text{Count}(w, z_k) \quad M = \sum_{w,k} 1$$

Other Distributions

- Unigram is a simple distribution; smoothing is a rudimentary approximation for allowing unseen data
- More sophisticated: multinomial with Dirichlet prior
- Simple example: binomial; how many 6s in 10 die tosses?
- $\Pr(K = k) = \binom{n}{k} p^k (1-p)^{n-k}$, $p = \frac{1}{6}$, $n = 10$
- Multinomial: generalization of this

$$f(x_1, \ldots, x_k; n, p_1, \ldots, p_k) = \Pr(X_1 = x_1 \text{ and } \ldots \text{ and } X_k = x_k)$$

$$= \begin{cases} \dfrac{n!}{x_1! \cdots x_k!} p_1^{x_1} \cdots p_k^{x_k}, & \text{when } \sum_{i=1}^{k} x_i = n \\ 0 & \text{otherwise,} \end{cases}$$

Dirichlet Details

- k-dimensional Dirichlet random variable

$$p(\theta|\alpha) = \frac{\Gamma\left(\sum_{i=1}^{k}\alpha_i\right)}{\prod_{i=1}^{k}\Gamma(\alpha_i)}\theta_1^{\alpha_1-1}\cdots\theta_k^{\alpha_k-1}$$

- Real-number generalization of the factorial:
  - $\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}\,dt$, for integers $\Gamma(n) = (n-1)!$
- α is the parameter vector
- takes values in the $(k-1)$-simplex (i.e, sums to 1)
- Dirichlet: conjugate prior distribution to the multinomial
  - Prior $\Pr(\theta)$ conjugate to likelihood function class $\Pr(x|\theta)$ if posterior likelihood $\Pr(\theta|x)$ in the same family as $\Pr(\theta)$

Latent Dirichlet Allocation

- LDA [Blei, Jordan, 2003]: $P(z_k)$ modeled with multinomial distribution, Dirichlet prior
- Generative process:
  - Choose multinomial parameters $\theta \sim \mathrm{Dirichlet}(\alpha)$
  - Choose a topic $z_k \sim \mathrm{Multinomial}(\theta)$
    - Choose a text $w \sim \mathrm{Multinomial}(\phi_k)$
- Inference: decode maximum *a posteriori* sequence of topic labels accounting for sentence

$$k = \arg\max_k \Pr(z|w) = \arg\max_k \Pr(w|z_k)\Pr(z_k)$$

Learning

- For unigram models, can optimize directly with EM
- Optimizing all LDA parameters is intractable
- Variational inference
  - Remove part of the conditionality of generative model
  - Replace with free variational parameters, optimize
  - Find true distribution closest (in KL-divergence) to variational distribution
- Another possibility: sampling methods (MCMC)

LDA Results: Perplexity

- Evaluate perplexity (roughly, 10% held-out data likelihood)
  - Single Unigram
  - Mixture of unigrams

$$\Pr(w) = \sum_k \Pr(w|z_k)\Pr(z_k)$$

  - PLSA
- Nematode: 5,255 biology abstracts
- AP: 16,333 newswire articles

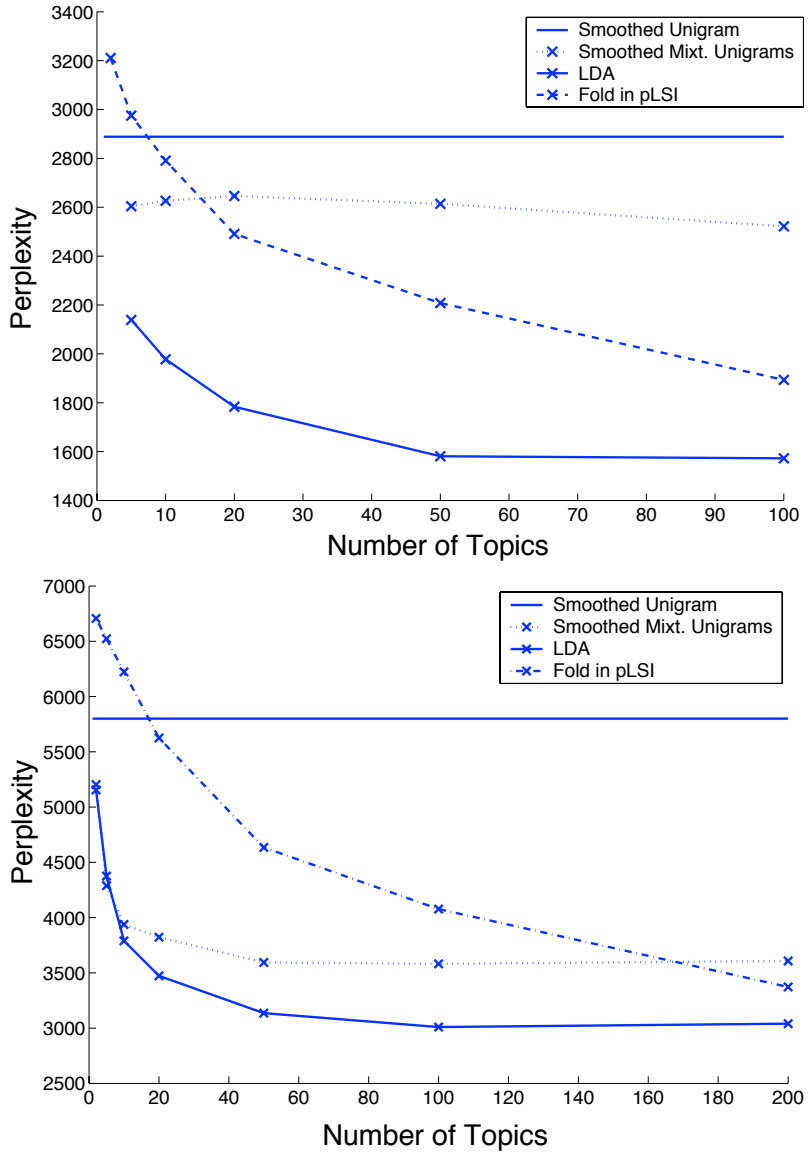
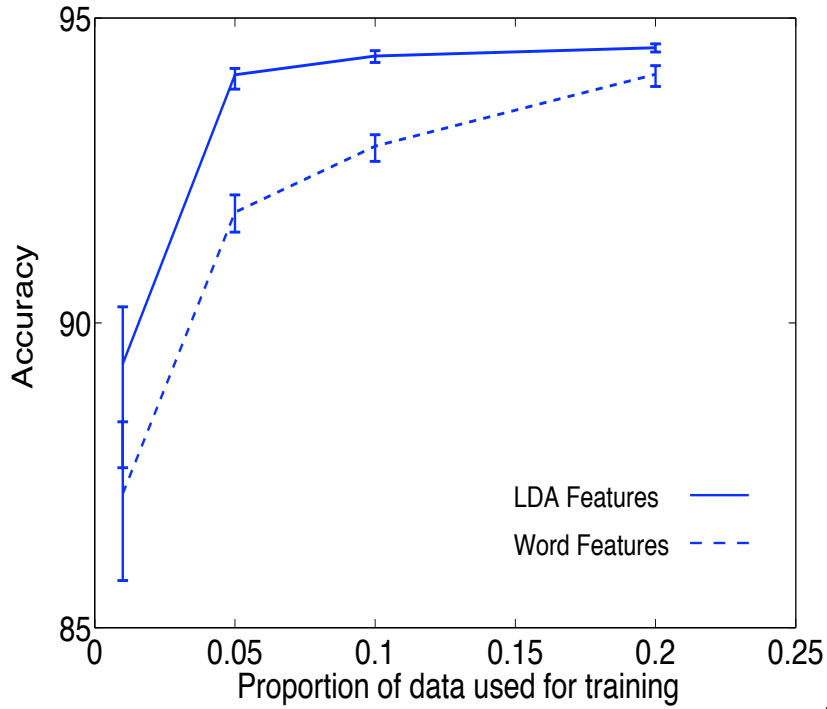
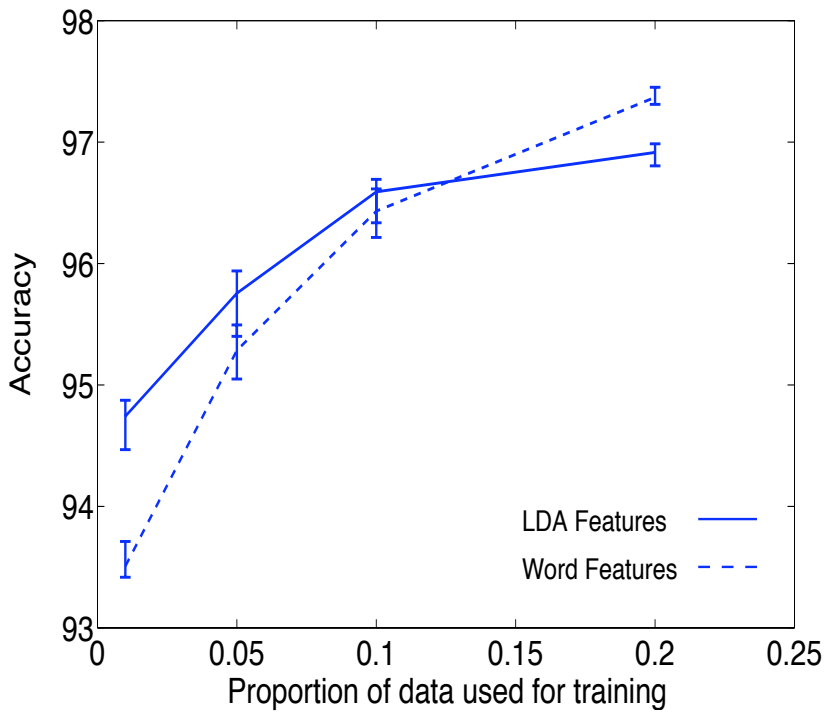


Figure 9: Perplexity results on the nematode (Top) and AP (Bottom) corpora for LDA, the unigram

LDA Results: Classification

- Train binary SVM classifier on (1) LDA posteriors; (2) the words themselves
- Reuters-21578: 8000 documents labeled with classes





Context Dependency

- All the models so far labels bags-of-words in isolation
- We want to model the transition from topic to topic
- Typical approach: embed topic model in HMM
  - i.e., assign a penalty to changing topics
- [Yamron et al. '97]: hand-tuned penalty to move between unigram topic models
- [Blei+Moreno, '01]: add HMM structure to PLSA model
- [Gruber et al., '06]: add HMM structure to LDA; HMM transitions learned at the same time as LDA parameters

Topic Segmentation

- Breaking up a stream of text or speech into topic-coherent segments is of independent interest
  - e.g., for presentation of indexed audio collections, etc.
- A topic model such as e.g., PLSA, LDA, HTMM implicitly gives topic segmentation
- But what if only the correct segmentation matters?
  - Can we give algorithms directly focused on segmentation?
  - How do we evaluate their performance?

TextTiling

- [Hearst '94] Split text into windows of size k, for all pairs of adjacent windows (gaps)
  - Calculate cosine measure $sim(b_1, b_2) = \frac{\sum_t w_{t,b_1} w_{t,b_2}}{\sqrt{\sum_t w_{t,b_1}^2 \sum_{t=1}^{n} w_{t,b_2}^2}}$
  - Smooth with average smoothing
- Segmentation is based on heuristics
  - Measure peak-to-trough differences in cosine signal
  - Hypothesize boundaries when difference above cutoff
  - Cutoff heuristic: $\Delta > \bar{s} - \sigma/2$

TextTiling Results

- Stargazers text: ~400 paragraphs
- Compare TextTiling to
  - Sanity-check segmentations (33, 41% of paragraph gaps)
  - Human segmentations

	Precision		Recall	
	$\bar{s}$	σ	$\bar{s}$	σ
Baseline 33%	.44	.08	.37	.04
Baseline 41%	.43	.08	.42	.03
Chains	.64	.17	.58	.17
Blocks	.66	.18	.61	.13
Judges	.81	.06	.71	.06

Topic Segmentation Models

- [Beeferman et al. '99]: learn model $q(b|X), b \in \{\text{YES}, \text{NO}\}$
- Exponential linear model form $q(\text{YES} \mid X) = \frac{1}{Z_\lambda(X)} e^{\lambda \cdot f(X)}$
- Look for model minimizing the KL-divergence to the empirical distribution $\tilde{p}(b \mid X)$

$$D(\tilde{p} \,\|\, q) = \sum_X \tilde{p}(X) \sum_{b \in \{\text{YES,NO}\}} \tilde{p}(b \mid X) \log \frac{\tilde{p}(b \mid X)}{q(b \mid X)} .$$

- Use iterative scaling algorithm to learn parameters λ
- Greedy feature selection algorithm
  - Iteratively add feature most improving objective

Segmentation Features

- Model: combination of trigram and salient word pairs

$$p_{\text{exp}}(w \mid X) = \frac{1}{Z_\lambda(X)} e^{\lambda \cdot f(w,X)} \, p_{\text{tri}}(w \mid w_{-2}, w_{-1})$$

$$Z_\lambda(X) = \sum_{w \in \mathcal{W}} e^{\lambda \cdot f(w,X)} \, p_{\text{tri}}(w \mid w_{-2}, w_{-1}) \, .$$

- X is the total history, $f(w, X)$ tests for trigger word pairs
- Final features are topicality

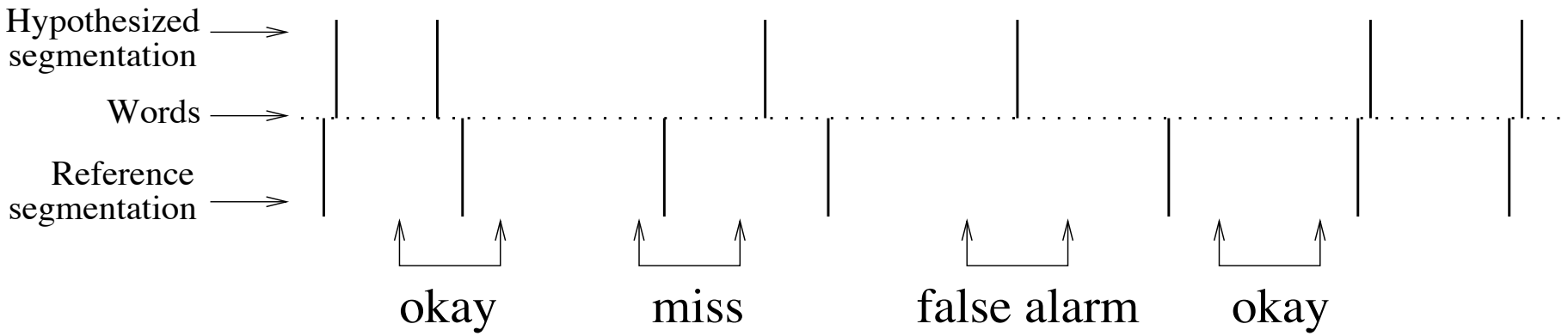
$$T(w, X) \equiv \log \frac{p_{\text{exp}}(w \mid X)}{p_{\text{tri}}(w \mid w_{-2}, w_{-1})}$$

- ... and cue word features

s, t	$e^{\lambda_{s,t}}$
RESIDUES, CARCINOGENS	2.3
CHARLESTON, SHIPYARDS	4.0
MICROSCOPIC, CUTICLE	4.1
DEFENSE, DEFENSE	8.4
TAX, TAX	10.5
KURDS, ANKARA	14.8
VLADIMIR, GENNADY	19.6
STEVE, STEVE	20.7
EDUCATION, EDUCATION	22.2
INSURANCE, INSURANCE	23.0
PULITZER, PRIZEWINNING	23.6
YELTSIN, YELTSIN	23.7
SAUCE, TEASPOON	27.1
FLOWER, PETALS	32.3
PICKET, SCAB	103.1

Scoring

- CoAP scoring [Beeferman et al. '99]
  - Move sliding window across text, measure fraction of agreements between reference and hypothesis



$$p(\text{error} \mid \texttt{ref}, \texttt{hyp}, k) =$$
$$p(\text{miss} \mid \texttt{ref}, \texttt{hyp}, \text{different } \texttt{ref} \text{ segments}, k)\, p(\text{different } \texttt{ref} \text{ segments} \mid \texttt{ref}, k)$$
$$+ p(\text{false alarm} \mid \texttt{ref}, \texttt{hyp}, \text{same } \texttt{ref} \text{ segment}, k)\, p(\text{same } \texttt{ref} \text{ segment} \mid \texttt{ref}, k)$$

Experiments

- Train decision trees, interpolate with regular exponential model
  - Compare to HMMs of [Yamron et al. '97]
  - TextTiling

1. Training/test: 2M/1M words, CNN transcripts
2. Training/test: 1M/325K WSJ articles

segmentation model	P_k	*miss probability*	*false alarm probability*
exponential model	9.5%	12.1%	6.8%
decision tree	11.3%	16%	6.6%
hidden Markov model	16.7%	16%	17.6%
interpolated (exp + dtree) models	7.8%	7.2%	8.4%
random boundaries	49.5%	60.1%	38.9%
all boundaries	47.5%	0%	100%
no boundaries	49.7%	100%	0%
evenly spaced	50.3%	50.3%	50.3%

segmentation model	P_k	*miss probability*	*false alarm probability*
exponential model	19.0%	24.0%	15.75%
decision tree	24.6%	32.7%	19.4%
interpolated (exp + dtree) models	18.5%	24.7%	14.5%
cue-word features only	23.7%	35.6%	15.8%
topicality features only	35.8%	45.4%	29.6%
TextTiling	29.6%	45.7%	19.1%

References

- Deerwester, Scott, Indexing by Latent Semantic Analysis , American Society for Information Science, Journal, 41:6, 1990.
- Marti A. Hearst, Multi-Paragraph Segmentation of Expository Text. Proceedings of the 32nd Meeting of the Association for Computational Linguistics, Los Cruces, NM, June, 1994.
- J.P. Yamron, I. Carp, L. Gillick, S. Lowe, and P. van Mulbregt, Event tracking and text segmentation via hidden markov models. ASRU, 1998.
- Thomas Hofmann. Probabilistic latent semantic indexing. SIGIR, Berkeley, California, 1999.
- Doug Beeferman, Adam Berger, and John Lafferty. Statistical models for text segmentation. Machine Learning}, 1999, pp. 177-210.
- David M. Blei and Pedro J. Moreno. Topic segmentation with an aspect hidden Markov model. SIGIR Conference on Research and Development in Information Retrieval. 2001, pp. 343-348, ACM Press.
- David M. Blei, Andrew Y. Ng, Michael I. Jordan, and John Lafferty. Latent dirichlet allocation. JMLR, vol. 3, 2003.
- Amit Gruber, Michal Rosen-Zvi, and Yair Weiss. Hidden topic markov models. AISTATS, San Juan, Puerto Rico, 2007.