OVERVIEW OF STATISTICAL NATURAL LANGUAGE PROCESSING

Fei Song, School of Computer Science

University of Guelph

Monday, Sept. 17, 2012

Outline

- Statistical Natural Language Processing (SNLP)
- Language Models
- Noisy Channel Framework
- Hidden Markov Models (HMM’s)
- Text Classification and Sentiment Analysis
- Topic Models (Latent Dirichlet Allocation)
- References

What is SNLP?

- Using statistical techniques to infer structures from text based on statistical language modeling:
  - Probability and Statistics
  - Information Theory
  - Computational Linguistics
- Wide applications: Information Retrieval, Information Extraction, Text Classification, Text Mining, and Biological Data Analysis.

Brief History

- Started in late 1950's and early 1960's:
  - Bayesian model for optical character recognition
  - Brown corpus of American English: 1 million word collection of 500 texts from different genres.
- Hidden Markov models for speech recognition (1970 to 1983): early success.
- Dominance of Empiricism and Statistical Methods (1983-present):
  - Incorporate probabilities for most language processing
  - Use large corpora for training and evaluation

Word Statistics in Tom Sawyer

- The text has less than half a megabyte of data.

Word Frequency	Frequency of Frequencies
1	3993
2	1292
3	664
4	410
5	243
6	199
7	172
8	131
9	82
10	91
11-50	540
50-100	99
>100	102

Word Statistics in Tom Sawyer

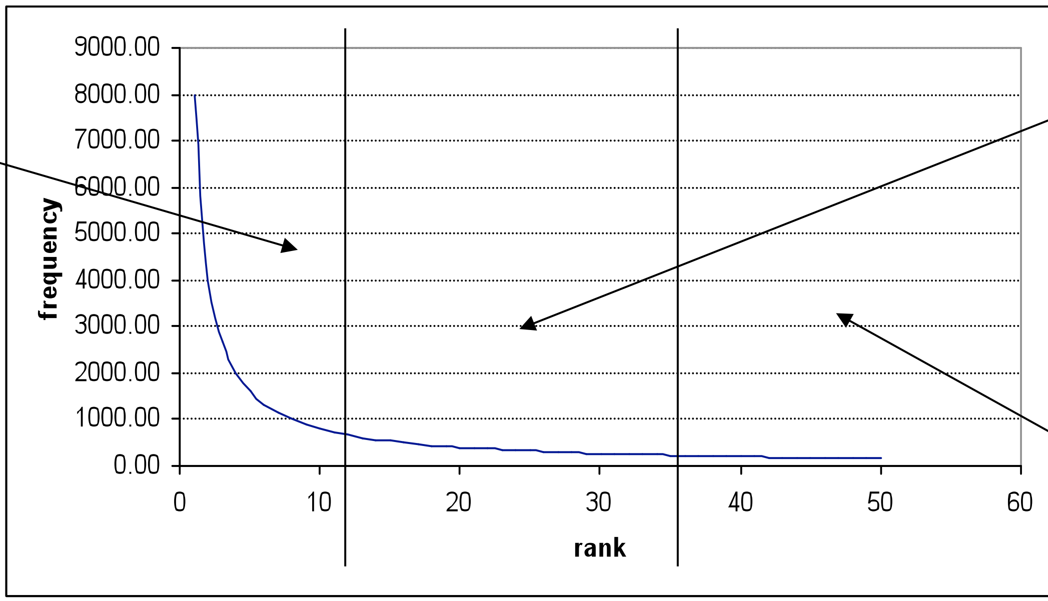
- Average word frequency: 8.9 (tokens/type)
- The most common 100 words account for 50.9% of the tokens in the text.
- 49.8% of the word types occur only once in the text.
  - Over 90% of the word types occur 10 times or less.
  - Over 12% of text made of words that occur 3 times or less.

Zipf's Law

- Given the frequency f of a word and its rank r in the list of words ordered by their frequencies:

$$f \propto 1/r \qquad \text{or} \qquad f \times r = k \text{ for a constant } k$$

A small number of common words



A reasonable number of medium-freq words

A large number of rare words

Language Modeling

- Chain rule:

  $P(w_1\ w_2\ \ldots\ w_n) = P(w_1)\ P(w_2|w_1)\ \ldots\ P(w_n|w_1\ w_2\ \ldots\ w_{n-1})$

  e.g., Jack went to the {hospital, number, if, … }

- Predict the next word given the previous words.
  - Shannon's game: guess the next letter in a text
- Left-context only?
  - The {big, pig} dog …
  - P(dog|the big) >> P(dog|the pig)

N-Gram Models

- Markov assumption: the probability of the next word depends only on the previous k words.

$$P(w_n | w_1 \ldots w_{n-1}) = P(w_n | w_{n-k}\ w_{n-k+1} \ldots w_{n-1})$$

(k+1)-gram or K^{th} order Markov approximation

- Common N-grams:
  - Unigram: $P(w_1\ w_2 \ldots w_n) = P(w_1)\ P(w_2) \ldots P(w_n)$
  - Bigram: $P(w_1\ w_2 \ldots w_n) = P(w_1)\ P(w_2 | w_1) \ldots P(w_n | w_{n-1})$
  - Trigram: $P(w_1\ w_2 \ldots w_n) = P(w_1)\ P(w_2 | w_1) \ldots P(w_n | w_{n-2}\ w_{n-1})$

Number of Parameters

- The larger the n, the more the number of parameters to estimate:

Models	Parameters
Unigram Bigram Trigram Four-gram	20,000 $20{,}000^2 = 400$ millions $20{,}000^3 = 8$ trillions $20{,}000^4 = 1.6 \times 10^{17}$

Maximum Likelihood Estimation (MLE)

- Given $C(w_1 w_2 \ldots w_n)$ as the frequency of $w_1 w_2 \ldots w_n$ in the training set and N as the total number of n-grams in the training set:

$$P_{MLE}(w_1 w_2 \ldots {}_{wn}) = C(w_1 w_2 \ldots w_n)/N$$

$$P_{MLE}(w_n | w_1 w_2 \ldots w_{n-1}) = C(w_1 w_2 \ldots w_n)/C(w_1 w_2 \ldots w_{n-1})$$

- Given the two words "come across", we may have:
  - P(as|come across) = 0.8, P(more|come across) = 0.1, P(a|come across) = 0.1, and P(x|come across) = 0 for any other word x.

Sparse Data Problem

- With MLE, a missing k-gram means zero probability and any longer n-gram that contains the k-gram will also have a zero probability.
- Zipf's law: no matter how big is a training set, there will always be a lot of rare events that may not be covered.
- Discounting/smoothing techniques: systematically allocate some probability mass for the missing n-grams.

Laplace's Law

- Adding one count for every bin:

  $$P_{LAP}(w_1 w_2 \ldots w_n) = [C(w_1 w_2 \ldots w_n) + 1] / (N + B)$$

  B --- the number of bins in a partition.

- Problem?
  - Give far too much probability mass to unseen events for a partition with a large vocabulary.

- Estimated frequency: $f_{LAP} = [C(w_1 w_2 \ldots w_n) + 1] \times N / (N + B)$

Estimated Frequencies for AP Data

$r = f_{MLE}$	$f_{empirical}$	f_{LAP}	f_{del}	f_{GT}
0	0.000027	0.000295	0.000037	0.000027
1	0.448	0.000589	0.396	0.446
2	1.25	0.000884	1.24	1.26
3	2.24	0.00118	2.23	2.24
4	3.23	0.00147	3.22	3.24
5	4.21	0.00177	4.22	4.22
6	5.23	0.00206	5.20	5.19
7	6.21	0.00236	6.21	6.21
8	7.21	0.00265	7.18	7.24
9	8.26	0.00295	8.18	8.25

Linear Interpolation

- Enhanced trigram model:

$$P_{li}(w_n | w_{n-2}, w_{n-1}) = \lambda_1 P_1(w_n) + \lambda_2 P_2(w_n | w_{n-1}) + \lambda_3 P_3(w_n | w_{n-2}, w_{n-1})$$

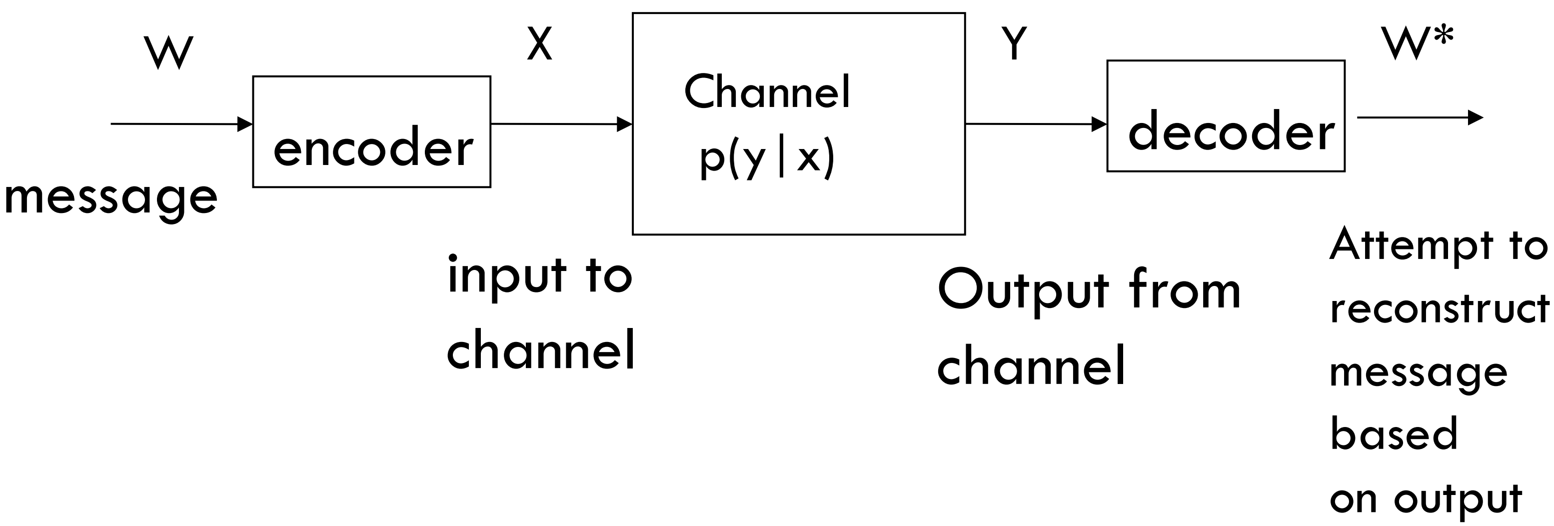
$$\text{where } 0 \leq \lambda_i \leq 1 \text{ and } \Sigma\lambda_i = 1$$

- In fact, $P_1(w_n)$ and $P_2(w_n | w_{n-1})$ are already needed for the start of any longer sequence.

- The weights may be set by hand, or computed automatically by an application of the Expectation Maximization (EM) algorithm.

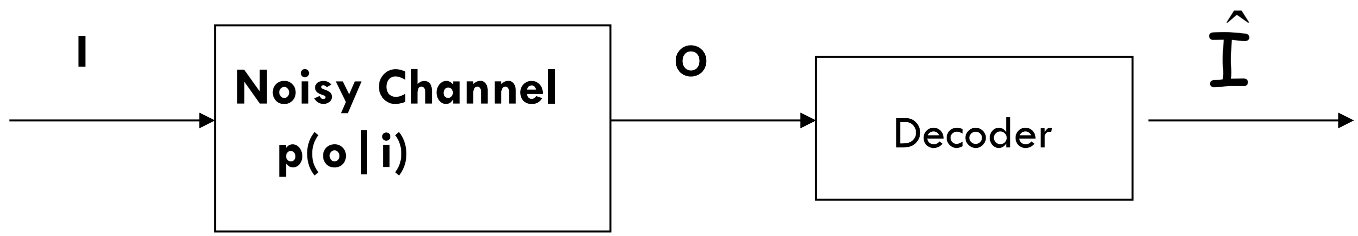
Noisy Channel Framework

- Goal: encode the message in such a way that it occupies minimal space while still containing enough redundancy to be able to detect and correct errors.



Noisy Channel Framework for SNLP

- In linguistics, we can't control the encoding phase, but we want to decode the output to give the most likely input.

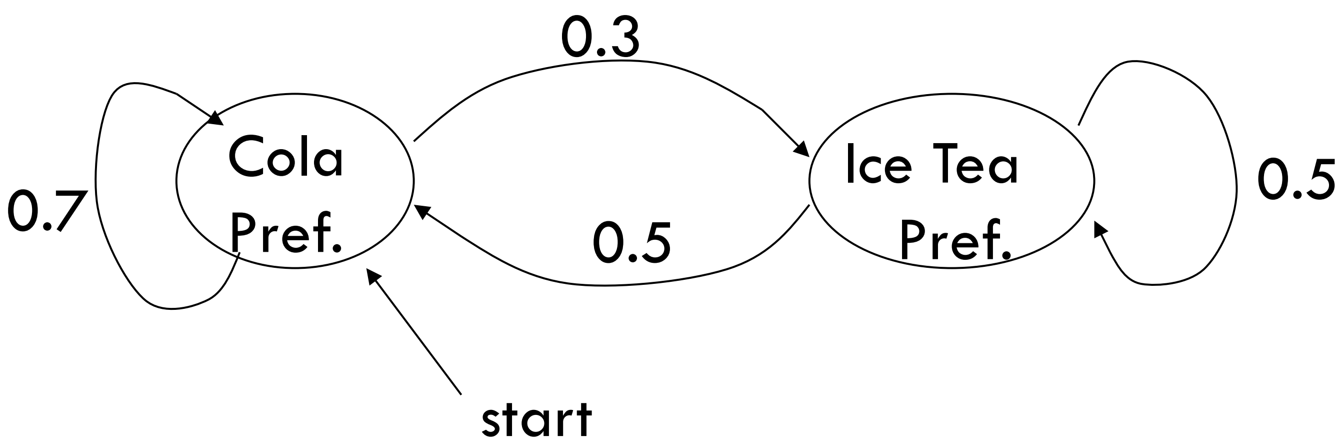


$$\hat{I} = \operatorname*{argmax}_i p(i|o) = \operatorname*{argmax}_i \frac{p(i)p(o|i)}{p(o)} = \operatorname*{argmax}_i p(i)p(o|i)$$

- Applications: machine translation, optical character recognition, speech recognition, spelling correction.

Crazy Soft Drink Machine

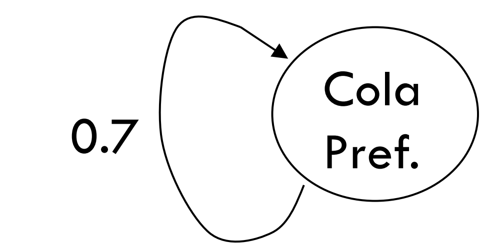
- Hidden Markov Model:

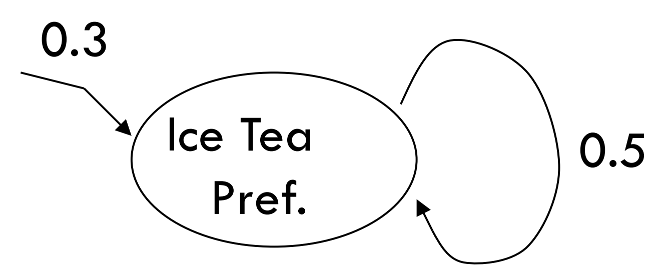


	Cola	Ice-Tea	Lemonade
CP	0.6	0.1	0.3
IP	0.1	0.7	0.2

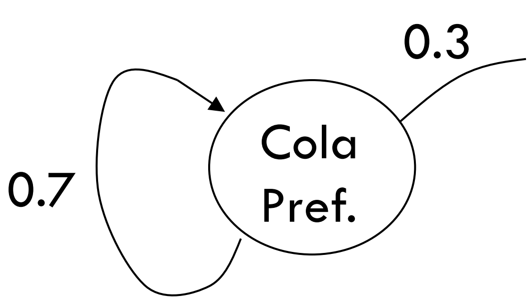
Crazy Soft Drink Machine

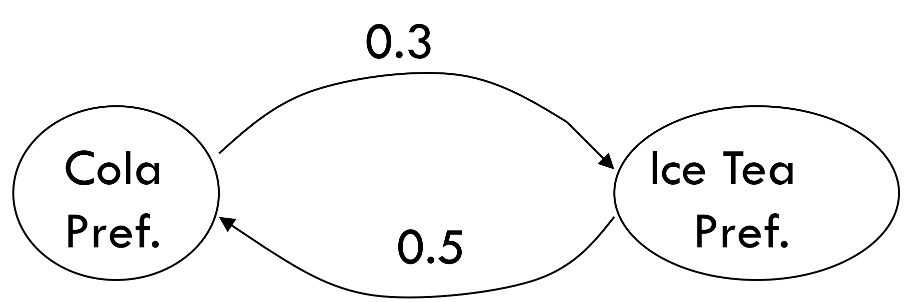
- What's the probability of seeing the output sequence {lemonade, ice-tea} if the machine always starts off in the cola preferring state?





$$
\begin{aligned}
&0.7 \times 0.3 \times 0.7 \times 0.1\ + \\
&0.7 \times 0.3 \times 0.3 \times 0.1\ + \\
&0.3 \times 0.3 \times 0.5 \times 0.7\ + \\
&0.3 \times 0.3 \times 0.5 \times 0.7 \\
&= 0.084
\end{aligned}
$$





Finding Probability of a Sequence

- Naïve solution: given the observation O = ($o_1,\ldots,o_T$) and a model μ = (A,B,Π):

$$P(O|\mu) = \sum_X P(O,X|\mu) = \sum_X P(X|\mu)P(O|X,\mu)$$

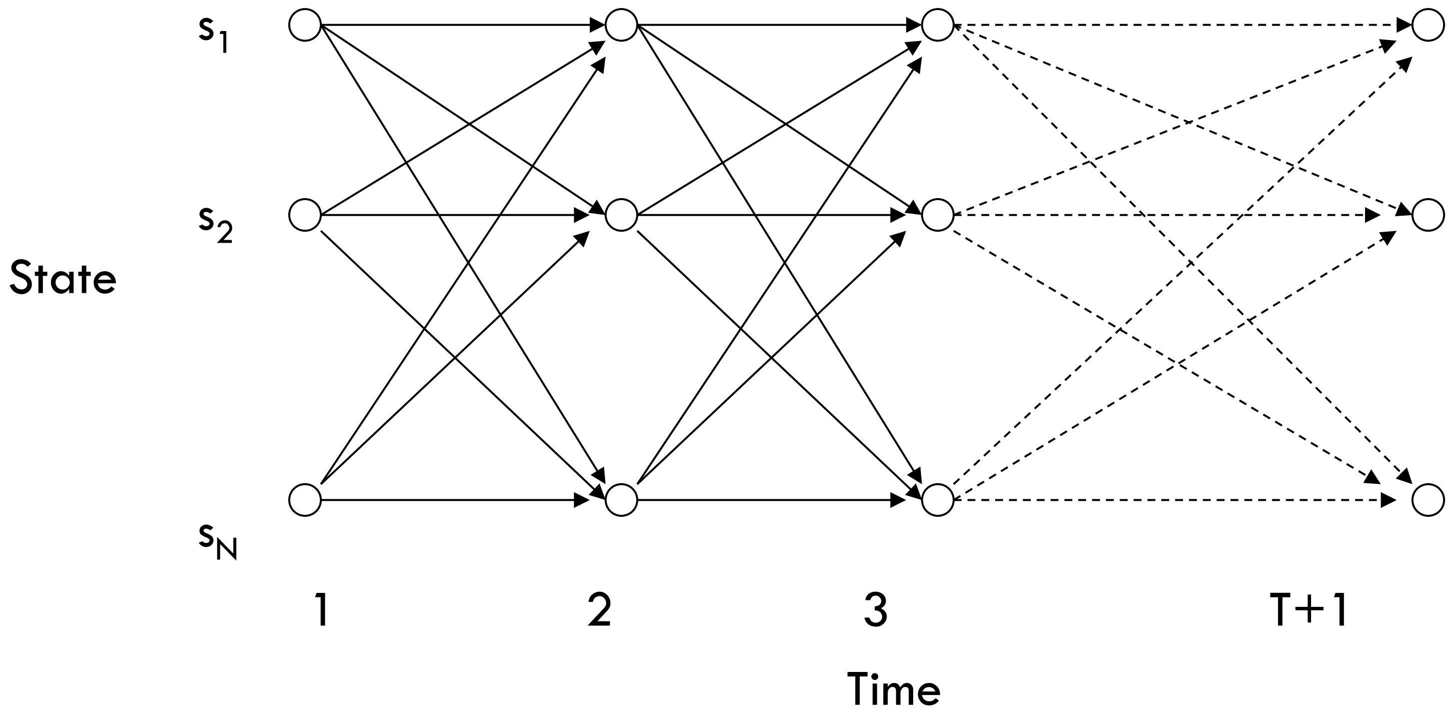
$$= \sum_{X_1 \ldots X_{T+1}} (\pi_{X_1} a_{X_1X_2} a_{X_2X_3} \ldots a_{X_TX_{T+1}})(b_{X_1X_2O_1} b_{X_2X_3O_2} \ldots b_{X_TX_{T+1}O_T})$$

$$= \sum_{X_1 \ldots X_{T+1}} \pi_{X_1} \prod_{t=1}^{T} a_{X_tX_{t+1}} b_{X_tX_{t+1}O_t}$$

- Complexity: require (2T+1) N^{T+1} multiplications.

Trellis/Lattice Structure

- Dynamic programming: reduce the complexity by memorizing partial results.



Forward Procedure

- Initialization:

$$\alpha_i(1) = \pi_i, \qquad 1 \le i \le N$$

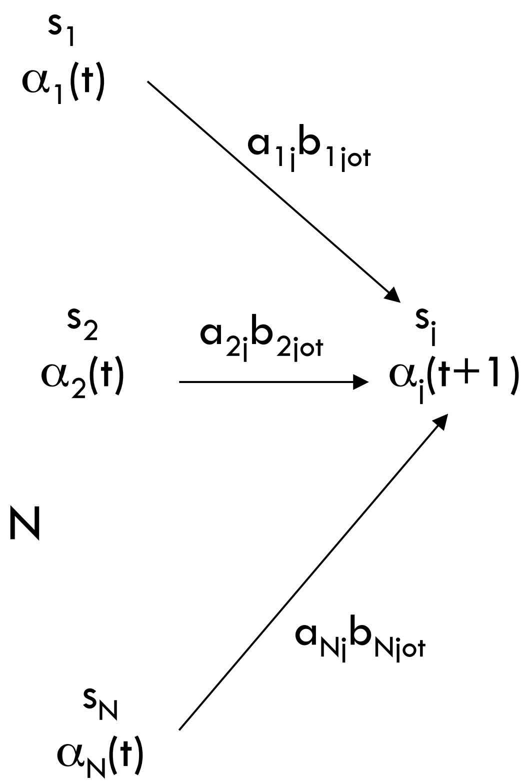
- Induction:

$$\alpha_j(t+1) = \sum_{i=1}^{N} \alpha_i(t) a_{ij} b_{ij o_t}, \quad 1 \le t \le T,\ 1 \le j \le N$$

- Total:

$$P(O \mid \mu) = \sum_{j=1}^{N} \alpha_j(T+1)$$

- Complexity: require $2N^2T$ multiplications.



Text Classifications/Categorizations

- Common classification problems:

Problems	Input	Categories
Tagging	context of a word	the word's tags
Disambiguation	context of a word	the word's senses
PP attachment	sentence	parse trees
Author identification	document	authors
Language identification	document	languages
Text categorization	document	topics

- Common classification methods: decision trees, maximum entropy modeling, neural networks, and clustering.

Text Categorization

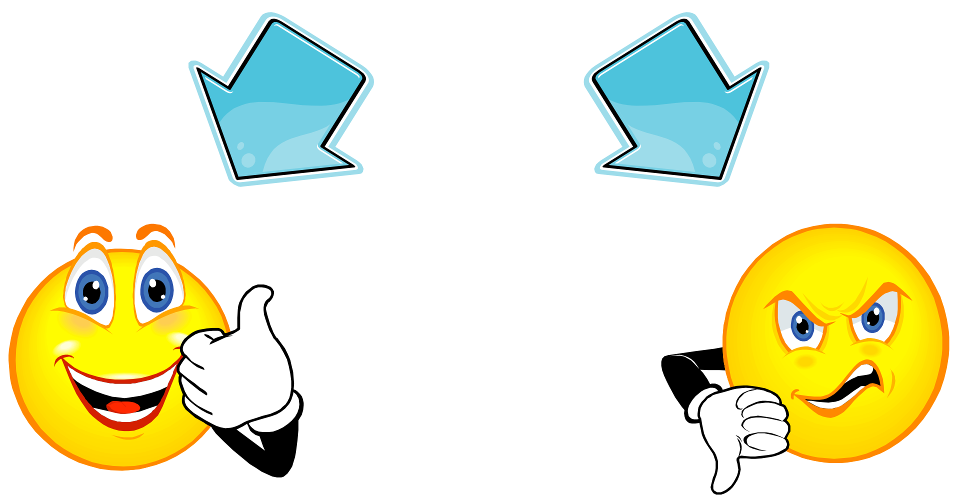
- Text categorization/classification: assign predefined categories to textual documents.

- Classification schemes:
  - Binary: spammer and non-spammer
  - Flat Classes: English, French, Spanish, etc.
  - Hierarchical: Arts, Sciences, Sports, Entertainments, etc.
    - Sciences: Physics, Chemistry, Biology, Medicine, etc.

- Applications: news routing and web content filtering.

What is Sentiment Analysis?

"… after a week of using the camera, I am very unhappy with the camera. The LCD screen is too small and the picture quality is poor. This camera is junk."



What is Sentiment Analysis?

- Software that classifies text input according to the opinions expressed in it
  - Positive or Negative

- Special case of text classification

- Types of Sentiment Analysis (SA)
  - Document-level
  - Aspect-based
  - Rating-based

Applications

- Online customer reviews
- Advertisement targeting
- Public relations/marketing
- Analytics/reputation mining
- Web content filtering
  - Cyber-bullying
  - Inflammatory text
- Information retrieval
  - Multi-perspective question answering
  - Automatic summarization

Problems

- Special case of topical text classification with special challenges
  - Topic-important features are frequent in a sub-set of documents and infrequent in the rest
  - Polar terms are infrequent in specific documents but occur in many documents
- Adjectives, Adverbs, Verbs and Nouns are all important to SA (in varying degrees)
- Separate polar words from topical words

Subjective Words

- A consumer is unlikely to write: "This camera is great. It takes great pictures. The LCD screen is great. I love this camera".

- But more likely to write: "This camera is great. It takes breathtaking pictures. The LCD screen is bright and clear. I love this camera".

- More diverse usage of subjective words: infrequent within but frequent across documents.

Challenges

"The movie was unpredictable"

"The car steering is unpredictable"

- Evaluative expressions are context dependent

Challenges

"How can anyone sit through this movie?"

- Some opinions are more subtle, containing no adjectives or adverbs.

Challenges

"My **wonderful** boyfriend took me to see this movie for our anniversary. It was **terrible**."

- Users mix their opinions with other information

Challenges

"The slow, methodical way he spoke. I loved it! It made him seem more arrogant and even **more evil**."

- Additional challenge in movie reviews: bad things can be favorable.

Dimensionality Reduction

- NLP: computational approaches for understanding and generating natural language
- Major issues
  - “Curse of dimensionality”: difficult to efficiently process text, and connect related meanings
  - Natural language is richly structured, highly variable, and complex
- Major goal
  - Dimensionality reduction while maintaining meaning

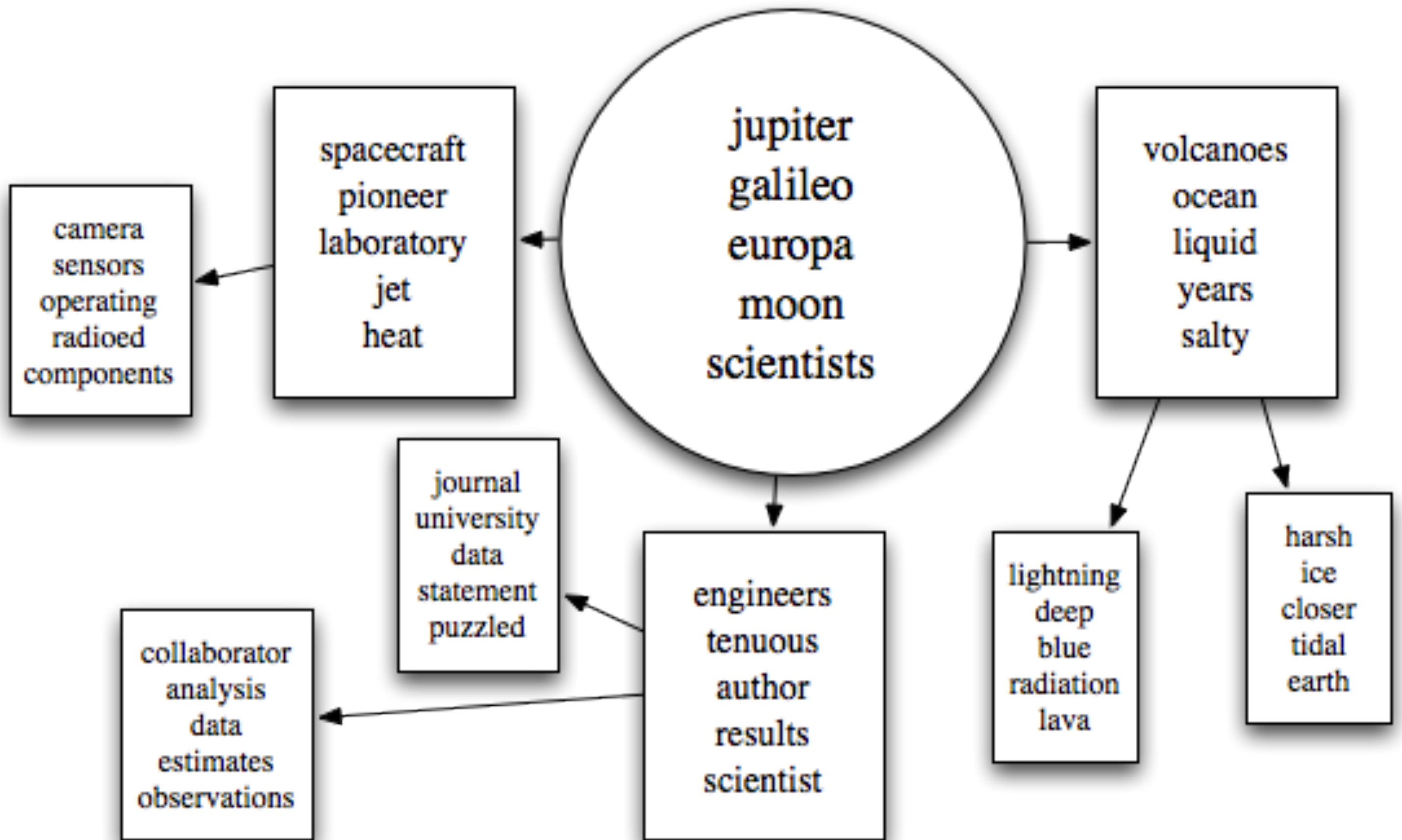
Topic Models

- Topic modeling is a relatively new statistical approach to understanding the thematic structure in a collection of data
- Used to uncover the hidden topical patterns in a corpus of documents
  - Dimensionality reduction from words down to topics
- Topic models are generative probabilistic admixture models in that we model the process by which documents are created and we posit that this is as a result of a random process

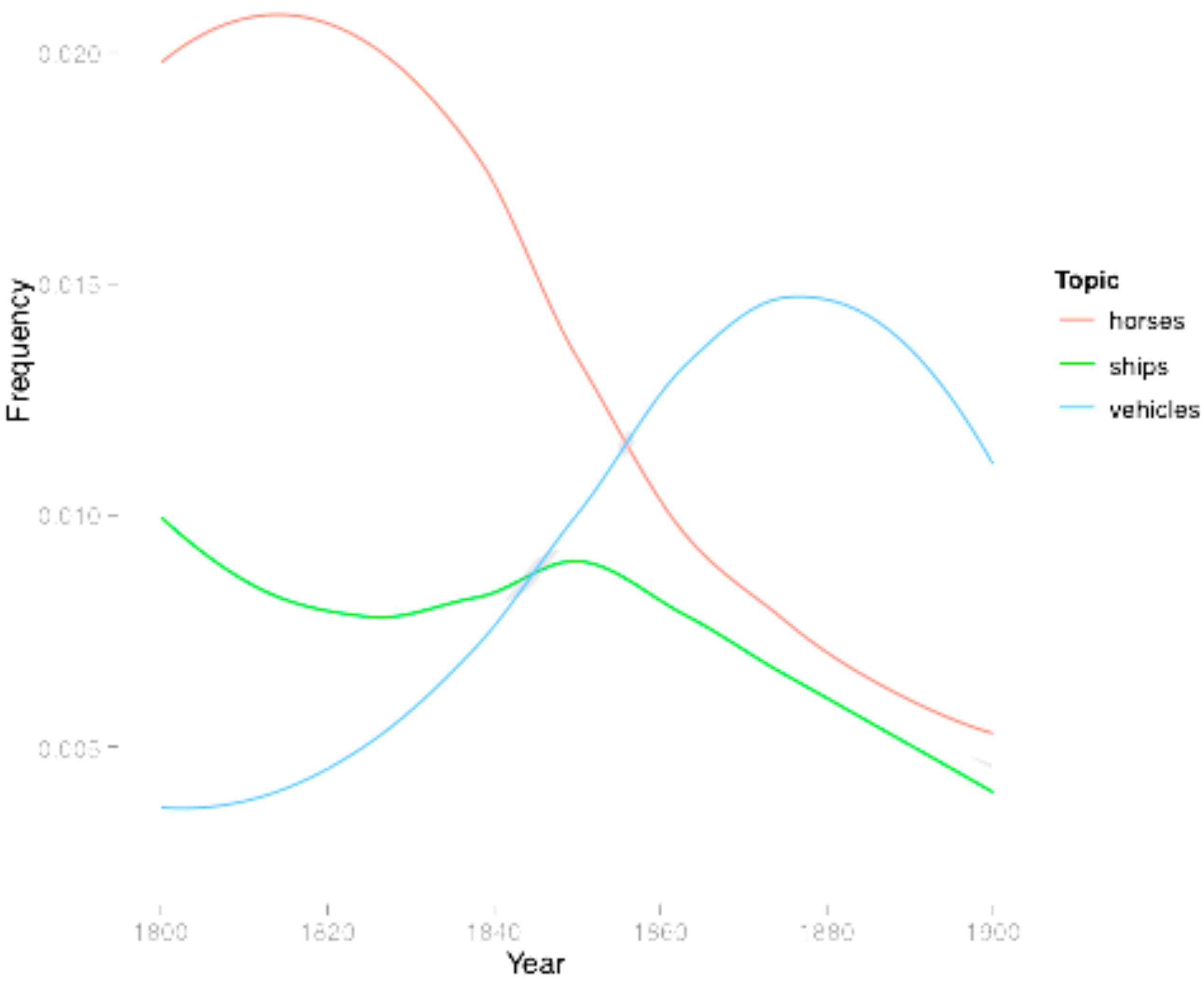
Discover Topics

charles	study	bush	surface
prince	found	protest	atmosphere
london	drug	texas	space
marriage	research	bushs	system
parker	risk	iraq	earth
camilla	drugs	president	probe
bowles	researchers	cindy	european
wedding	dr	war	moon
british	patients	ranch	huygens
thursday	disease	crawford	titan
king	vioxx	sheehan	mission
royal	health	son	friday
married	increased	casey	nasa
marry	merck	killed	scientists
wales	text	antiwar	cassini
queen	brain	california	saturns
diana	schizophrenia	george	agency
april	studies	mother	data
relationship	medical	road	titans
couple	effects	peace	14

Discover Hierarchies



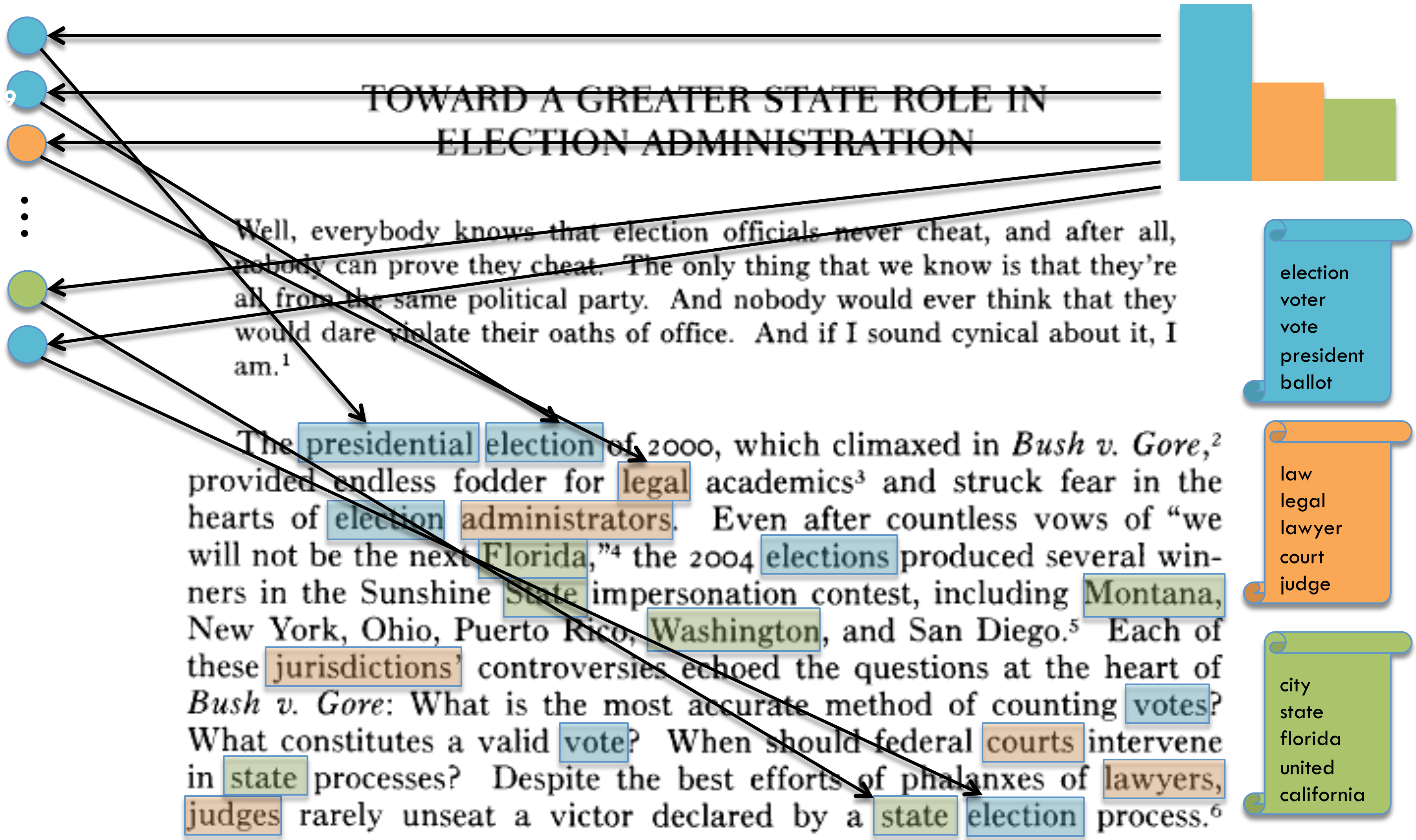
Topic Use Changing Through Time



This is the generative model of LDA

Each word is randomly drawn from a topic

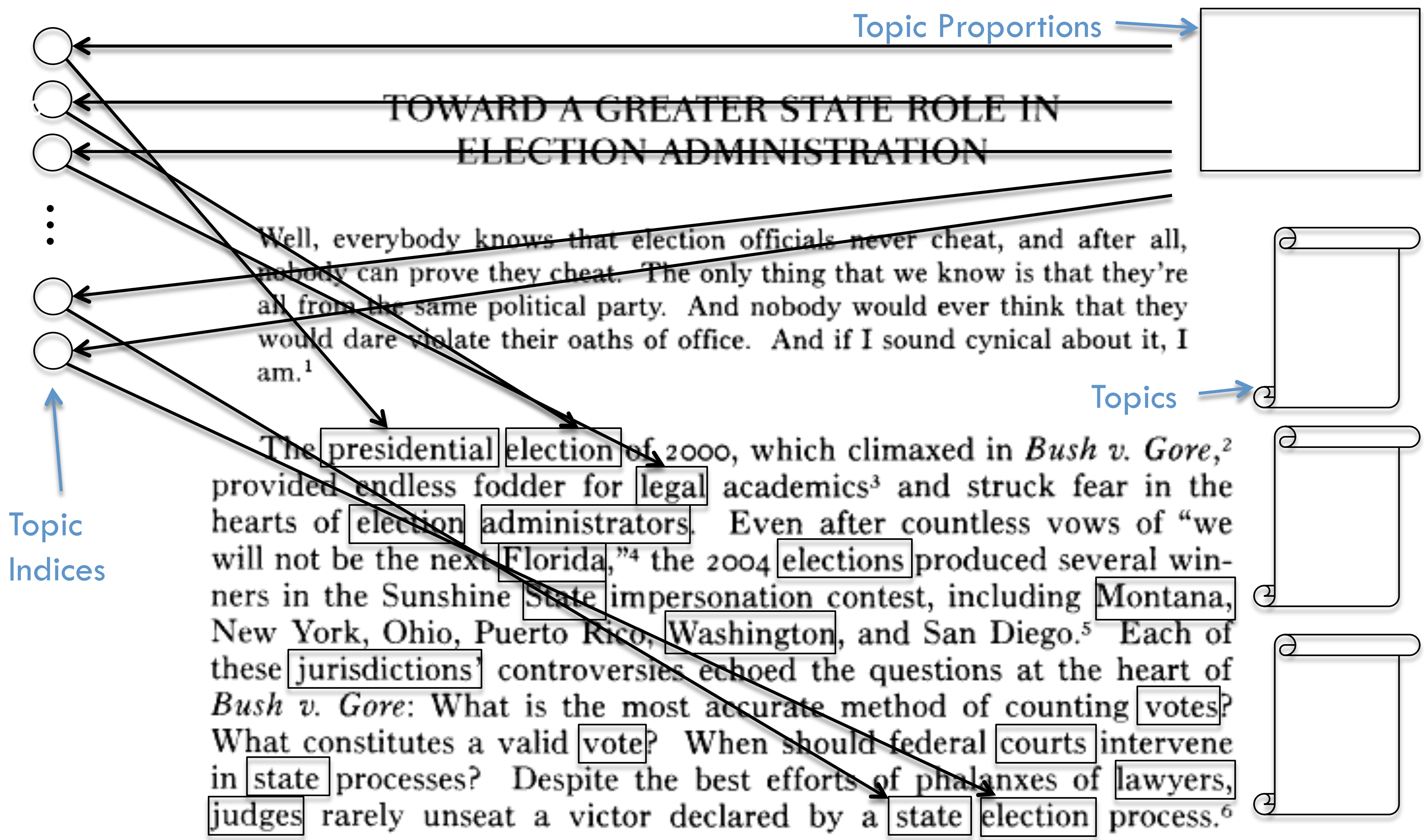
… of topics that are shared across the corpus…



*Harvard Law Review, Vol. 118, No. 7 (May, 2005), pp. 2314-2335 (Note).

Using probabilistic *reasoning*, we wish to *infer* the *latent structure* of the documents

However, all of this information is hidden. We only observe the documents.

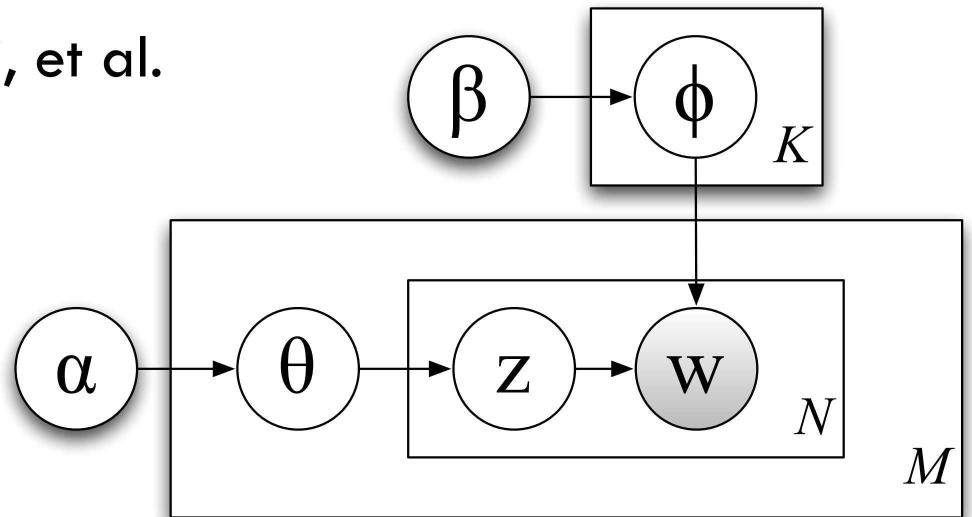


*Harvard Law Review, Vol. 118, No. 7 (May, 2005), pp. 2314-2335 (Note).

Latent Dirichlet Allocation (LDA)

Initially proposed by Blei, et al.

Generative Process:



1. $\phi^{(k)} \sim \mathrm{Dir}(\beta)$
2. For each document $d \in M$:
   a. $\theta_d \sim \mathrm{Dir}(\alpha)$
   b. For each word $w \in d$:
      i. $z \sim \mathrm{Discrete}(\theta_d)$
      ii. $w \sim \mathrm{Discrete}(\phi^{(z)})$

Dirichlet Distribution

- A family of continuous multivariate probability distributions parameterized by a vector α of positive real numbers
- A distribution of distributions: each sample is a multinomial distribution
- Often used a prior in Bayesian Statistics and is conjugate with multinomial distribution
- Smaller α's correspond to sparse distributions of the related elements

Inference

- We are interested in the posterior distributions for $\boldsymbol{\phi}$, $\boldsymbol{z}$ and $\boldsymbol{\theta}$
- Computing these distributions exactly is intractable
- We therefore turn to approximate inference techniques:
  - Gibbs sampling, variational inference, …
- *Collapsed* Gibbs sampling
  - The multinomial parameters are integrated out before sampling

Gibbs Sampling

- Popular MCMC (Markov Chain Monte Carlo) method that samples from the conditional distributions for the posterior variables

- For the joint distribution p(**x**)=p(x_1,x_2,…, x_m):
  1. Randomly initialize each x_i
  2. For t = 1, 2, …, T:
     - 2.1. $x_1^{t+1} \sim p(x_1 \mid x_2^t, x_3^t, \cdots, x_m^t)$
     - 2.2. $x_2^{t+1} \sim p(x_2 \mid x_1^{t+1}, x_3^t, \cdots, x_m^t)$
     - …
     - 2.m. $x_m^{t+1} \sim p(x_m \mid x_1^{t+1}, x_2^{t+1}, \cdots, x_{m-1}^{t+1})$

(Collapsed) Gibbs Sampling

- We integrate out the multinomial parameters so that the Markov chain stabilizes more quickly and we have less variables to sample

- Our sampling equation is given as follows:

$$p(z_i \mid z_{-i}, w) \propto \frac{n_{z_i}^{(d)} + \alpha_{z_i}}{n_{\cdot}^{(d)} + \alpha_{\cdot}} \times \frac{n_w^{(z_i)} + \beta}{n_{\cdot}^{(z_i)} + W\beta}$$

- GibbsLDA++: a free C/C++ implementation of LDA

References

- Chris Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.
- Chris Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008 (online copy available on the web)
- Daniel Jurafsky and James H. Martin. *Speech and Language Processing*. Second Edition. Pearson Education, 2008.

References

- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.
- Sharon Goldwater and Tom Griffiths. A fully Bayesian approach to unsupervised part-of-speech tagging. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 744–751, Prague, Czech Republic, June 2007.

References

- Brigitte Krenn and Christer Samuelsson. *The Linguist's Guide to Statistics*. http://www.essex.ac.uk/linguistics/clmt/papers/stats
- Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, 1991.
- Richard Durbin, Sean Eddy, Anders Krogh, and Graeme Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.