OVERVIEW OF STATISTICAL NATURAL LANGUAGE PROCESSING

Dr. Fei Song
School of Computer Science
University of Guelph

May 18, 2016

Outline

- What is Statistical Natural Language Processing (SNLP)?
- Language Models for Information Retrieval
- Text Classification and Sentiment Analysis
- Probabilistic Models (LDA, Bayesian HMM, and POSLDA) for language processing
- References

What is SNLP?

- Infer and rank the structures from text based on statistical language modeling.
  - Probability and Statistics
  - Machine Learning Techniques
- Started in late 1950's, but didn't get popular until early 1980's.
- Many applications: Information Retrieval, Information Extraction, Text Classification, Text Mining, and Biological Data Analysis.

Language Modeling

- A statistical language model requires the estimates for such probabilities:

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

- Probabilities to word sequences?

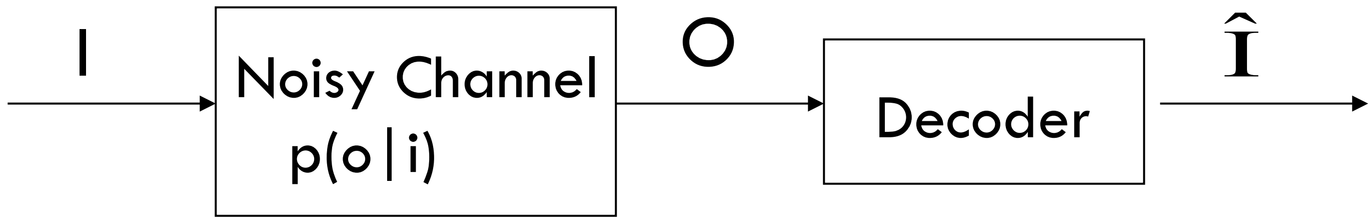
  $P(w_1\ w_2\ \ldots\ w_n) = P(w_1)\ P(w_2|w_1)\ \ldots\ P(w_n|w_1\ w_2\ \ldots\ w_{n-1})$

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

- Left-context only?
  - The {big, pig} dog …
  - P(dog|the big) >> P(dog|the pig)

Noisy Channel Framework

- Through decoding, we want to find the most likely input for the given observation.



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

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

Language Models for IR

- N-gram models:

Unigram: $P(w_{1,n}) = P(w_1)\ P(w_2)\ \ldots\ P(w_n)$

Bigram: $P(w_{1,n}) = P(w_1)\ P(w_2 \mid w_1)\ \ldots\ P(w_n \mid w_{n-1})$

Trigram: $P(w_{1,n}) = P(w_1)\ P(w_2 \mid w_1)\ \ldots\ P(w_n \mid w_{n-2,n-1})$

- Documents as language samples:

$$P(t_1, t_2, \ldots, t_n \mid d) = \prod_{i=1}^{n} P(t_i \mid d)$$

Language Models for IR

- Query as a generation process:

$$P(d \mid t_1, t_2, \ldots, t_m)$$

$$\Rightarrow P(d)P(t_1, t_2, \ldots, t_m \mid d) / P(t_1, t_2, \ldots, t_m)$$

(Bayesian theorem)

$$\Rightarrow P(d)P(t_1, t_2, \ldots, t_m \mid d)$$

(Uniform prior documents)

$$\Rightarrow P(t_1, t_2, \ldots, t_m \mid d) \Rightarrow \prod_{i=1}^{m} P(t_i \mid d)$$

(Unigram terms)

A Naïve Solution

- Maximum likelihood estimate:

$$P_{mle}(t \mid d) = \frac{tf_{t,d}}{dl_d}$$

$tf_{t,d}$: the raw term frequency of term t in document d

dl_d : the total number of tokens in document d.

Sparse Data Problem

- A document size is often too small

$$P(t_i \mid d) = 0 \Rightarrow \prod_{i=1}^{m} P(t_i \mid d) \Rightarrow 0$$

- A document size is fixed:

  P(information, retrieval|d) > 0 && keyword $\notin$d && crocodile $\notin$d

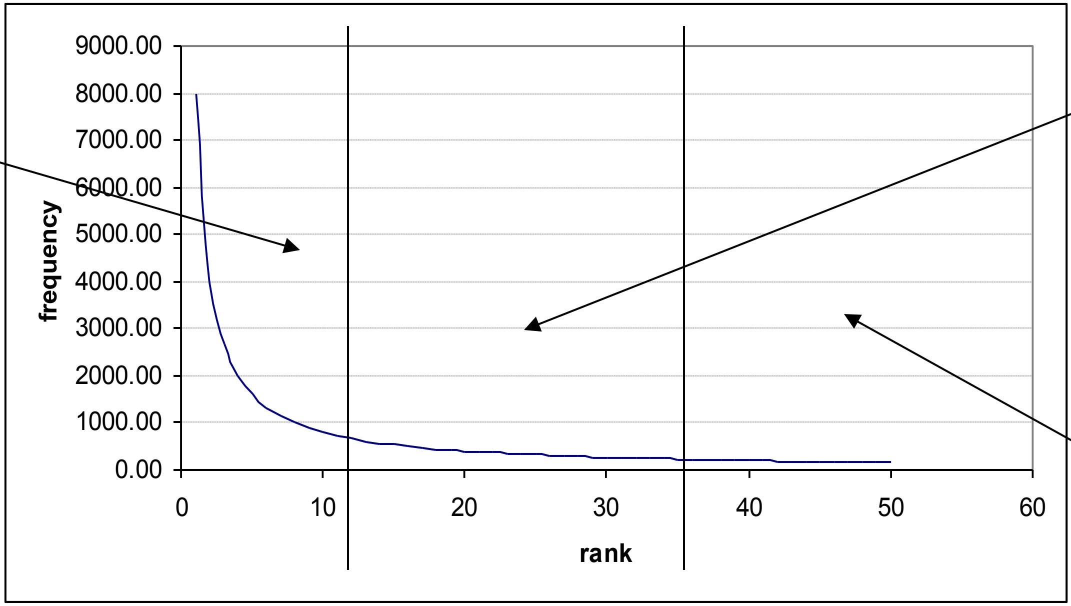
  => P(keyword|d) >> P(crocodile|d).

Zipf's Law

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

$f \propto 1/r$ or $f \times r = k$ for a constant k

A small number of common words



A reasonable number of medium-freq words

A large number of rare words

Data Smoothing

- Laplace's Law: T is the max number of terms.

$$P_{LAP}(t \mid d) = \frac{tf_{t,d} + 1}{dl_d + T}$$

- Extensions to Laplace's: Lidstone's Law.

$$P_{LID}(t \mid d) = \frac{tf_{t,d} + \lambda}{dl_d + T\lambda} = \mu P_{mle}(t \mid d) + (1 - \mu)/T$$

where $\mu = dl_d / (dl_d + T\lambda)$

Data Smoothing

- Smoothed with the collection model:

$$P_{combined}(t \mid d) = \omega \times P_{document}(t \mid d) + (1-\omega) P_{collection}(t)$$

  - The combined probability is still normalized with values between 0 and 1.
  - Further differentiation between missing terms such as "keyword" and "crocodile".
  - Collection model can be made stable by adding more documents into the collection.

Text Classifications/Categorizations

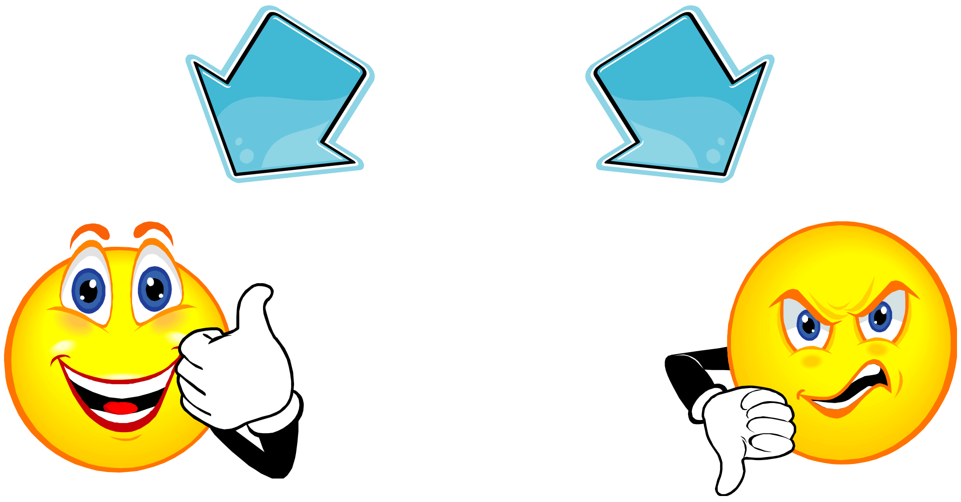
- Common classification problems:

Problems	Input	Categories
Tagging	context of a word	tag for the word
Disambiguation	context of a word	sense for the word
PP attachment	sentence	parse trees
Author identification	document	author(s)
Language identification	document	language(s)
Text categorization	document	topic(s)

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

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."



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.

Topic Models

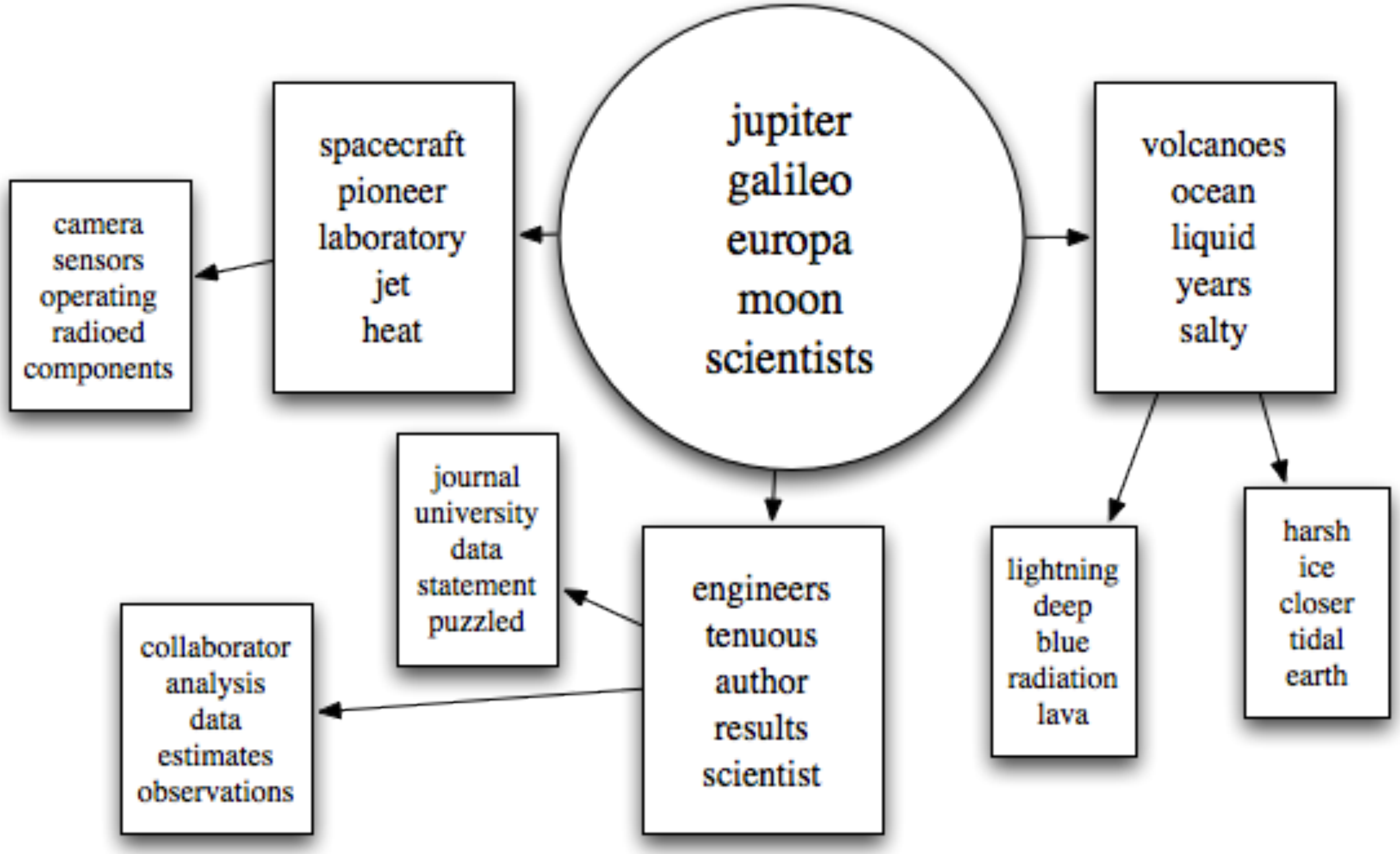
- Topic modeling is a relatively new statistical approach to understanding the thematic structure in a collection of data
  - Uncovering hidden topics in a corpus of documents
  - Reducing dimensionality from words down to topics

- Topic models treat the document creation as a random process of determining a topic proportion and selecting words from the related topic distributions.

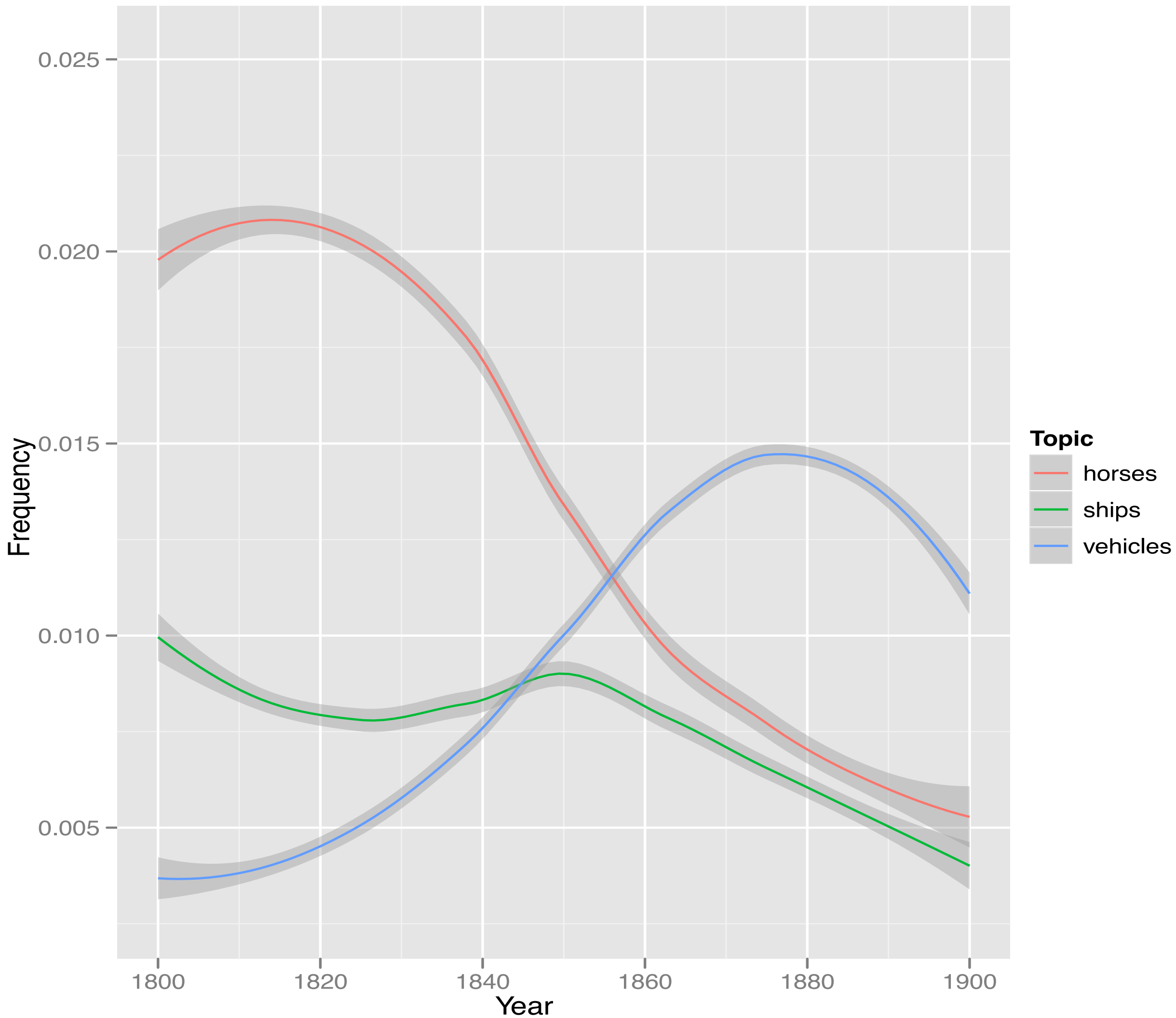
Discover Topics

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



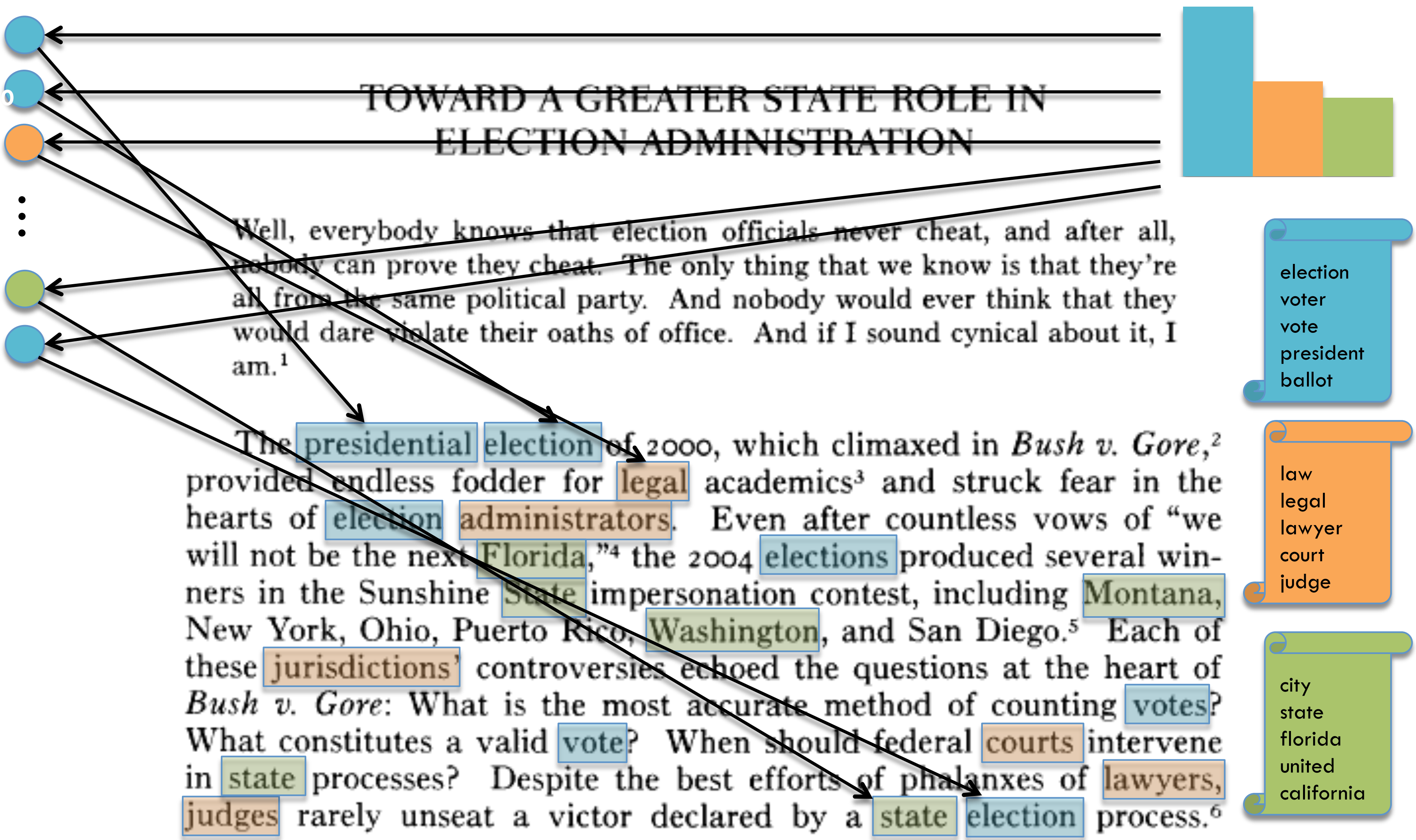
Topic Use Changing Through Time



This is the generative model of LDA

Each word is randomly drawn from a topic

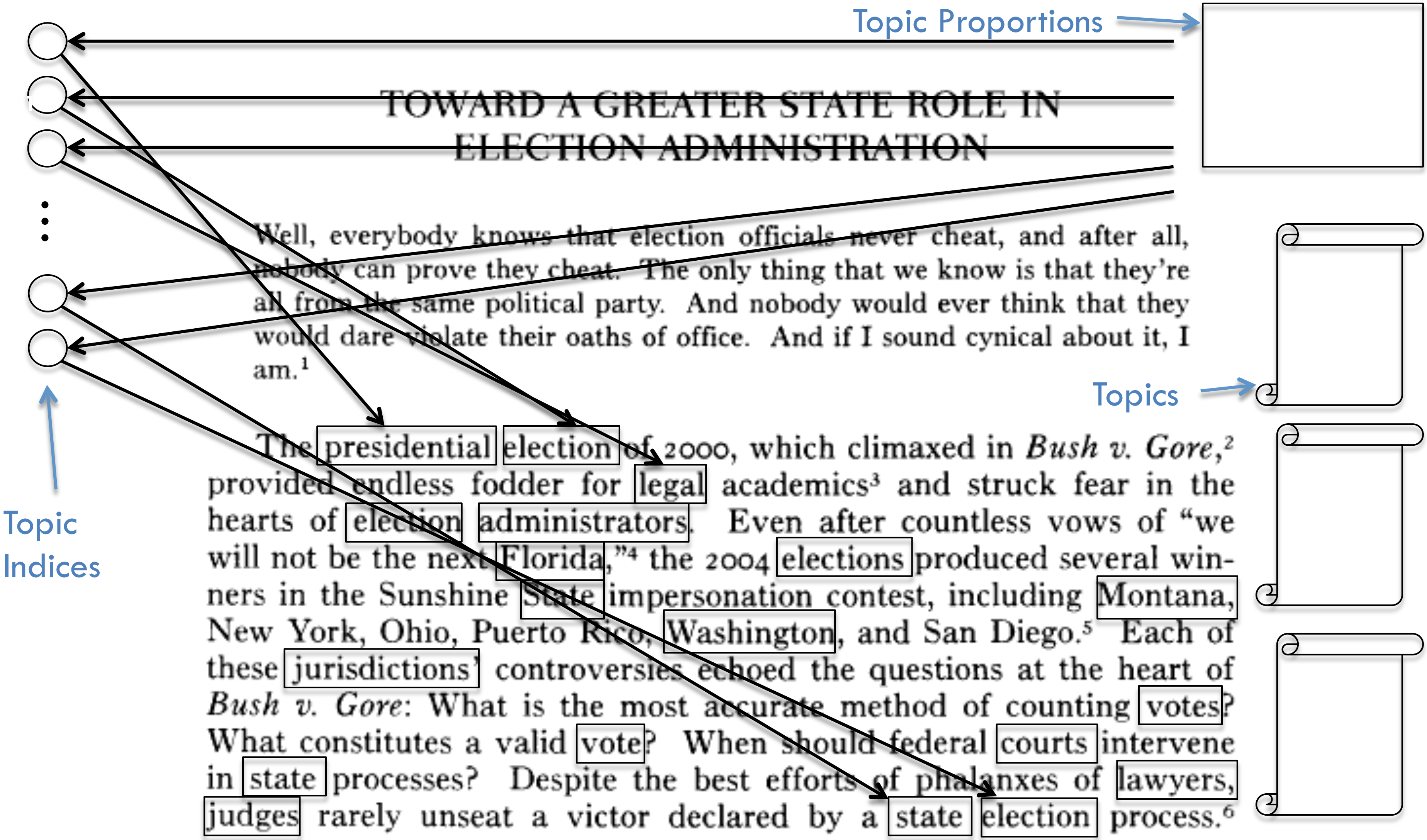
… of topics that are shared across the corpus…



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

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

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



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

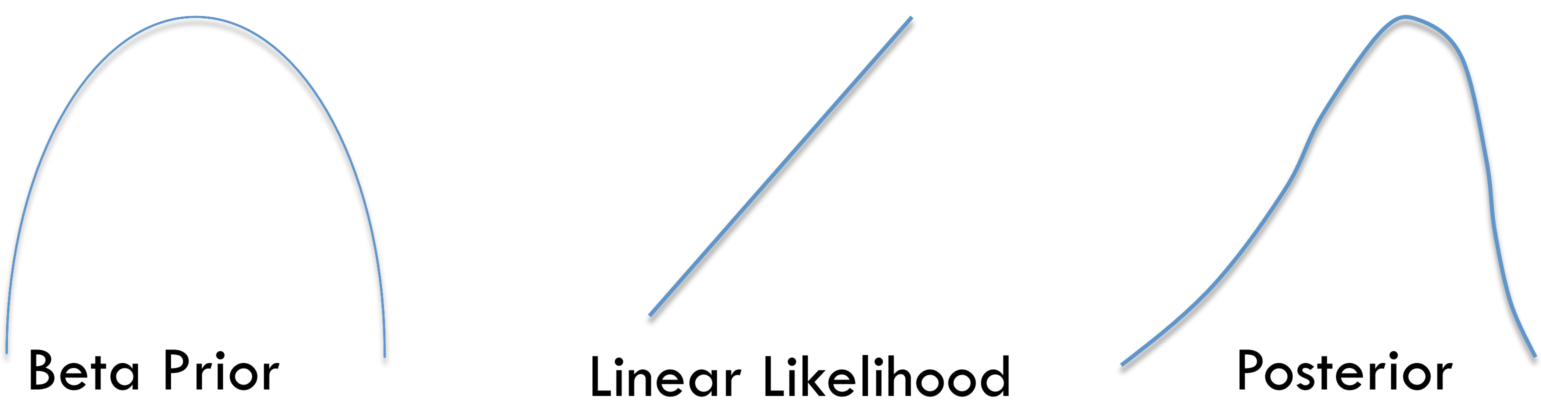
Bayesian Probability

- Bayes' Theorem

$$P(\theta \mid x) = \frac{p(x \mid \theta)\, p(\theta)}{p(x)}$$

$$posterior \propto likelihood \times prior$$

- Subjective probability: model prior by a given distribution

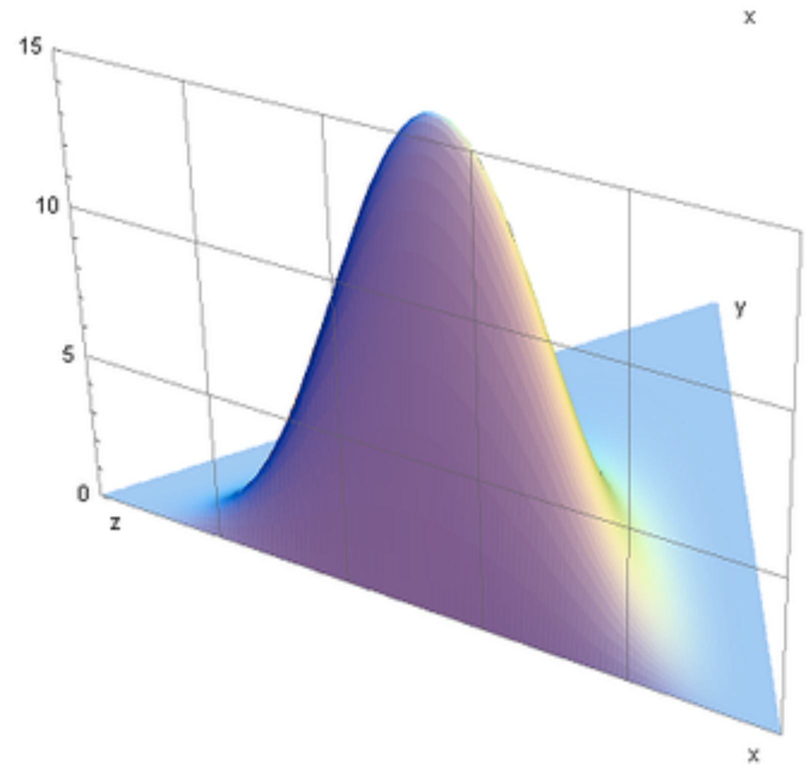


Dirichlet Distribution

- Distribution over distributions:

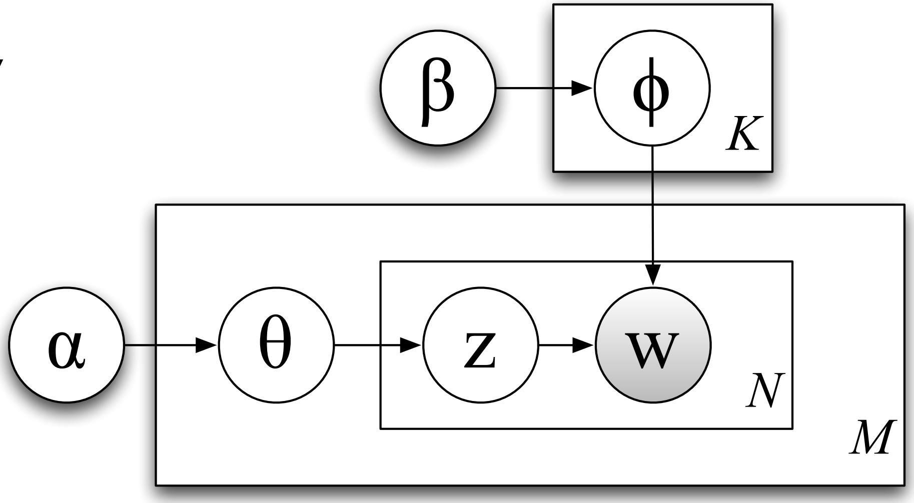
$$P(\theta \mid \alpha) = \frac{1}{B(\alpha)} \prod_{i=1}^{K} \theta^{\alpha_i - 1}$$

$$B(\alpha) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{K} \alpha_i)}$$



Latent Dirichlet Allocation (LDA)

- Initially proposed by Blei, et al. (2003):



Generative Process:

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

$$p(w,z,\theta,\phi \mid \alpha,\beta) = \prod_{k=1}^{K} p(\phi_k \mid \beta) \times \prod_{m=1}^{M} p(\theta_m \mid \alpha) \times \prod_{m=1}^{M} \prod_{n=1}^{N_m} p(z_{m,n} \mid \theta_m) p(w_{m,n} \mid \phi_{z_m})$$

Inference

- We are interested in the posterior distributions for ϕ, $\boldsymbol{z}$ and θ
- 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 $p(\mathbf{x})=p(x_1,x_2,\ldots, x_m)$:
  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 ϕ and θ so that the Markov chain stabilizes more quickly and we have less variables to sample.

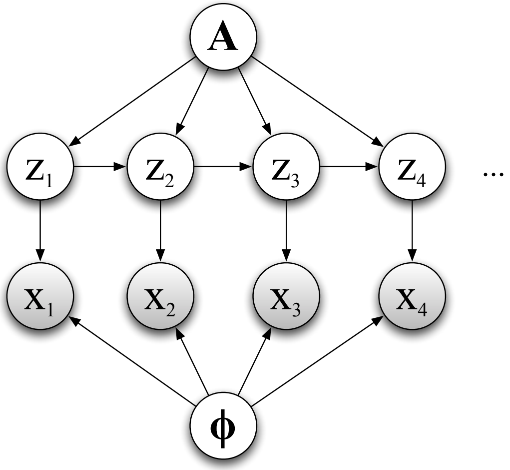
- Our sampling equation is given as follows:

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

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

Syntax Models

- Hidden Markov Model (HMM): The probability distribution of the latent variable z_i follows the Markov property and depends on the value of the previous latent variable z_{i-1}



- Each latent state z has a unique emission probability
  - This is a mixture model like LDA
- Useful for unsupervised POS tagging
  - Language exhibits a structure due to syntax rules
  - State-of-the-art: "Bayesian" HMM where transition rows and emission probabilities are random variables drawn from Dirichlet distributions [3]

Combining Topic and Syntax Models?

- Considering both axes of information can help us model text more precisely and can thus aid in prediction, processing, and ultimately many NLP tasks

- Example 1:
  - *Our favourite city during the trip was* __________.
  - How do we reason about what the missing word might be?
  - An HMM should be able to predict that it's a noun
  - LDA might be able to predict that it's a travel word*
  - A combined model could theoretically determine that it's a *noun* about *travel*

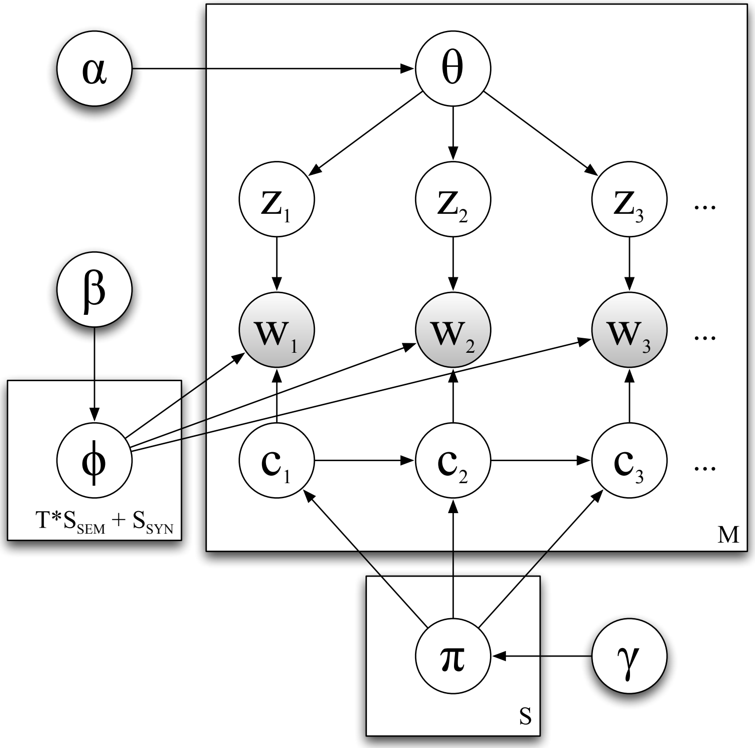
Combining Topic and Syntax Models?

- Example 2:
  - Is the word "*book*" a noun or a verb?
    - If we know that a "library" topic generated it, it's much more likely to be a noun
    - If we know that an "airline" topic generated it, it's more likely to be a verb ("to *book* a flight")

- Example 3:
  - We know that the word "*seal*" is a noun, what is its topic?
    - More likely to be related to "marine mammals" than "construction" ("to *seal* a crack")

POSLDA (Part-Of-Speech LDA) Model

- A "multi-faceted" topic model where word *w* depends on both topic *z* and class c when c is a "semantic" class
  - $w_i \sim p(w_i \mid c_i , z_i)$
- When c is a "syntactic" class the emitted word only depends on class c itself
- This model results in POS-specific topics and can automatically filter out "stop-words" that must be manually removed in LDA



POSLDA Generative Process

1. For each row $\pi_r \in \boldsymbol{\pi}$:
   a. Draw $\pi_r \sim$ Dirichlet(γ)
2. For each word distribution $\phi_n \in \boldsymbol{\phi}$:
   a. Draw $\phi_n \sim$ Dirichlet(β)
3. For each document $d \in D$:
   a. Draw $\theta_d \sim$ Dirichlet(α)
   b. For each token $i \in d$:
      i. Draw $c_i \sim \pi(c_{i-1})$
      ii. If $c_i \in C_{SYN}$:
         A. Draw $w_i \sim \phi^{\mathrm{SYN}}(c_i)$
      iii. Else ($c_i \in C_{SEM}$):
         i. Draw $z_i \sim \theta_d$
         ii. Draw $w_i \sim \phi^{\mathrm{SEM}}(c_i, z_i)$

POSLDA Interpretability

- Learned word distributions from TREC AP corpus:

"law"			"finance"			"health"		
adj	*verb*	*noun*	*adj*	*verb*	*noun*	*adj*	*verb*	*noun*
federal	filed	attorney	stock	rose	exchange	health	died	study
court	ruled	judge	wall	averaged	stock	medical	suffered	research
supreme	agreed	district	bond	issued	securities	aids	received	hospital
legal	contends	calif	million	fell	dow	drug	underwent	virus
civil	claims	county	american	gained	york	blood	found	report
appeals	contended	board	financial	dropped	inc	heart	carried	disease
tax	refused	loan	composite	rated	totaled	research	suffers	university
illegal	sued	san	common	traded	drexel	immune	leaves	doctor
government	won	court	business	stocks	commission	hospital	kills	person
financial	wrote	justice	dow	closed	lambert	cancer	took	patient

AUXILIARY	CONJUNCTION	DETERMINER	RELATIVE
is	and	the	that
was	but	a	which
be	or	an	who
are	&	this	when
has	so	some	what
have	both	such	how
will	times	any	where
would	nor	many	whose
says	plus	those	why
were	yet	these	whom

Generalized Probabilistic Model

- POSLDA reduces to LDA when the number of classes $S = 1$.
- POSLDA reduces to Bayesian HMM when the number of topics $K = 1$.
- POSLDA reduces to HMMLDA when the number of semantic classes $S_{sem} = 1$.

FS from Semantic Classes

- Research has shown that semantic classes such as adjectives, adverbs, and verbs are more useful for SA.
- Select representative words for a semantic class by picking the top-ranked words with the accumulative probability $\geq \theta$ (e.g., 75% or 90%).
- Merge all selected words into one set W_{sem}, and reduce it further by DF-cutoff if needed.

FS from Semantic Classes with Tagging

- POSLDA is unsupervised and the results do not usually match with human labeled answers.
- A tagging dictionary contains all the POS tags that can be used for the given words in a corpus.
- With a tagging dictionary, a word is only assigned to its related POS classes, but if not in the dictionary, the word will participate in all POS classes, same as the unsupervised process for POSLDA.

FS with Automatic Stopword Removal

- Similar to W_{sem}, we can also build W_{syn} from the syntactic classes to extract topic-independent stopwords.
- Such a process is both automatic and corpus-specific, avoiding under- or over-removal of the related words.
- Although POSLDA can separate semantic and syntactic classes, removing stopwords explicitly helps reduce the noise in the dataset.

FS for Aspect-Based SA

- POSLDA associates each topic with its related semantic classes such as “nouns about sports” and “verbs about travel”.
- By modeling topics as aspects, we can then select features from the corresponding semantic classes using the methods described earlier.
- To model aspects, we use manually prepared seed lists (possibly extended with a bootstrapping method), and pin them in the related aspects during the modeling process.

Questions?

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

- Bo Pang , Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. *Processings of EMNLP*, 2002.
- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. Journal of Machine Learning Research, 3:993-1022, 2003.
- Sharon Goldwater and Thomas Griffiths. A fully bayesian approach to unsupervised part-of-speech tagging. Proceedings of the 45th Annual Meeting of the ACL, 2007.

References

- William M. Darling. Generalized Probabilistic Topic and Syntax Models for Natural Language Processing. Ph.D. Thesis, University of Guelph, 2012.
- Haochen Zhou and Fei Song. Aspect-Level Sentiment Analysis Based on a Generalized Probabilistic Topic and Syntax Model. *Proceedings of FLAIRS-28*, 2015.