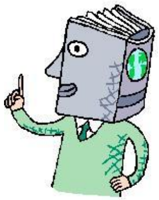


UNIVERSITY OF WASHINGTON

Latent Variable Models of Lexical Semantics

Alan Ritter

aritter@cs.washington.edu

Agenda

- Topic Modeling Tutorial
- Quick Overview of Lexical Semantics
- Examples & Practical Tips
  - Selectional Preferences
    - Argument types for relations extracted from the web
  - Event Type Induction
    - Induce types for events extracted from social media
  - Weakly Supervised Named Entity Classification
    - Classify named entities extracted from social media into types such as PERSON, LOCATION, PRODUCT, etc…

TOPIC MODELING

Useful References:

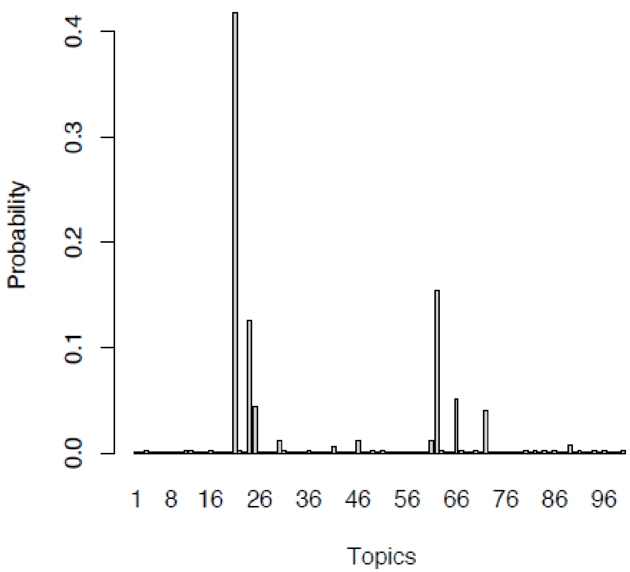
- **Parameter estimation for text analysis**
  – Gregor Heinrich
  – http://www.arbylon.net/publications/text-est.pdf
- **Gibbs Sampling for the Uninitiated**
  – Philip Resnik, Eric Hardisty
  – http://www.cs.umd.edu/~hardisty/papers/gsfu.pdf
- **Bayesian Inference with Tears**
  – Kevin Knight
  – http://www.isi.edu/natural-language/people/bayes-with-tears.pdf

Topic Modeling: Motivation

- Uncover Latent Structure
- Lower Dimensional Representation

- Probably a good fit if you have:
  - Large amount of grouped data
  - Unlabeled
  - Not even sure what the labels should be?

Topic Modeling: Motivation



"Genetics"	"Evolution"	"Disease"	"Computers"
human	evolution	disease	computer
genome	evolutionary	host	models
dna	species	bacteria	information
genetic	organisms	diseases	data
genes	life	resistance	computers
sequence	origin	bacterial	system
gene	biology	new	network
molecular	groups	strains	systems
sequencing	phylogenetic	control	model
map	living	infectious	parallel
information	diversity	malaria	methods
genetics	group	parasite	networks
mapping	new	parasites	software
project	two	united	new
sequences	common	tuberculosis	simulations

Figure 2: **Real inference with LDA.** We fit a 100-topic LDA model to 17,000 articles from the journal *Science*. At left is the inferred topic proportions for the example article in Figure 1. At right are the top 15 most frequent words from the most frequent topics found in

Terminology

- I’m going to use the terms:
  - Documents / Words / Topics
- Really these models apply to any kind of grouped data:
  - Images / Pixels / Objects
  - Users / Friends / Groups
  - Music / Notes / Genres
  - Archeological Sites / Artifacts / Building Type

Estimating Parameters from Text

- Consider document as a bag of random words:

$$P(D|\theta) = \prod_j P\left(w_j|\theta\right) = \prod_j \theta_{w_j} = \prod_w \theta_w{}^{N_w}$$

- How to estimate θ?

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

What is $P(\theta)$?

Dirichlet Distribution ($\mathrm{Dir}(\theta|\alpha)$):

$$P(\theta|\alpha) = \frac{1}{Z(\theta)} \times \prod {\theta_j}^{\alpha_j - 1}$$

$$(\theta|D, \alpha) = \frac{\prod_w {\theta_w}^{n_w} P(\theta|\alpha)}{P(D)}$$

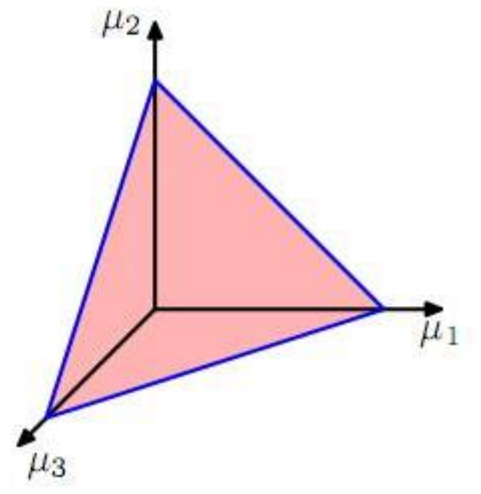
$$= \mathrm{Dir}(\theta|n + \alpha)$$

What is $P(\theta)$?

- Dirichlet Distribution ($\mathrm{Dir}(\theta|\alpha)$):

$$P(\theta|\alpha) = \frac{1}{Z(\theta)} \times \prod {\theta_j}^{\alpha_j - 1}$$

The Dirichlet distribution over three variables μ_1, μ_2, μ_3 is confined to a simplex (a bounded linear manifold) of the form shown, as a consequence of the constraints $0 \leqslant \mu_k \leqslant 1$ and $\sum_k \mu_k = 1$.



What is $P(\theta)$?

- Dirichlet Distribution ($\mathrm{Dir}(\theta|\alpha)$):

$$P(\theta|\alpha) = \frac{\Gamma(\sum\theta_j)}{\prod\Gamma(\theta_j)}\prod{\theta_j}^{\alpha_j-1}$$

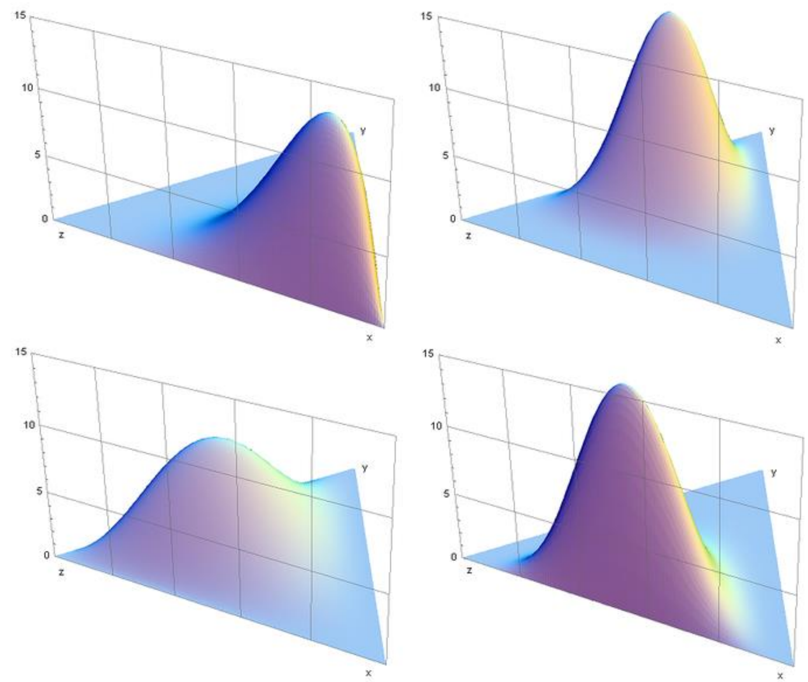
- Examples:

  $\mathrm{Dir}(6,2,2)$

  $\mathrm{Dir}(3,7,5)$

  $\mathrm{Dir}(6,2,6)$

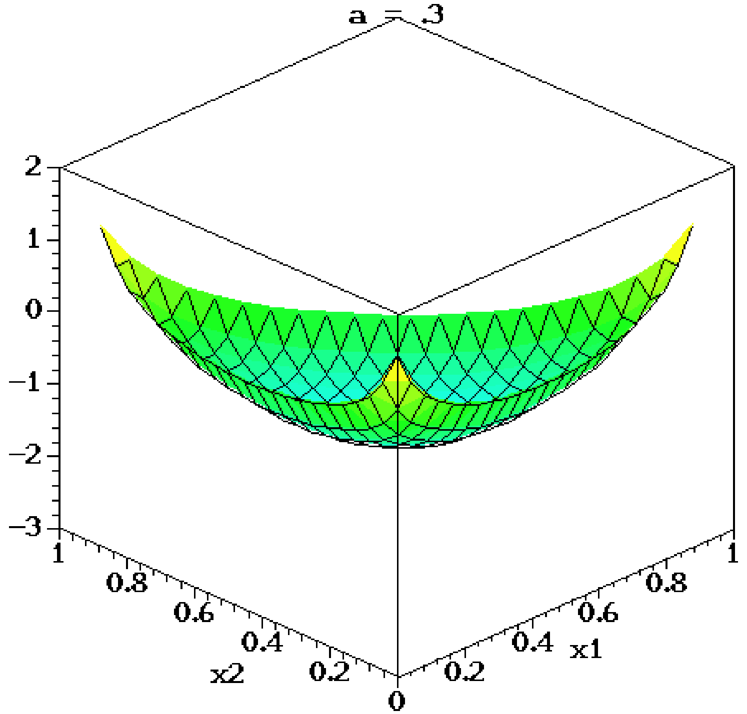
  $\mathrm{Dir}(2,3,4)$



What is $P(\theta)$?

- Sparse Prior: $0 < \alpha < 1$

$$\log(Dir(\theta|\alpha, \alpha, \alpha))$$

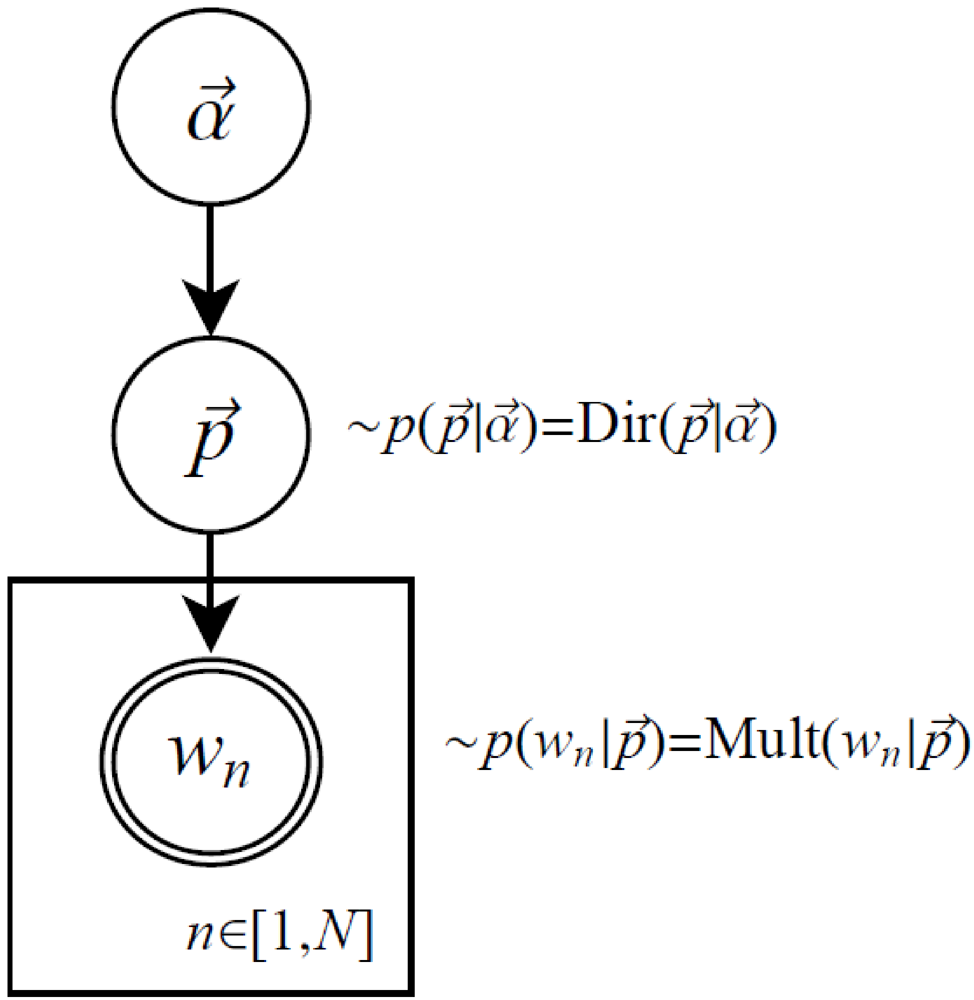


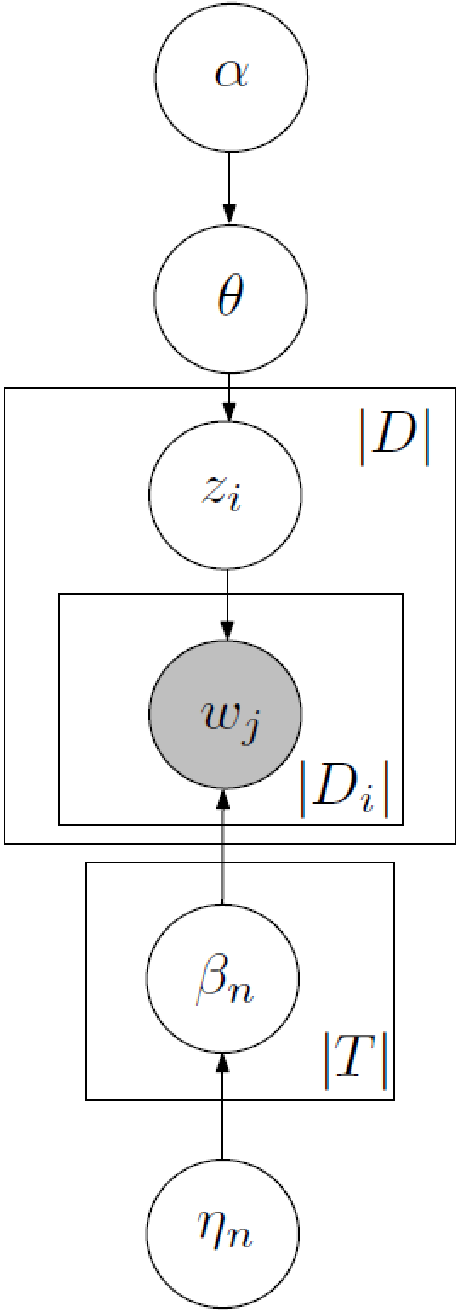
Generative Latent Variable Models

- Make up a story about how the data were generated
  - Involves Latent Variables which we never see
- Apply Bayesian inference to invert generative model
- Generative Story:

$$\theta \sim \mathrm{Dir}(\theta|\alpha)$$
$$w \sim \mathrm{Mult}(w|\theta)$$

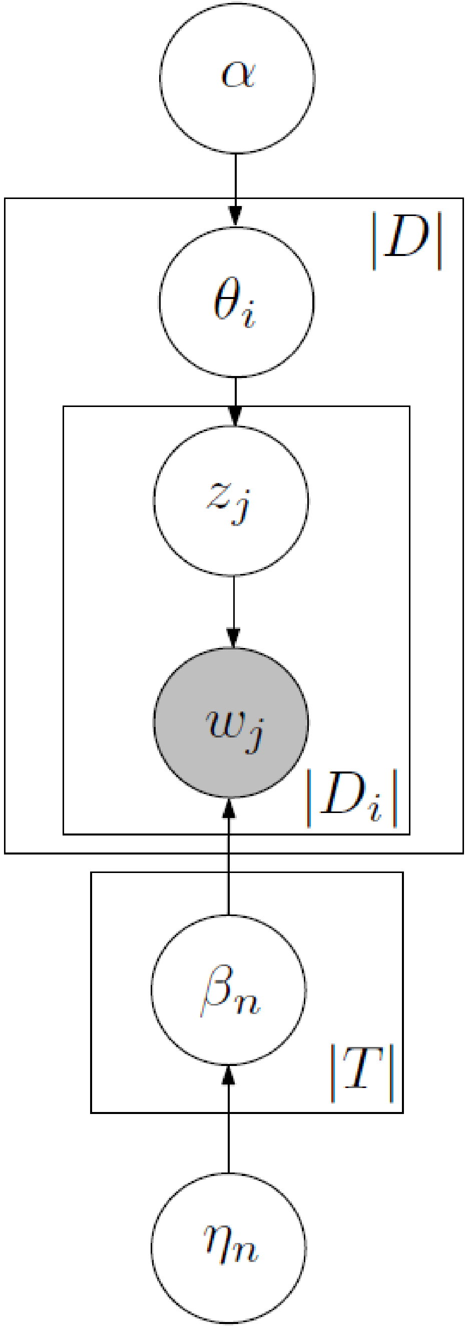
Graphical Models: Plate Notation





Naïve Bayes Mixture Model

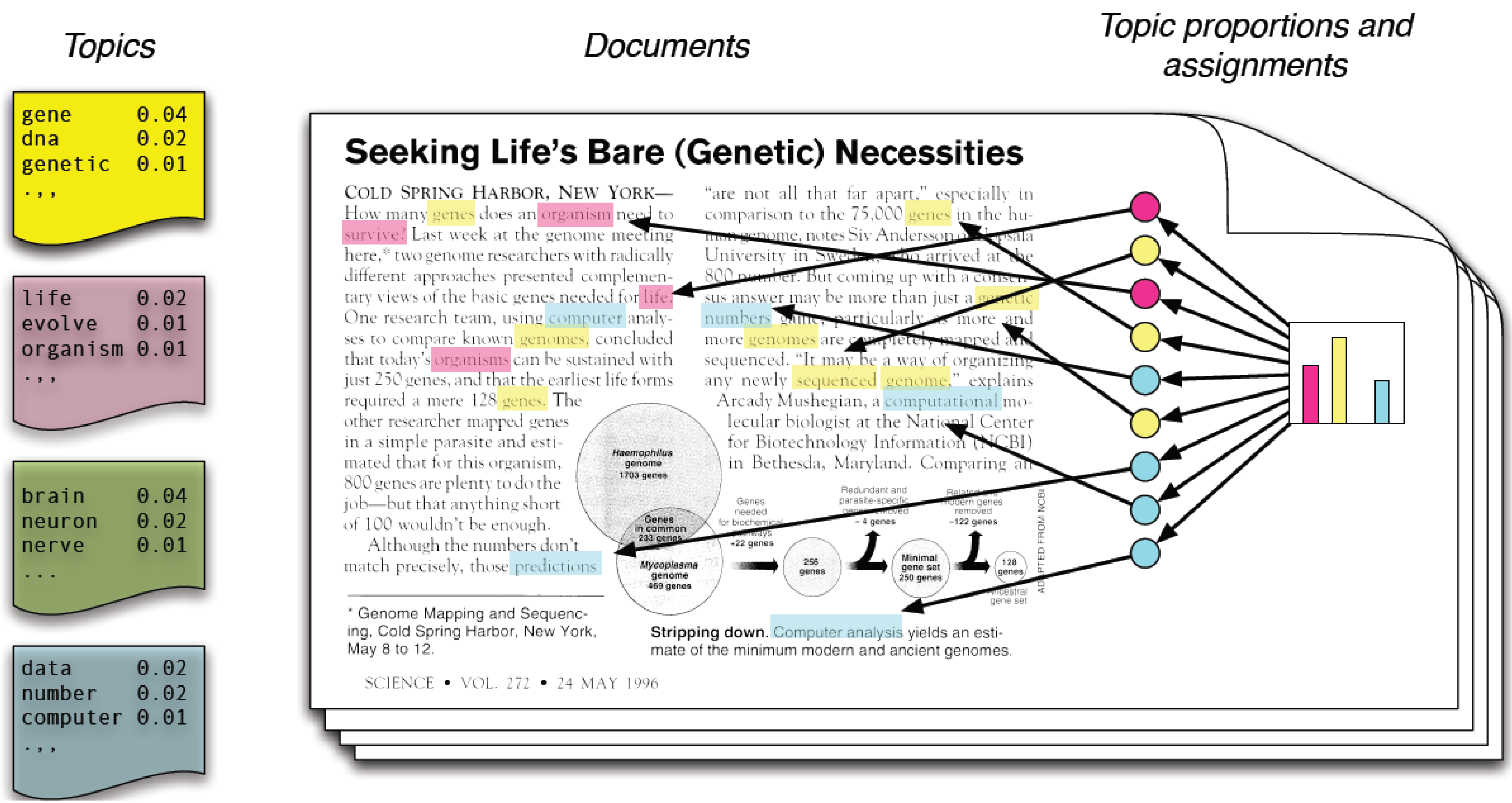
- Data is a set of "documents"
- Hidden Variables are categories
- Inference typically performed using EM
  - Maximize lower bound on likelihood

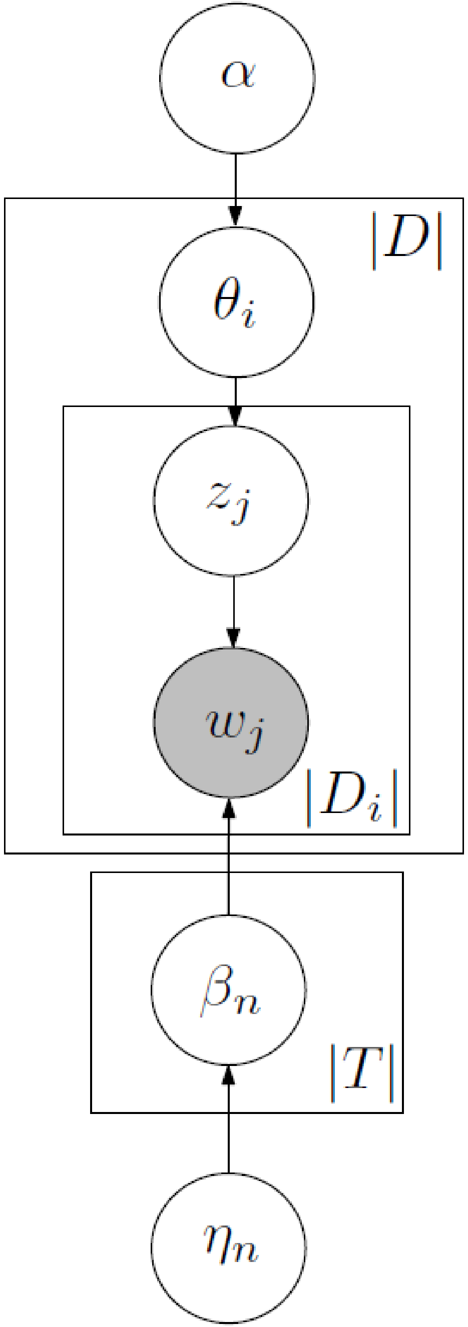


Latent Dirichlet Allocation

- Admixture Model
- Grouped Data
  - Each group is a mixture of underlying causes

Latent Dirichlet Allocation





Inference

- Exact Inference is Generally Intractable
- Instead, use approximate inference:
  - **Collapsed Gibbs Sampling**
  - Mean-Field Variational Inference
  - Expectation Propagation

Markov Chain Mote Carlo (MCMC)

1. Hidden Variables assignments are state space
2. Design a transition function
3. Start with arbitrary assignment to hidden variables
4. Randomly transition around the state space
5. After sufficient “burn in” samples come from the desired posterior

Gibbs Sampling

- Start with random assignment to hidden variables
- Iteratively sample each hidden variable
  - Forget the value of z_i
  - Condition on assignment to all other z's
  - Analytically integrate out parameters
- Want to sample from:

$$z_j \sim P\left(z_j \middle| Z_{-j}, W\right)$$

Sampling $z_j \sim P(z_i | Z_{-i}, W)$

- $P\left(z_j \middle| Z_{-j}, W\right) = \frac{P(z_j, Z_{-j}, W)}{P\left(Z_{-j}, W\right)}$
- Note: parameters integrated out
  – A bunch of gamma functions cancel out…

$$P\left(z_j = z \middle| Z_{-j}, W\right) = \frac{n_{w_{-j}}^{z} + \beta_{w_j}}{n^z + \beta} \times \frac{n_{z_{-j}}^{d} + \alpha_{z_j}}{n^d + \alpha}$$

Collapsed Gibbs Sampling for LDA

- Maintain counts:
  - $n_d^z =$ Document topic counts
  - $n_z^w =$ Topic word counts
- For each word position j in document i:
  - Forget z_j:
    - Decrement $n_{d_i}^{z_j}$
    - Decrement $n_{z_j}^{w_j}$
  - Sample new z_j:
    - $z_j \sim P\left(z_j \middle| Z_{-j}, W\right)$
  - Increment counts for new z_j:
    - Increment $n_{d_i}^{z_j}$
    - Increment $n_{z_j}^{w_j}$

Collapsed Gibbs Sampling for LDA

- Maintain counts:
  - $n_d^z =$ Document topic counts
  - $n_z^w =$ Topic word counts
- For each word position j in document i:
  - Forget z_j:
    - Decrement $n_{d_i}^{z_j}$
    - Decrement $n_{z_j}^{w_j}$
  - Sample new z_j:
    - $z_j \sim P\left(z_j \middle| Z_{-j}, W\right)$
  - Increment counts for new z_j:
    - Increment $n_{d_i}^{z_j}$
    - Increment $n_{z_j}^{w_j}$

One Gibbs Iteration

Gibbs Sampling in Practice

- Start out with random assignment to z_j's
- Run gibbs sampling for some number of “burn in iterations”
  - No way to really tell when the markov chain has converged to the posterior distribution
  - 1000 iterations often works well in practice…
- Use final assignment to z_j's to estimate θ_i's and β_k's

Gibbs Sampling: Scalability

- For each word, have to enumerate distribution over possible topics before sampling:
  - $O(w \times t)$
- Parallelization **(Newmann et. al. 2009)**
  - Simple approach to parallelization
  - Approximation (to approximate inference technique)
  - Works great in practice

LEXICAL SEMANTICS

Semantics

- Semantic Parsing:
  - Translate sentences into meaning representations

Semantics

- Semantic Parsing:
  - Translate sentences into meaning representations

$$
\begin{array}{cccc}
\text{What} & \text{states} & \text{border} & \text{Texas} \\
\hline
(S/(S\backslash NP))/N & N & (S\backslash NP)/NP & NP \\
\lambda f.\lambda g.\lambda x.f(x) \wedge g(x) & \lambda x.state(x) & \lambda x.\lambda y.borders(y,x) & texas \\
\hline
\multicolumn{2}{c}{>} & \multicolumn{2}{c}{>} \\
\multicolumn{2}{c}{S/(S\backslash NP)} & \multicolumn{2}{c}{(S\backslash NP)} \\
\multicolumn{2}{c}{\lambda g.\lambda x.state(x) \wedge g(x)} & \multicolumn{2}{c}{\lambda y.borders(y,texas)} \\
\hline
\multicolumn{4}{c}{>} \\
\multicolumn{4}{c}{S} \\
\multicolumn{4}{c}{\lambda x.state(x) \wedge borders(x,texas)}
\end{array}
$$

Semantics

- Semantic Parsing:
  - Translate sentences into meaning representations

$$
\begin{array}{cccc}
\text{What} & \text{states} & \text{border} & \text{Texas} \\
\hline
(S/(S\backslash NP))/N & N & (S\backslash NP)/NP & NP \\
\lambda f.\lambda g.\lambda x.f(x) \wedge g(x) & \lambda x.state(x) & \lambda x.\lambda y.borders(y,x) & texas
\end{array}
$$

$$
\xrightarrow{\quad\quad\quad\quad\quad\quad} >
\qquad
\xrightarrow{\quad\quad\quad\quad\quad\quad} >
$$

$$
\begin{array}{cc}
S/(S\backslash NP) & (S\backslash NP) \\
\lambda g.\lambda x.state(x) \wedge g(x) & \lambda y.borders(y,texas)
\end{array}
$$

$$
\xrightarrow{\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad} >
$$

$$
\begin{array}{c}
S \\
\lambda x.\boxed{state}(x) \wedge \boxed{borders}(x,\boxed{texas})
\end{array}
$$

S: (n) **state**, province (the territory occupied by one of the constituent administrative districts of a nation) *"his state is in the deep south"*

S: (n) **state** (the way something is with respect to its main attributes) *"the current state of knowledge"; "his state of health"; "in a weak financial state"*

S: (n) **state** (the group of people comprising the government of a sovereign state) *"the state has lowered its income tax"*

S: (n) **state**, nation, country, land, commonwealth, res publica, body politic (a politically organized body of people under a single government) *"the state has elected a new president"; "African nations"; "students who had come to the nation's capitol"; "the country's largest manufacturer"; "an industrialized land"*

S: (n) state of matter, **state** ((chemistry) the three traditional states of matter are solids (fixed shape and volume) and liquids (fixed volume and shaped by the container) and gases (filling the container)) *"the solid state of water is called ice"*

S: (n) **state** (a state of depression or agitation) *"he was in such a state you just couldn't reason with him"*

S: (n) country, **state**, land (the territory occupied by a nation) *"he returned to the land of his birth"; "he visited several European countries"*

S: (n) Department of State, United States Department of State, State Department, **State**, DoS (the federal department in the United States that sets and maintains foreign policies) *"the Department of State was created in 1789"*

- Se

  – ons

Texas

$(S/(S$ … $\qquad NP$

$\lambda f.\lambda g.\lambda$ … $\qquad texas$

$\lambda g.\lambda x.state(x) \wedge g(x) \qquad \lambda y.borders(y, texas)$

S

$$\lambda x.\boxed{state}(x) \wedge \boxed{borders}(x, \boxed{texas})$$

Why Latent Variable Models?

- Automatically Induce dictionaries
  - Similar to topics in topic models
  - Matches the domain
- Little Supervision/Annotation Required
  - Annotate/Train/Test paradigm doesn’t scale well to problems is Lexical Semantics
- Generative Models Provide Principled Answers:
  - How to learn dictionaries from corpora?
  - How to disambiguate words in context?
  - Answer a wide variety of task-specific queries

Latent Variables in Lexical Semantics: Selected Work

- Word Sense Induction
  (Brody and Lapata 2009)
  (Reisinger and Mooney 2011)
- **Selectional Preferences**
  (Ritter, Mausam, Etzioni 2010)
  (O Seaghda 2010)
- **Named Entity Classification**
  (Elsner, Charniak, Johnson 2009)
  (Ritter, Clark, Mausam, Etzioni 2011)
- **Event Type Induction**
  (Ritter, Mausam, Etzioni, Clark 2012)

Selectional Preferences

- Encode admissible arguments for a predicate
  - E.g. “eat X”

Plausible	Implausible
chicken	Windows XP
eggs	physics
cookies	the document
…	…

Selectional Preferences

- Encode admissible arguments for a predicate
  - E.g. “eat X”

Plausible	Implausible
FOOD	Windows XP
	physics
	the document
…	…

Motivating Examples

- *“…the **Lions** defeated the Giants….”*

Motivating Examples

- *"…the **Lions** **defeated** the Giants…."*


Motivating Examples

- *"...the **Lions** **defeated** the Giants...."*

- X defeated Y => X played Y
  - Lions ***defeated*** the Giants
  - Britian ***defeated*** Nazi Germany

Topic Modeling For Selectional Preferences

born_in(Sergey Brin,Moscow)
headquartered_in(Microsoft, Redmond)

born_in(Bill Gates, Seattle)

born_in(Einstein, March)
founded_in(Google, 1998)

headquartered_in(Google, Mountain View)
born_in(Sergey Brin,1973)
founded_in(Microsoft, Albuquerque)
born_in(Einstein, Ulm)
founded_in(Microsoft, 1973)

Topic Modeling For Selectional Preferences

```
headquartered_in(Google,       Mountain View)
headquartered_in(Microsoft,    Redmond)
```

```
born_in(Sergey Brin,1973)
born_in(Einstein,   March)
born_in(Einstein,   Ulm)
born_in(Sergey Brin,Moscow)
born_in(Bill Gates, Seattle)
```

```
founded_in(Microsoft,    Albuquerque)
founded_in(Google,       1998)
founded_in(Microsoft,    1973)
```

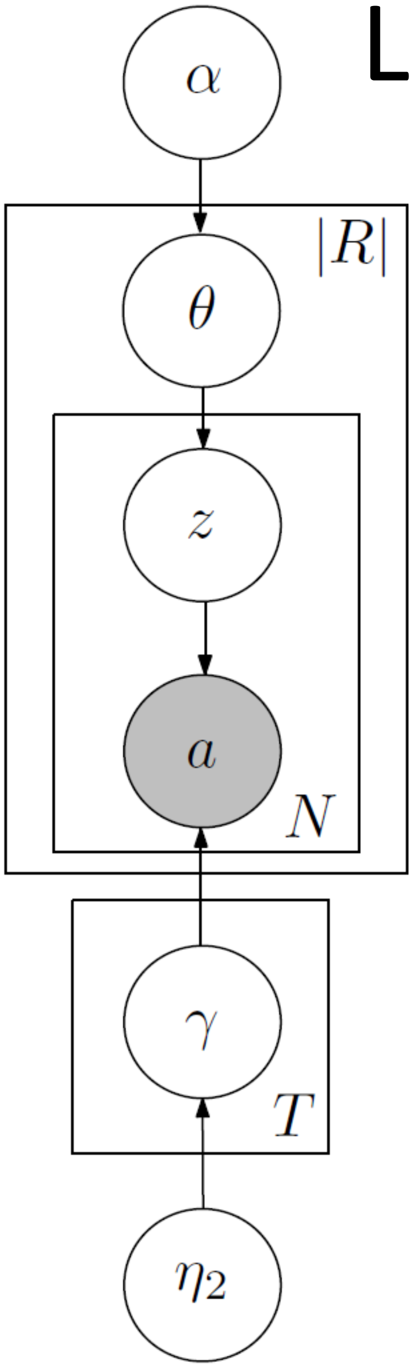
Verbs as "Documents"

```
headquartered_in(Google,        Mountain View)
headquartered_in(Microsoft,     Redmond)

born_in(Sergey Brin  1973)
born_in(Einstein,    March)
born_in(Einstein,    Ulm)
born_in(Sergey Brin  Moscow)
born_in(Bill Gates,  Seattle)

founded_in(Microsoft,      Albuquerque)
founded_in(Google,         1998)
founded_in(Microsoft,      1973)
```

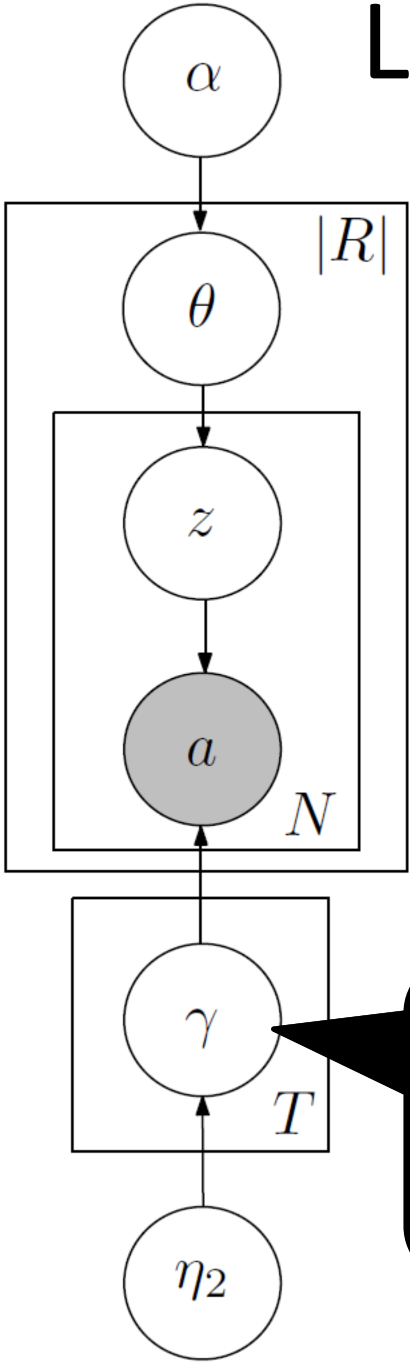
Args can have multiple Types

```
headquartered_in(Google,        Mountain View)
headquartered_in(Microsoft,     Redmond)

born_in(Sergey Brin  1973)
born_in(Einstein,    March)
born_in(Einstein,    Ulm)
born_in(Sergey Brin  Moscow)
born_in(Bill Gates,  Seattle)

founded_in(Microsoft,      Albuquerque)
founded_in(Google,         1998)
founded_in(Microsoft,      1973)
```

LDA Generative “Story”

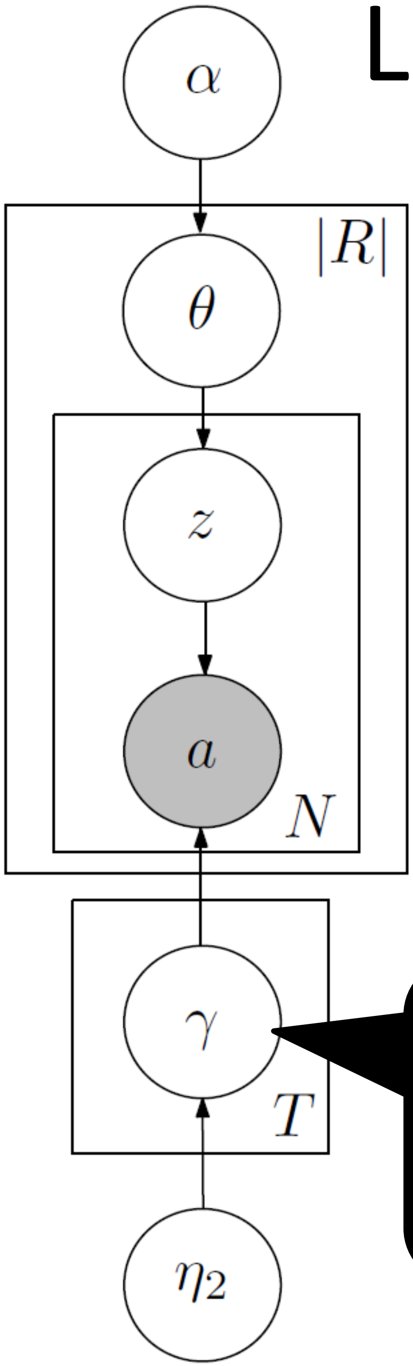


LDA Generative "Story"



For each type, pick a random distribution over words

LDA Generative "Story"



For each type, pick a random distribution over words

Type 1: **Location**

P(New York|T1)= 0.02

P(Moscow|T1)= 0.001

…

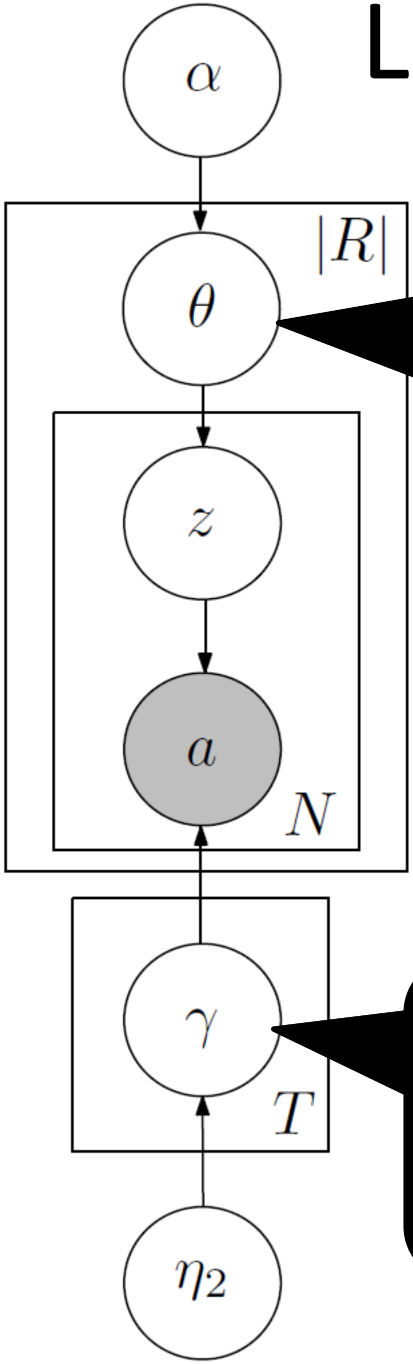
Type 2: **Date**

P(June|T2)=0.05

P(1988|T2)=0.002

…

LDA Generative "Story"



For each relation, randomly pick a distribution over types

For each type, pick a random distribution over words

Type 1: **Location**

P(New York|T1)= 0.02

P(Moscow|T1)= 0.001

…

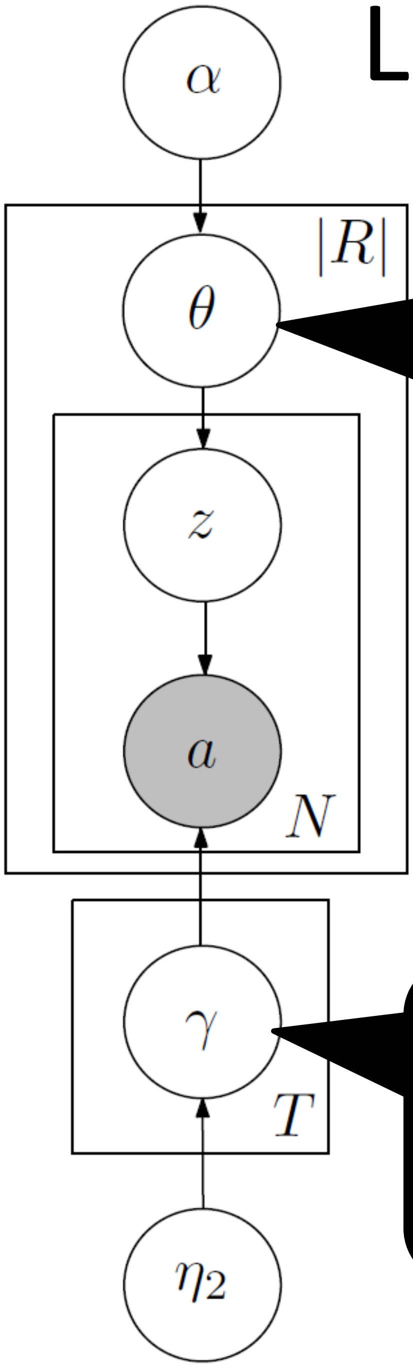
Type 2: **Date**

P(June|T2)=0.05

P(1988|T2)=0.002

…

LDA Generative "Story"



For each relation, randomly pick a distribution over types

born_in X

P(**Location**|**born_in**)= 0.5

P(**Date**|**born_in**)= 0.3

…

For each type, pick a random distribution over words

Type 1: **Location**

P(New York|T1)= 0.02

P(Moscow|T1)= 0.001

…

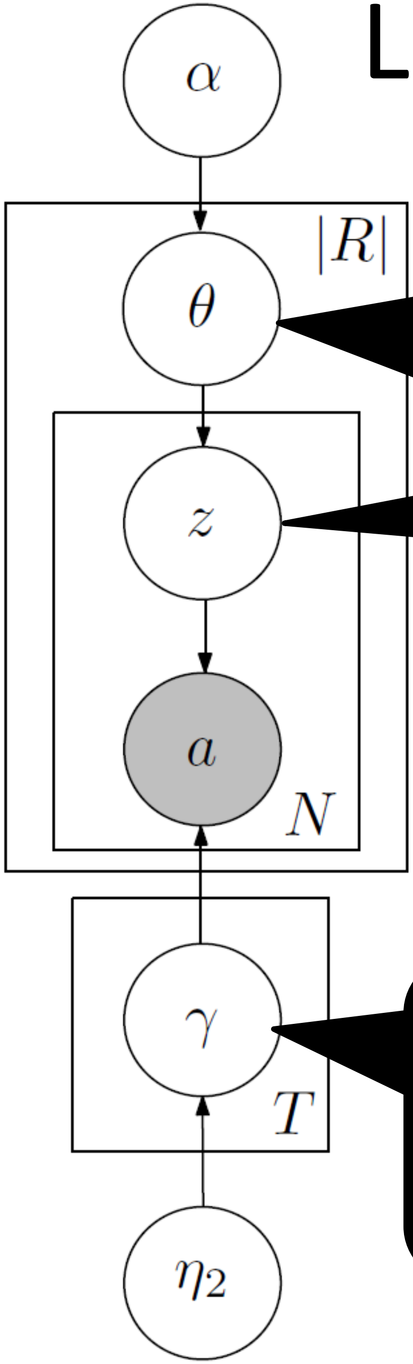
Type 2: **Date**

P(June|T2)=0.05

P(1988|T2)=0.002

…

LDA Generative "Story"



For each relation, randomly pick a distribution over types

For each extraction, first pick a type

For each type, pick a random distribution over words

born_in X

P(Location|born_in)= 0.5

P(Date|born_in)= 0.3

…

Type 1: Location

P(New York|T1)= 0.02

P(Moscow|T1)= 0.001

…

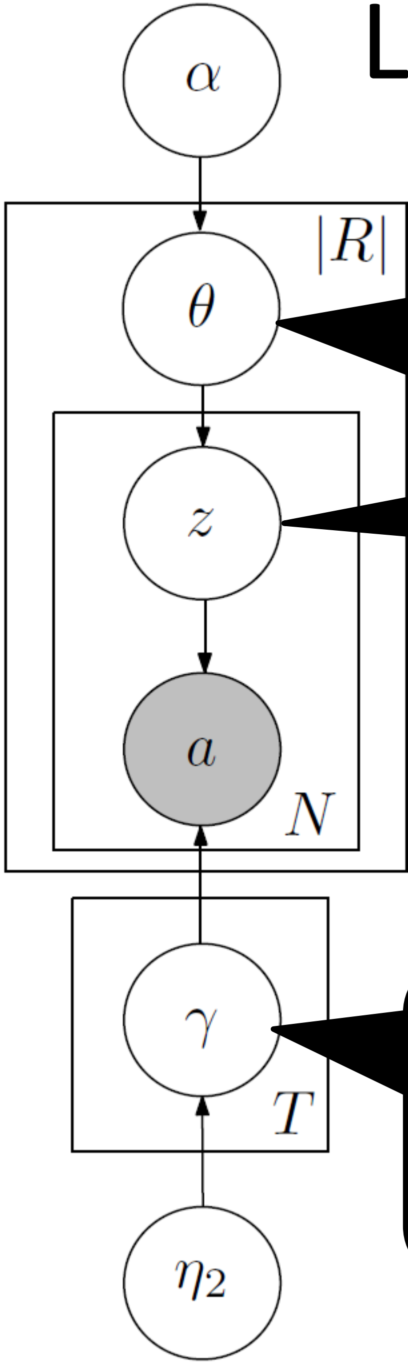
Type 2: Date

P(June|T2)=0.05

P(1988|T2)=0.002

…

LDA Generative "Story"



For each relation, randomly pick a distribution over types

`born_in X`

P(Location|`born_in`)= 0.5

P(Date|`born_in`)= 0.3

…

For each extraction, first pick a type

`born_in` Location

For each type, pick a random distribution over words

Type 1: Location

P(New York|T1)= 0.02

P(Moscow|T1)= 0.001

…

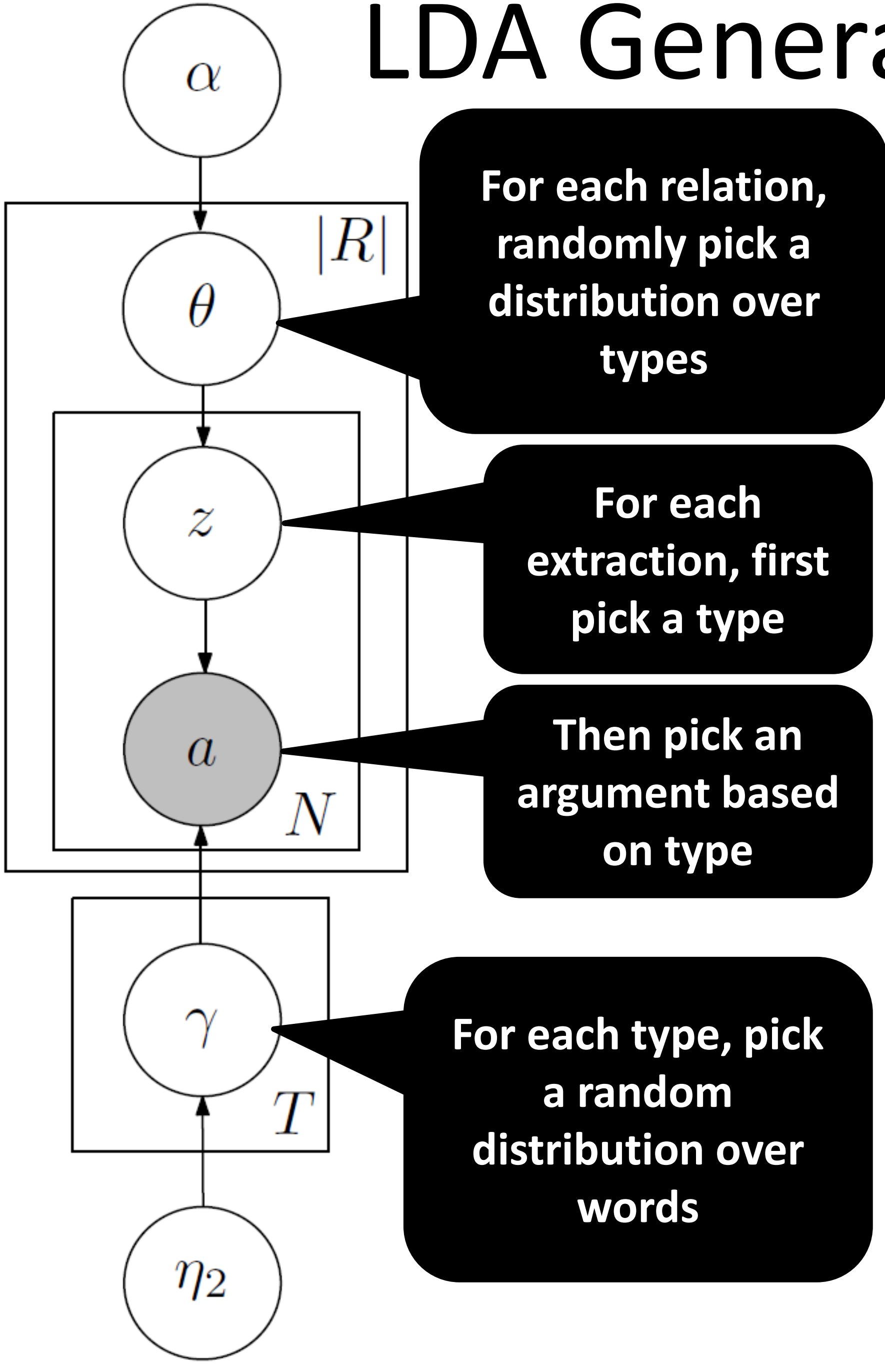
Type 2: Date

P(June|T2)=0.05

P(1988|T2)=0.002

…

LDA Generative “Story”



born_in X

P(Location|born_in)= 0.5

P(Date|born_in)= 0.3

…

born_in Location

Type 1: Location

P(New York|T1)= 0.02

P(Moscow|T1)= 0.001

…

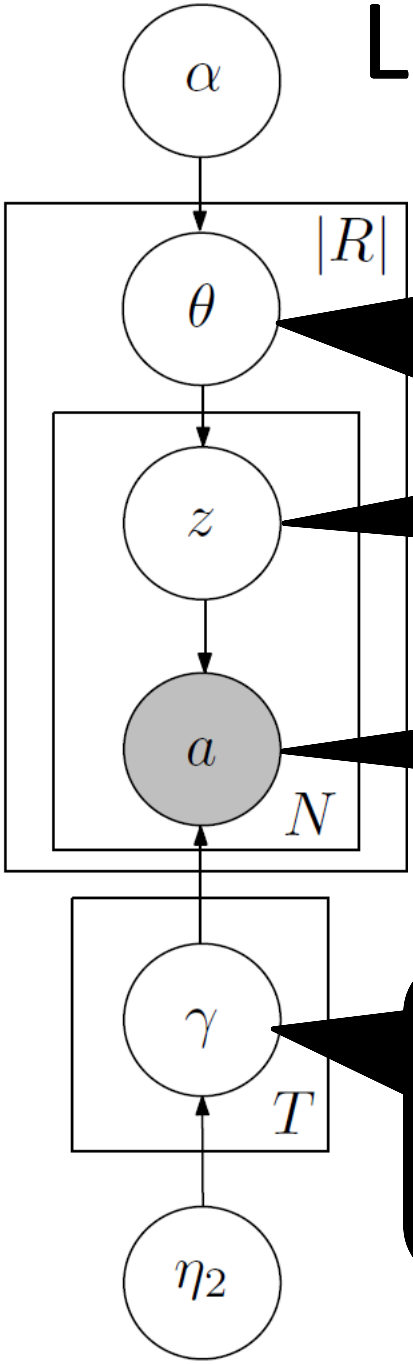
Type 2: Date

P(June|T2)=0.05

P(1988|T2)=0.002

…

LDA Generative "Story"



For each relation, randomly pick a distribution over types

`born_in X`

P(**Location**|`born_in`)= 0.5

P(**Date**|`born_in`)= 0.3

…

For each extraction, first pick a type

`born_in` **Location**

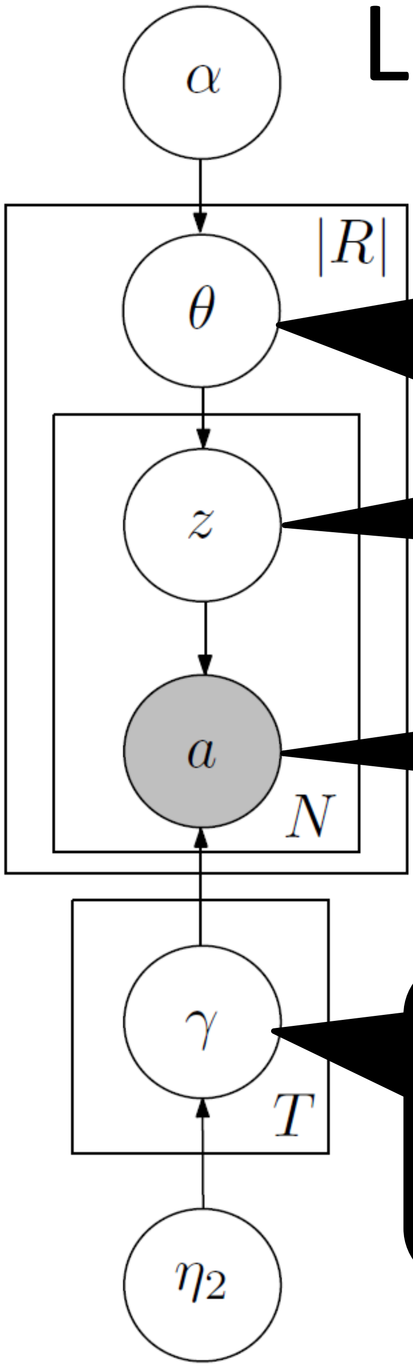
Then pick an argument based on type

`born_in` New York

For each type, pick a random distribution over words

Type 1: **Location**	Type 2: **Date**
P(New York|T1)= 0.02	P(June|T2)=0.05
P(Moscow|T1)= 0.001	P(1988|T2)=0.002
…	…

LDA Generative "Story"



For each relation, randomly pick a distribution over types

For each extraction, first pick a type

Then pick an argument based on type

For each type, pick a random distribution over words

born_in X

P(**Location**|**born_in**)= 0.5

P(**Date**|**born_in**)= 0.3

…

born_in **Location**   **born_in** **Date**

born_in New York

Type 1: **Location**

P(New York|T1)= 0.02

P(Moscow|T1)= 0.001

…

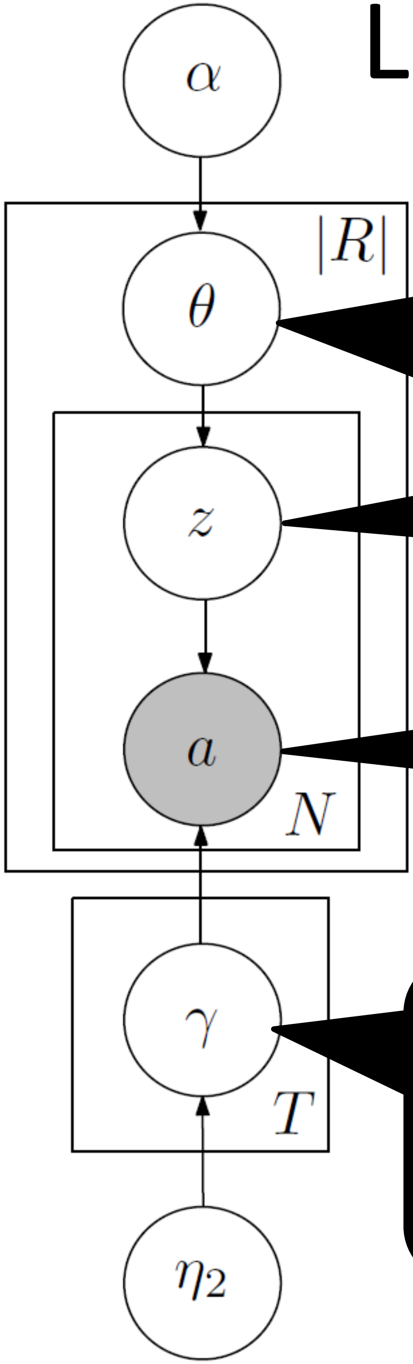
Type 2: **Date**

P(June|T2)=0.05

P(1988|T2)=0.002

…

LDA Generative "Story"



For each relation, randomly pick a distribution over types

For each extraction, first pick a type

Then pick an argument based on type

For each type, pick a random distribution over words

`born_in X`

P(Location|`born_in`)= 0.5
P(Date|`born_in`)= 0.3
…

`born_in` Location   `born_in` Date

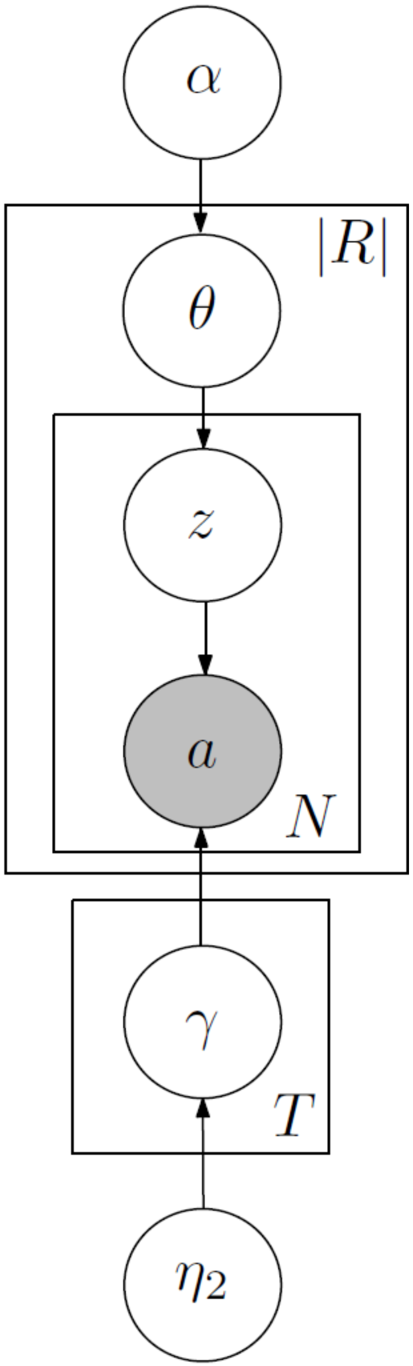
`born_in` New York   `born_in` 1988

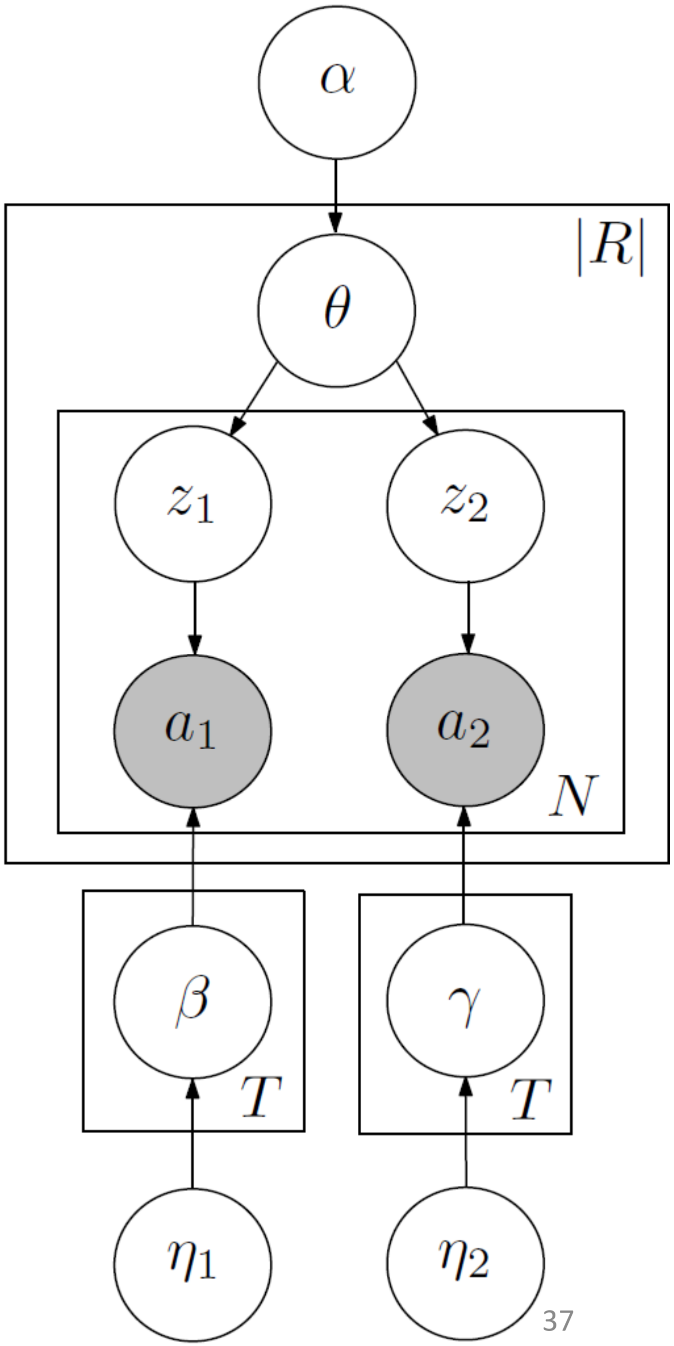
Type 1: Location
P(New York|T1)= 0.02
P(Moscow|T1)= 0.001
…

Type 2: Date
P(June|T2)=0.05
P(1988|T2)=0.002
…

LinkLDA

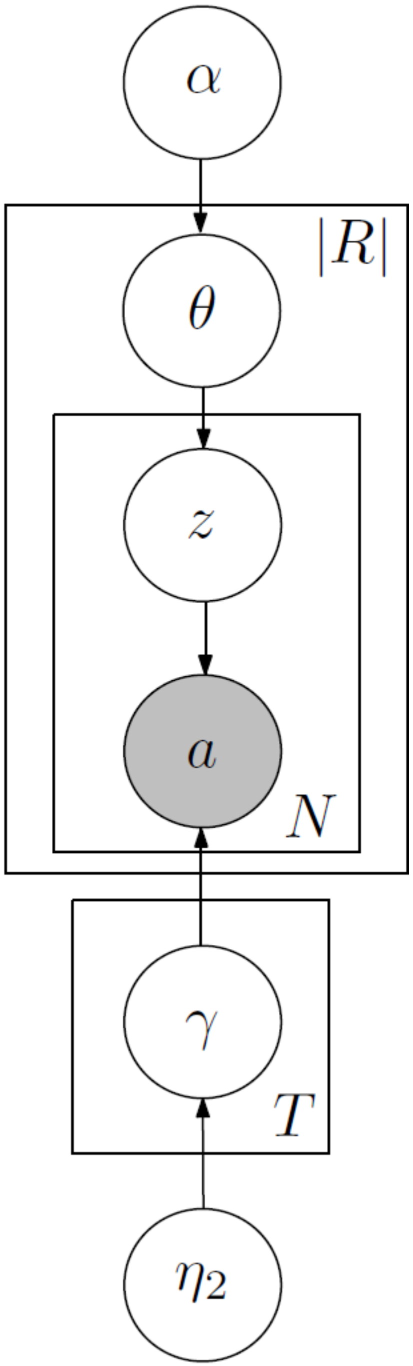
[Erosheva et. al. 2004]



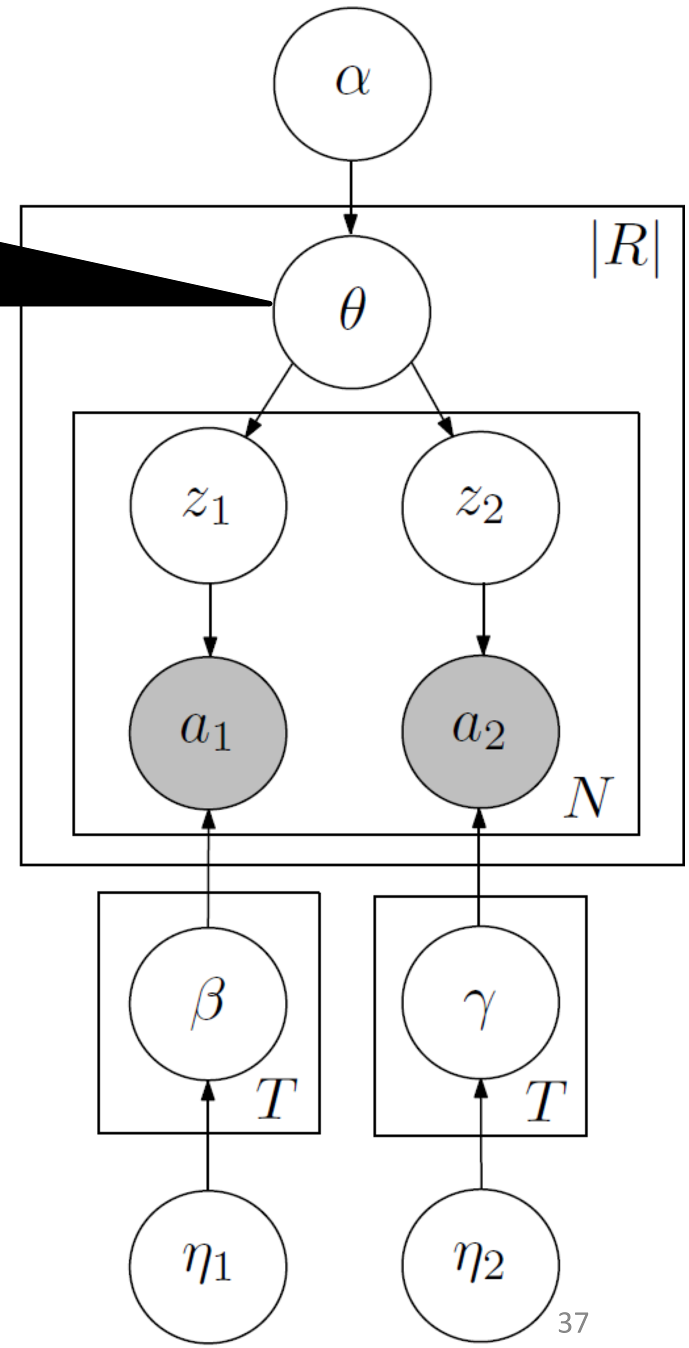


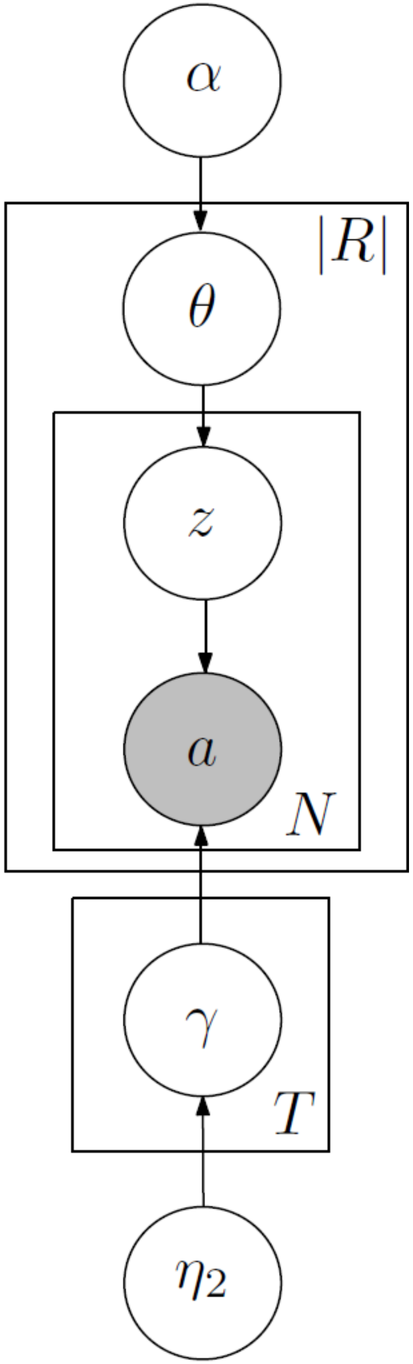
LinkLDA

[Erosheva et. al. 2004]



Both arguments share a distribution over topics



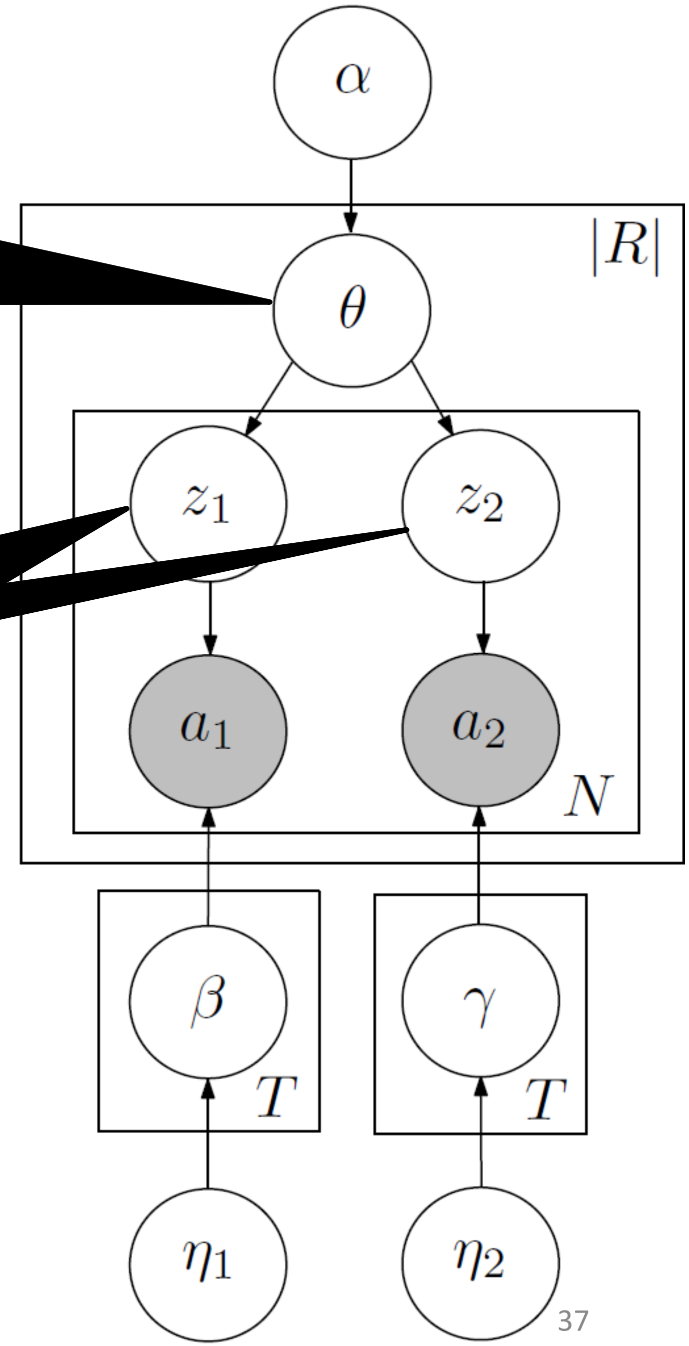


LinkLDA

[Erosheva et. al. 2004]

Both arguments share a distribution over topics

Likely that z1 = z2 (Both drawn from same distribution)



Example inferred topics

Topic t	Arg1	Relations which assign highest probability to t	Arg2
18	The residue - The mixture - The reaction mixture - The solution - the mixture - the reaction mixture - the residue - The reaction - the solution - The filtrate - the reaction - The product - The crude product - The pellet - The organic layer - Thereto - This solution - The resulting solution - Next - The organic phase - The resulting mixture - C.)	was treated with, is treated with, was poured into, was extracted with, was purified by, was diluted with, was filtered through, is disolved in, is washed with	EtOAc - CH2Cl2 - H2O - CH.sub.2Cl.sub.2 - H.sub.2O - water - MeOH - NaHCO3 - Et2O - NHCl - CHCl.sub.3 - NHCl - dropwise - CH2Cl.sub.2 - Celite - Et.sub.2O - Cl.sub.2 - NaOH - AcOEt - CH2C12 - the mixture - saturated NaHCO3 - SiO2 - H2O - N hydrochloric acid - NHCl - preparative HPLC - to0 C
151	the Court - The Court - the Supreme Court - The Supreme Court - this Court - Court - The US Supreme Court - the court - This Court - the US Supreme Court - The court - Supreme Court - Judge - the Court of Appeals - A federal judge	will hear, ruled in, decides, upholds, struck down, overturned, sided with, affirms	the case - the appeal - arguments - a case - evidence - this case - the decision - the law - testimony - the State - an interview - an appeal - cases - the Court - that decision - Congress - a decision - the complaint - oral arguments - a law - the statute
211	President Bush - Bush - The President - Clinton - the President - President Clinton - President George W. Bush - Mr. Bush - The Governor - the Governor - Romney - McCain - The White House - President - Schwarzenegger - Obama	hailed, vetoed, promoted, will deliver, favors, denounced, defended	the bill - a bill - the decision - the war - the idea - the plan - the move - the legislation - legislation - the measure - the proposal - the deal - this bill - a measure - the program - the law - the resolution - efforts - the agreement - gay marriage - the report - abortion
224	Google - Software - the CPU - Clicking - Excel - the user - Firefox - System - The CPU - Internet Explorer - the ability - Program - users - Option - SQL Server - Code - the OS - the BIOS	will display, to store, to load, processes, cannot find, invokes, to search for, to delete	data - files - the data - the file - the URL - information - the files - images - a URL - the information - the IP address - the user - text - the code - a file - the page - IP addresses - PDF files - messages - pages - an IP address

Event Type Induction

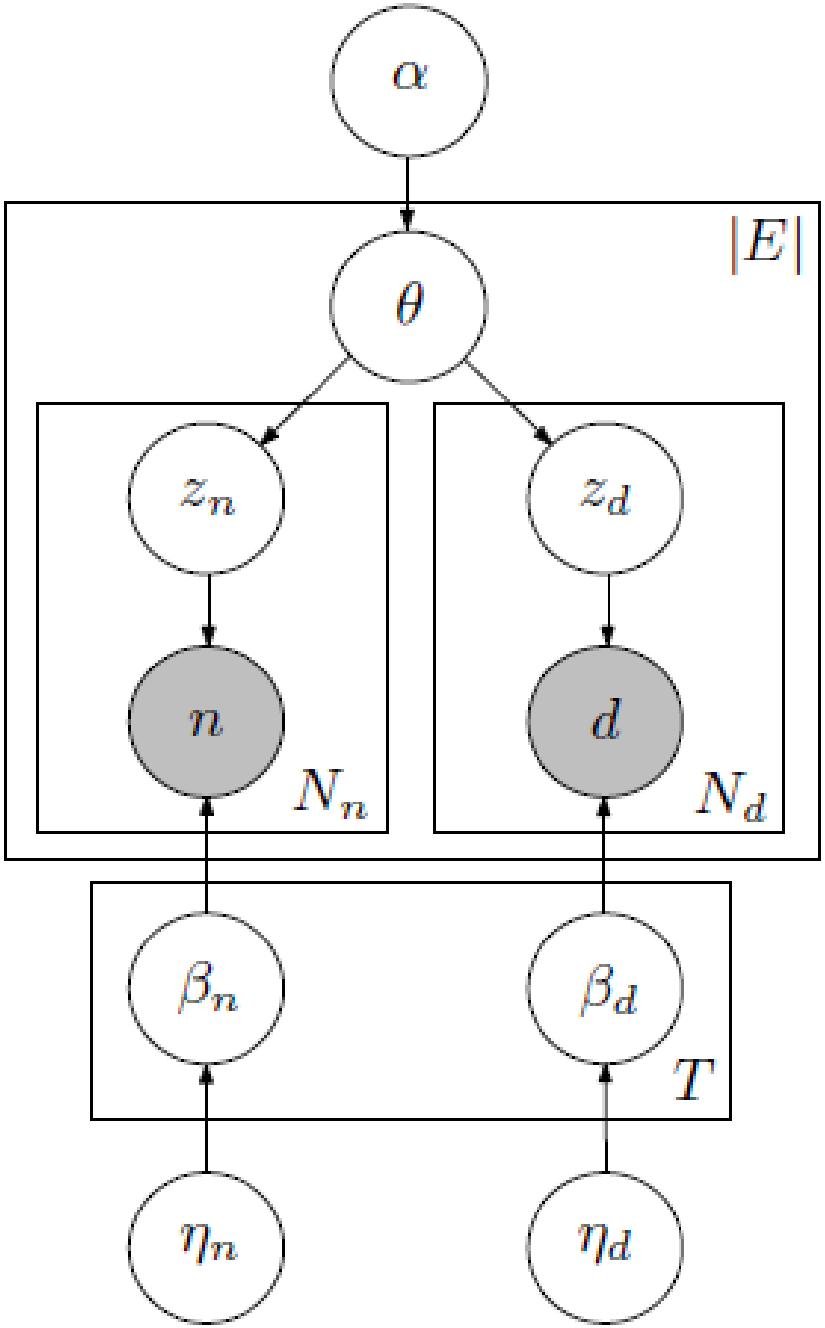
Entity	Event Phrase	Date
Steve Jobs	died	10/6/11
iPhone	announcement	10/4/11
GOP	debate	9/7/11
...	...	...

Event Type Induction

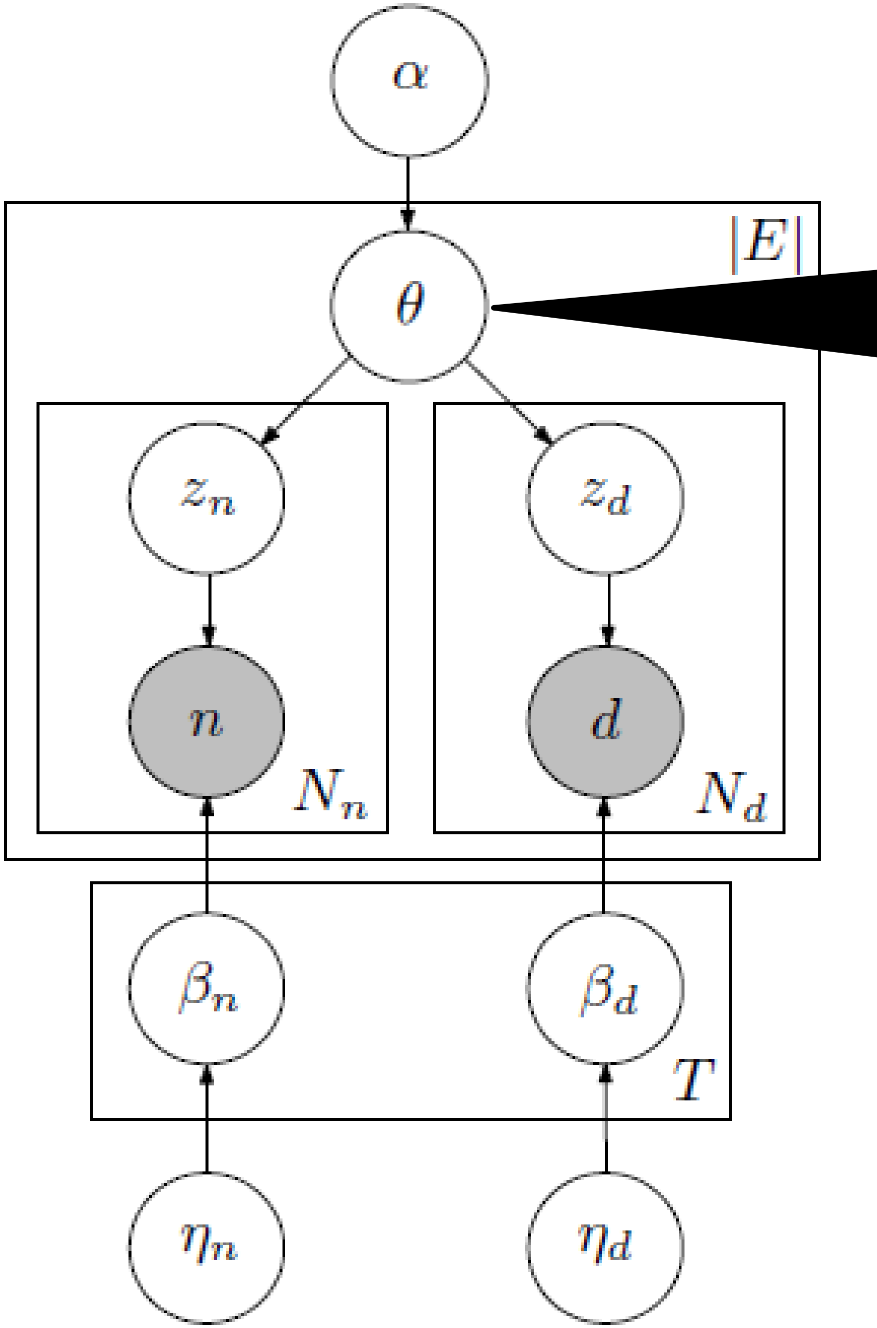
Entity	Event Phrase	Date	Type
Steve Jobs	died	10/6/11	DEATH
iPhone	announcement	10/4/11	PRODUCT LAUNCH
GOP	debate	9/7/11	POLITICAL EVENT
...	...	...	...

Classifying Events: Challenges

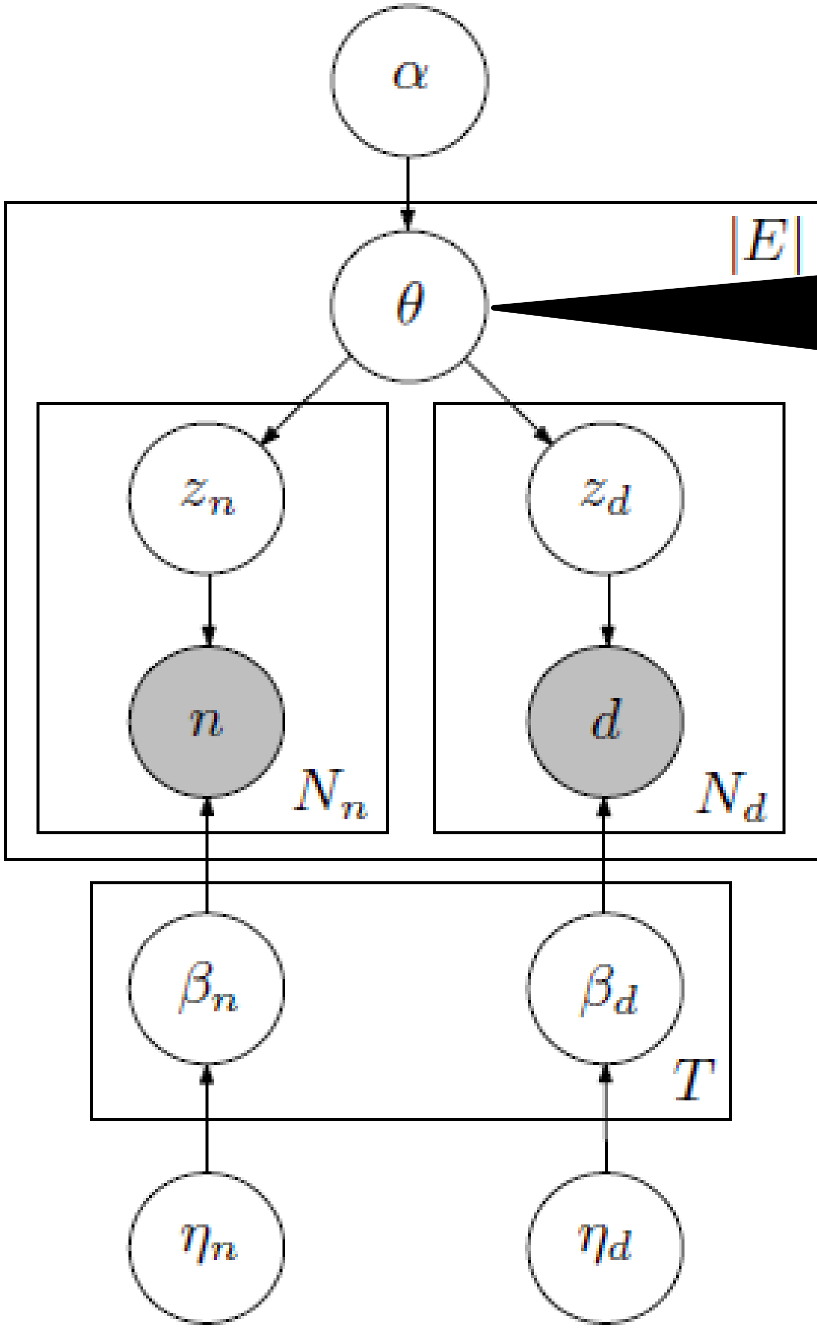
- Many Different Types
- Not sure what is the right set of types
- Set of types might change
  - Might start talking about different things
  - Might want to focus on different groups of users



α
$|E|$
θ
z_n
z_d
n
d
N_n
N_d
β_n
β_d
T
η_n
η_d



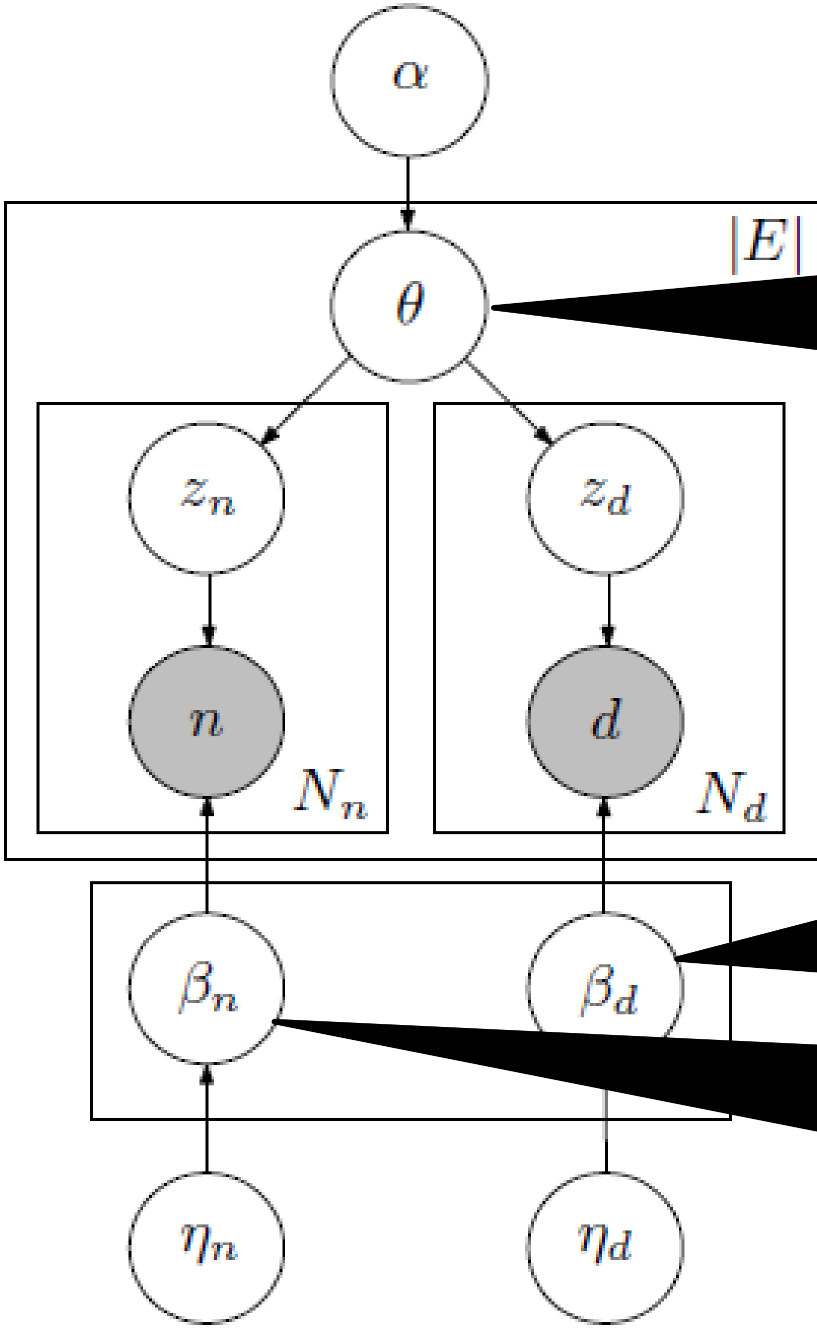
Each **Event Phrase** is modeled as a mixture of types



Each **Event Phrase** is modeled as a mixture of types

P(**SPORTS**|`cheered`)= 0.6

P(**POLITICS**|`cheered`)= 0.4



Each **Event Phrase** is modeled as a mixture of types

P(**SPORTS**|`cheered`)= 0.6

P(**POLITICS**|`cheered`)= 0.4

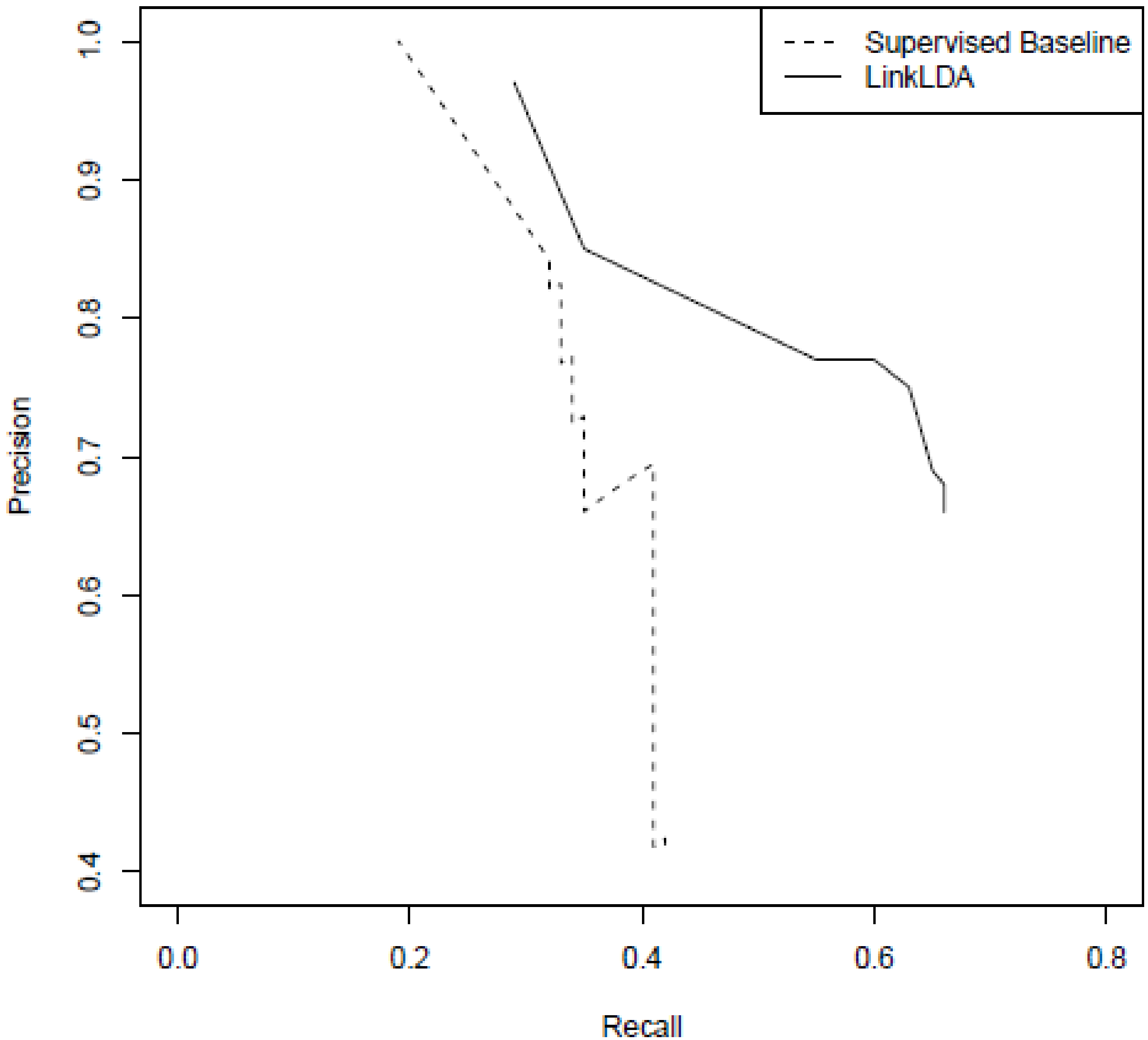
Each **Event Type** is Associated with a Distribution over Entities and Dates

Topic t	Entity	Event Phrases which assign highest probability to t	Date
60	obama, gop, anna, president obama, india, reuters, libya, congress, perry, cnn, israel, anna hazare, china, pakistan, delhi, washington, republican, senate, new delhi, rick perry, white house, america tripoli, #bahrain, romney, parliament, syria, ians, egypt, #syria, cairo, ron paul, gaddafi, #libya, ontario, iran, president barack obama, florida, united states	accused - address - addressing - advised - approved - arrested - attack - attacked - backs - bill - bombing - boost - briefing - campaign - charged - debate - discuss - elected - election - endorsed - fast - fasting - fired - meets - movement - plan - plea - poll - proposed - protests - reform - rejected - scam - seeks - sends - speaks - speech - talks - urged - violence - visits - vote - voted - voting - warned	20110922, 20110907, 20110912, 20110919, 20110913, 20110926, 20110927, 20110920, 20110928, 20110923, 20111004, 20111006, 20110908, 20110921, 20111003, 20110915, 20111005, 20110816, 20110906, 20110929, 20110826, 20110914, 20110825, 20110824, 20110817, 20111011, 20111007, 20110930, 20111017, 20110909, 20110827, 20110901
4	itunes, ipod, cole, drake, carter iv, pandora, wayne, cd, christmas, uk, ep, carter, unbroken, cole world, amazon, nirvana, nevermind, mv, j cole, mixtape, australia, blink, rihanna, walmart, midnight, lil wayne, japan, madden, blue slide park, kung fu panda 2, prince	album - bumping - cop - copy - download - downloaded - downloading - droppin - dropping - drops - leaked - preordered - produced - release - released - releases - releasing - single	20110927, 20110920, 20111004, 20110829, 20110926, 20110913, 20110919, 20110923, 20110921, 20110928, 20111003, 20110816, 20110916, 20111007, 20110922, 20110914, 20110912, 20111005, 20111024, 20110906, 20110929, 20110915, 20110924, 20110823
50	england, eagles, arsenal, nfl, chelsea, espn, yankees, ireland, wales, red sox, packers, stoke, michigan, dallas, lsu, saints, jets, united, good luck, detroit, spurs, barcelona, liverpool, rangers, colts, auburn, falcons, scotland, miami, cowboys, lions, bolton, tigers, ravens, alabama, rugby, mlb, the lions, phillies, bears, romo, the game, florida, redskins, manchester united, france, college, texas, tom brady, clemson, ortiz, torres	#football - action - assist - baseball - beat - beaten - beating - beats - betting - bounce back - career - cheer - cheering - choke - clash - clinch - clinched - coaching - collapsed - combined - comeback - completed - conceded - crush - crushing - debut - defeat - defeated - defending - destroy - destroyed - dissapointing - dominate - dominated - dominating - draft - draw - drew - eliminated - facing - football - game - gameday - games - goals - great result - great win - hopping - improve - injured - injury - inning - kick	20110917, 20110924, 20110918, 20110925, 20110910, 20110911, 20111001, 20111002, 20110923, 20110916, 20111007, 20110926, 20111009, 20111008, 20110912, 20110903, 20110919, 20110927, 20110902, 20110908, 20110828, 20110920, 20110901, 20110928, 20110909, 20110930, 20110827, 20110921, 20110906, 20110913, 20111003, 20110922, 20110915, 20110826, 20110914, 20111006, 20111004, 20110821, 20110825, 20110904
79	jersey shore, glee, netflix, uk, america, mtv, abc, fox, nbc, true blood, dexter, big brother, hbo, cbs, vmas, charlie sheen, espn, vma, vampire diaries, bbc, x factor, harry potter, nfl, twilight, bb, grey, dvr, sons of anarchy, sky, big bang, terra nova, towie	air - airing - airs - breaking - cast - caught up - documentary - episode - new season - premier - premiere - premiered - premieres - premiering - premiers - returns - season - seasons - series - shore - shown - starts - stream - tuning - watch - watched - watching	20110919, 20110922, 20110925, 20110920, 20110923, 20110926, 20110924, 20110921, 20110915, 20110828, 20111002, 20110912, 20110917, 20110913, 20110918, 20110927, 20111004, 20110914, 20110928, 20111003, 20110829, 20111005

Experiment: Categorizing Events

- Randomly Sampled 100 (entity, date) pairs
- Annotated with event types
  - Using types discovered by the topic model
- Baseline:
  - Supervised classification using 10-fold cross validation
  - Treat event phrases like bag of words

Event Classification Performance



Named Entity Classification in Twitter: Challenges

- Plethora of distinctive, **infrequent** types
  - Bands, Movies, Products, etc…
  - Very Little training data for these
  - Can't simply rely on supervised classification
- Very terse (often contain insufficient context)

Named Entity Classification in Twitter: Challenges

- Plethora of distinctive, **infrequent** types
  - Bands, Movies, Products, etc…
  - Very Little training data for these
  - Can’t simply rely on supervised classification
- Very terse (often contain insufficient context)

KKTNY in 45min..........

Weakly Supervised NE Classification

(Collins and Singer 99) (Etzioni et. al. 05) (Kozareva 06)

- Freebase lists provide a source of supervision
- But entities often appear in many different lists, for example “China” could be:
  - A country
  - A band
  - A person (member of the band “metal boys”)
  - A film (released in 1943)

Weakly Supervised NE Classification

(Collins and Singer 99) (Etzioni et. al. 05) (Kozareva 06)

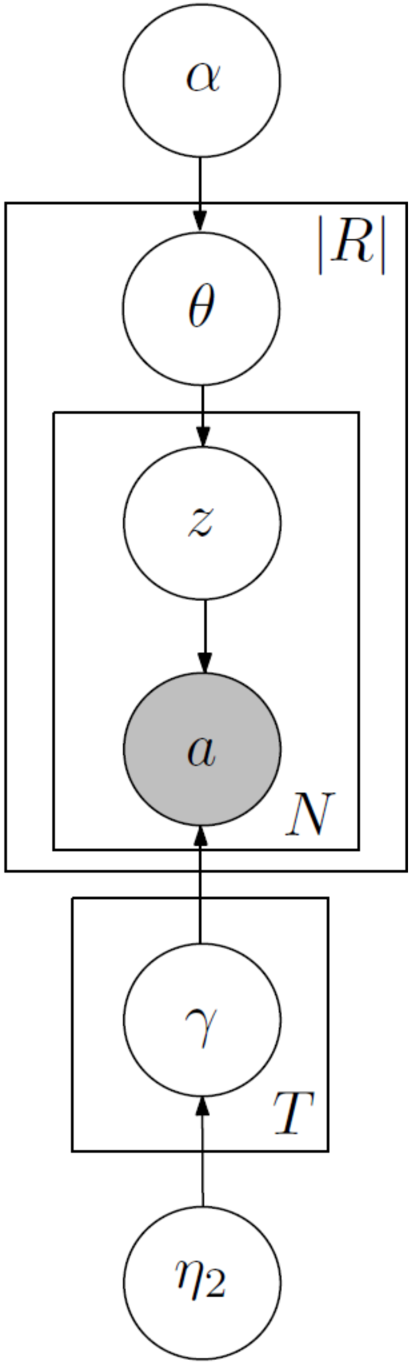
- Freebase lists provide a source of supervision
- But entities often appear in many different lists, for example "China" could be:
  - A country
  - A band
  - A person (member of t
  - A film (released

We need Some way to disambiguate

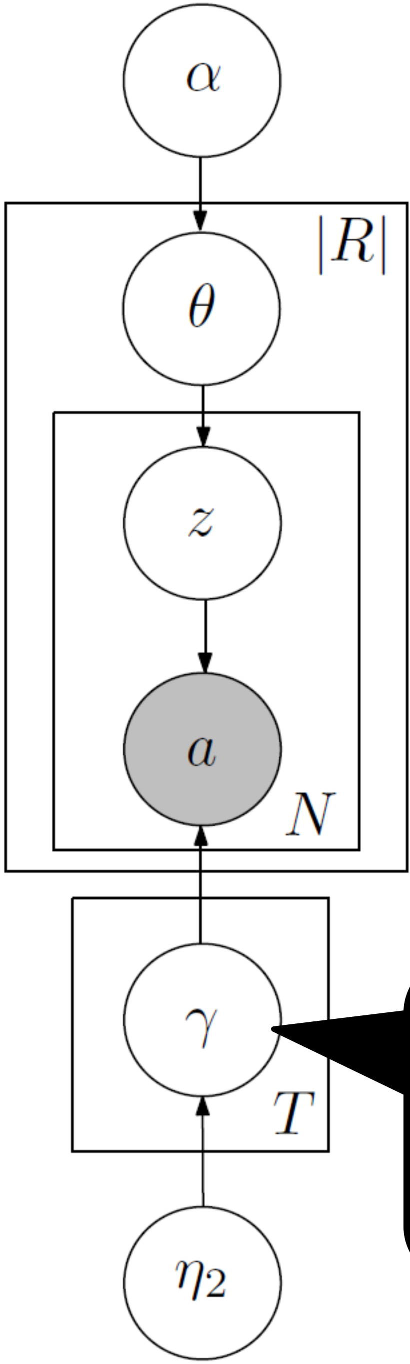
Distant Supervision With Topic Models

- Treat each entity as a “document”
  - Words in document are those which co-occur with entity
- LabeledLDA **(Ramage et. al. 2009)**
  - Constrained Topic Model
  - Each entity is associated with a distribution over topics
    - Constrained based on FB dictionaries
  - Each topic is associated with a type (in Freebase)

Generative Story

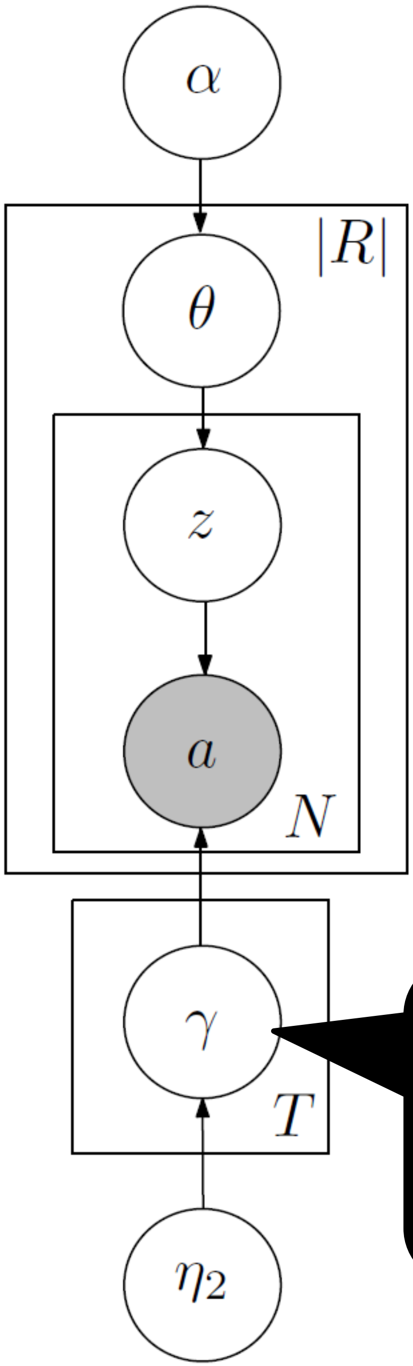


Generative Story



For each type, pick a random distribution over words

Generative Story

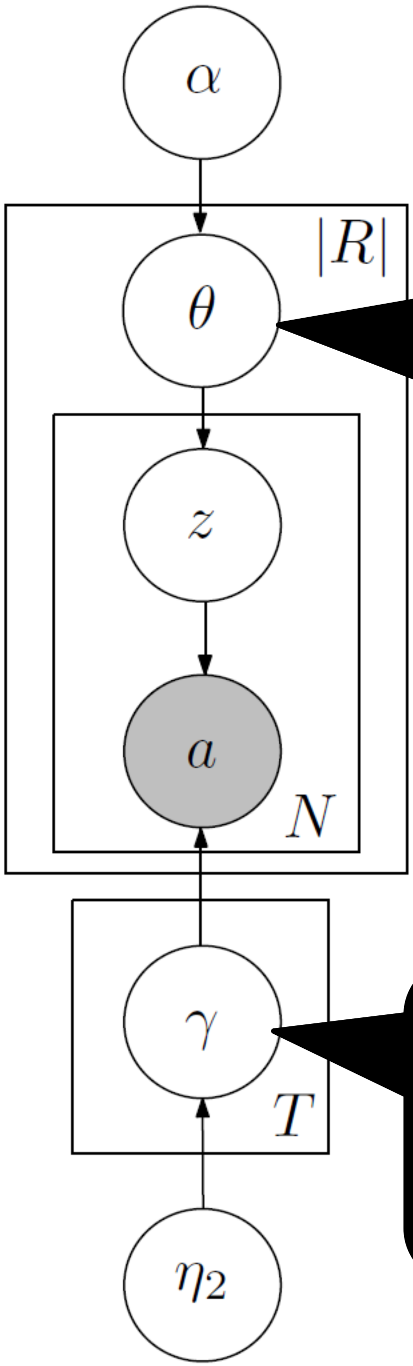


For each type, pick a random distribution over words

Type 1: **TEAM**
P(victory|T1)= 0.02
P(played|T1)= 0.01
…

Type 2: **LOCATION**
P(visiting|T2)=0.05
P(airport|T2)=0.02
…

Generative Story



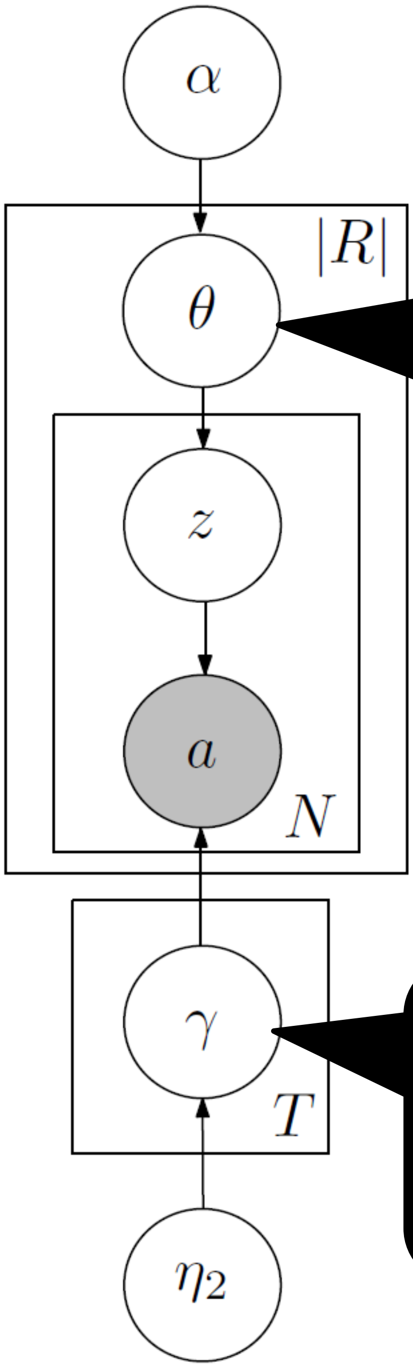
For each entity, pick a distribution over types (constrained by Freebase)

For each type, pick a random distribution over words

Type 1: **TEAM**
P(victory|T1)= 0.02
P(played|T1)= 0.01
…

Type 2: **LOCATION**
P(visiting|T2)=0.05
P(airport|T2)=0.02
…

Generative Story



For each entity, pick a distribution over types (constrained by Freebase)

Seattle

P(TEAM|**Seattle**)= 0.6

P(LOCATION|**Seattle**)= 0.4

For each type, pick a random distribution over words

Type 1: TEAM

P(victory|T1)= 0.02

P(played|T1)= 0.01

…

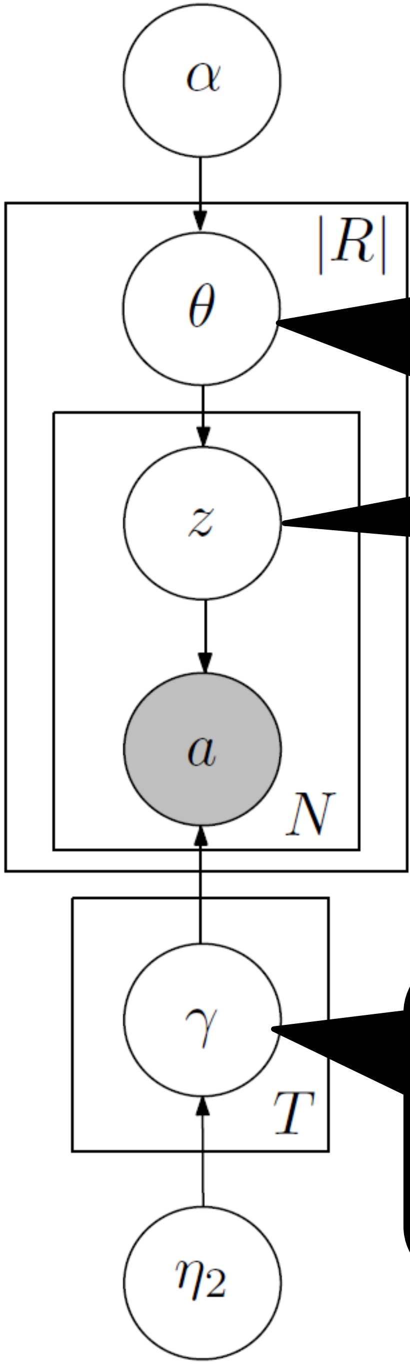
Type 2: LOCATION

P(visiting|T2)=0.05

P(airport|T2)=0.02

…

Generative Story



For each entity, pick a distribution over types (constrained by Freebase)

For each position, first pick a type

For each type, pick a random distribution over words

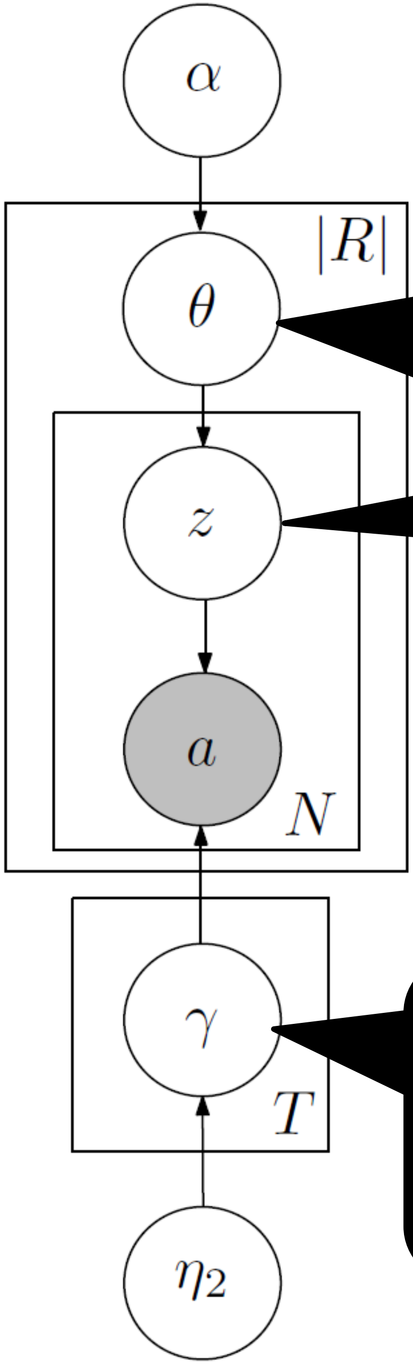
`Seattle`

P(TEAM|`Seattle`)= 0.6
P(LOCATION|`Seattle`)= 0.4

Type 1: TEAM
P(victory|T1)= 0.02
P(played|T1)= 0.01
…

Type 2: LOCATION
P(visiting|T2)=0.05
P(airport|T2)=0.02
…

Generative Story



For each entity, pick a distribution over types (constrained by Freebase)

Seattle
P(TEAM|**Seattle**)= 0.6
P(LOCATION|**Seattle**)= 0.4

For each position, first pick a type

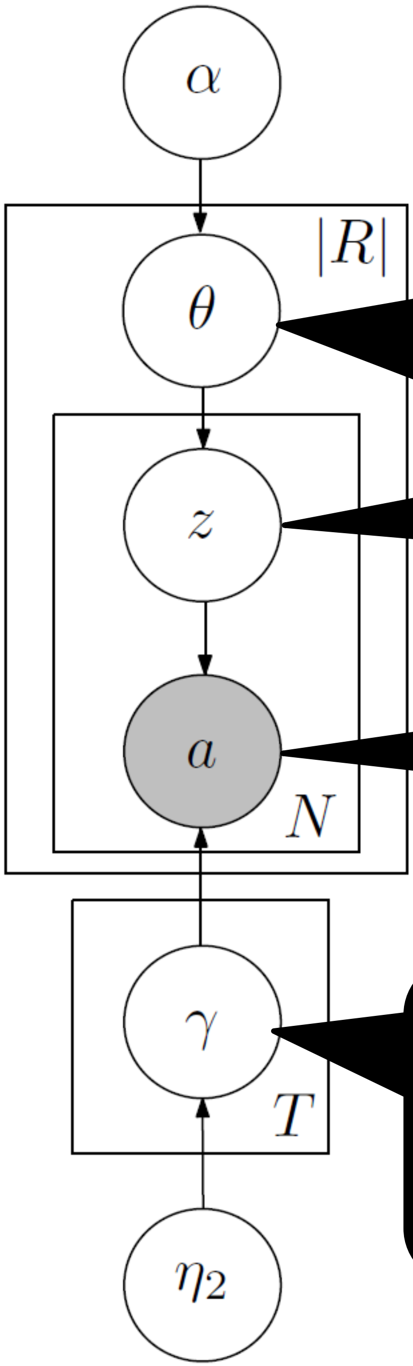
Is a TEAM

For each type, pick a random distribution over words

Type 1: TEAM
P(victory|T1)= 0.02
P(played|T1)= 0.01
…

Type 2: LOCATION
P(visiting|T2)=0.05
P(airport|T2)=0.02
…

Generative Story



For each entity, pick a distribution over types (constrained by Freebase)

For each position, first pick a type

Then pick an word based on type

For each type, pick a random distribution over words

`Seattle`

P(TEAM|`Seattle`)= 0.6

P(LOCATION|`Seattle`)= 0.4

`Is a` TEAM

Type 1: **TEAM**

P(victory|T1)= 0.02

P(played|T1)= 0.01

…

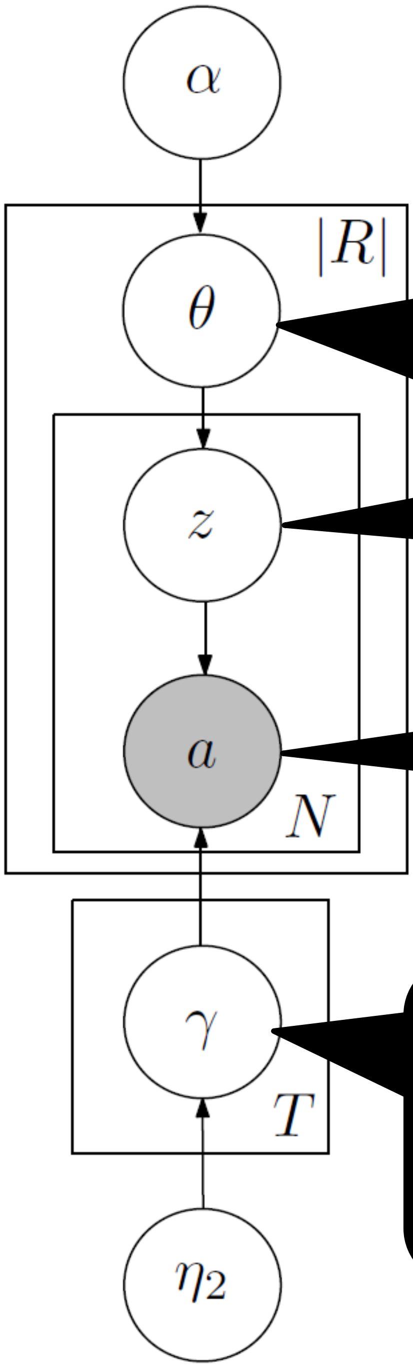
Type 2: **LOCATION**

P(visiting|T2)=0.05

P(airport|T2)=0.02

…

Generative Story



For each entity, pick a distribution over types (constrained by Freebase)

For each position, first pick a type

Then pick an word based on type

For each type, pick a random distribution over words

`Seattle`

P(TEAM|`Seattle`)= 0.6

P(LOCATION|`Seattle`)= 0.4

`Is a` TEAM

victory

Type 1: TEAM

P(victory|T1)= 0.02

P(played|T1)= 0.01

…

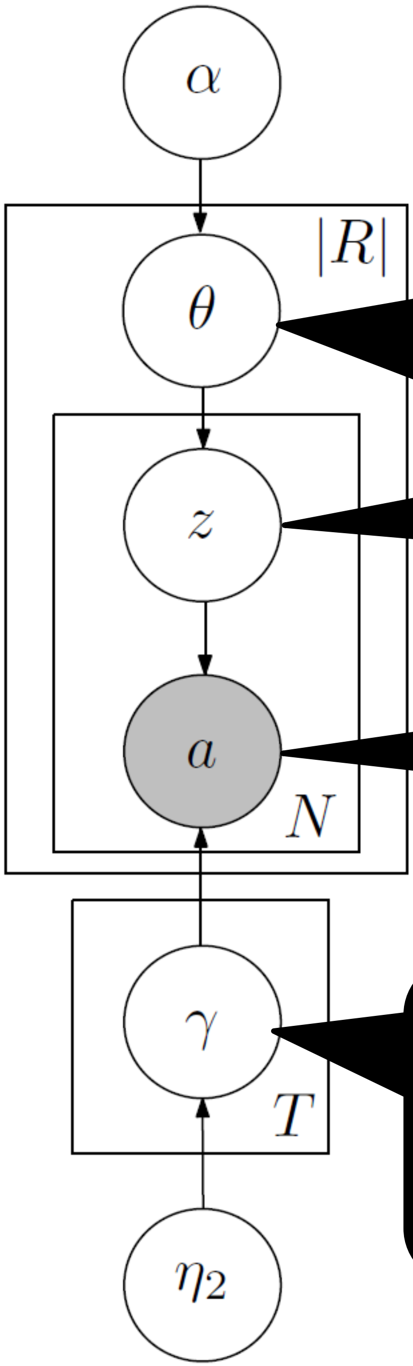
Type 2: LOCATION

P(visiting|T2)=0.05

P(airport|T2)=0.02

…

Generative Story



For each entity, pick a distribution over types (constrained by Freebase)

For each position, first pick a type

Then pick an word based on type

For each type, pick a random distribution over words

`Seattle`

P(TEAM|`Seattle`)= 0.6

P(LOCATION|`Seattle`)= 0.4

`Is a` TEAM

`Is a` LOCATION

victory

Type 1: TEAM

P(victory|T1)= 0.02

P(played|T1)= 0.01

…

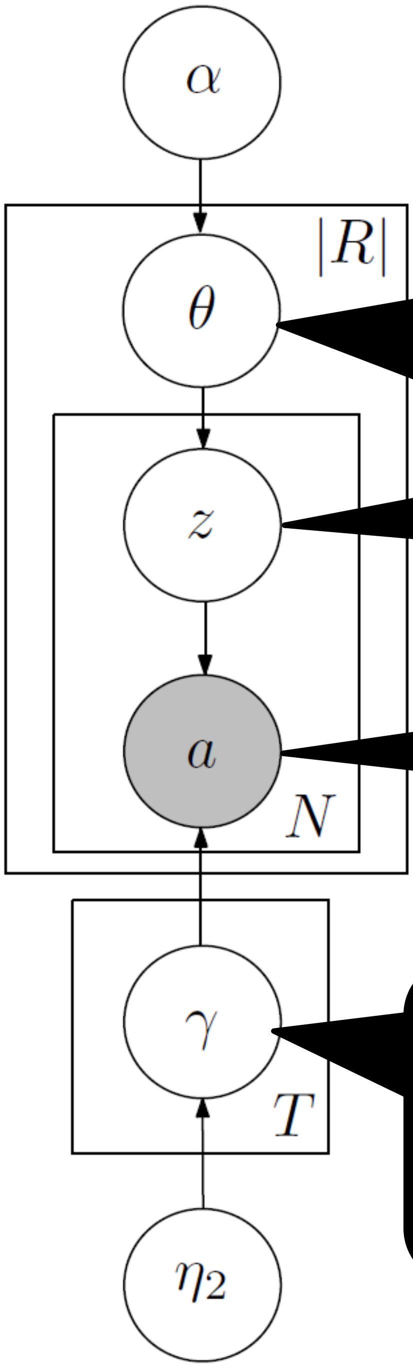
Type 2: LOCATION

P(visiting|T2)=0.05

P(airport|T2)=0.02

…

Generative Story



For each entity, pick a distribution over types **(constrained by Freebase)**

For each position, first pick a type

Then pick an word based on type

For each type, pick a random distribution over words

`Seattle`

P(**TEAM**|`Seattle`)= 0.6

P(**LOCATION**|`Seattle`)= 0.4

`Is a` **TEAM** `Is a` **LOCATION**

victory airport

Type 1: **TEAM**

P(victory|T1)= 0.02

P(played|T1)= 0.01

…

Type 2: **LOCATION**

P(visiting|T2)=0.05

P(airport|T2)=0.02

…

Type Lists

Type	Top 20 Entities not found in Freebase dictionaries
PRODUCT	nintendo ds lite, apple ipod, generation black, ipod nano, apple iphone, gb black, xperia, ipods, verizon media, mac app store, kde, hd video, nokia n8, ipads, iphone/ipod, galaxy tab, samsung galaxy, playstation portable, nintendo ds, vpn
TV-SHOW	pretty little, american skins, nof, order svu, greys, kktny, rhobh, parks & recreation, parks & rec, dawson 's creek, big fat gypsy weddings, big fat gypsy wedding, winter wipeout, jersey shores, idiot abroad, royle, jerseyshore, mr . sunshine, hawaii five-0, new jersey shore
FACILITY	voodoo lounge, grand ballroom, crash mansion, sullivan hall, memorial union, rogers arena, rockwood music hall, amway center, el mocambo, madison square, bridgestone arena, cat club, le poisson rouge, bryant park, mandalay bay, broadway bar, ritz carlton, mgm grand, olympia theatre, consol energy center

Type Lists

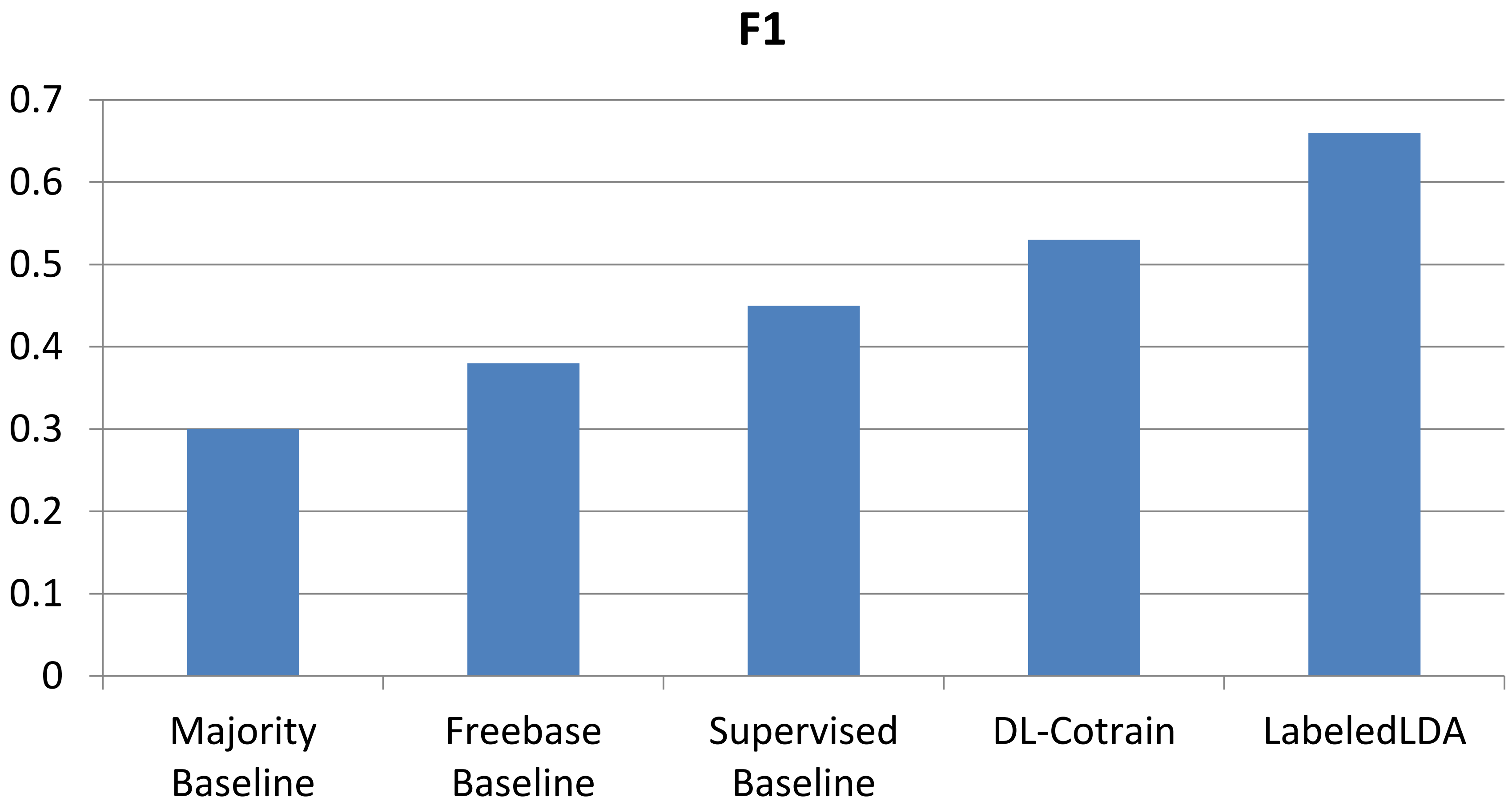
Type	Top 20 Entities not found in Freebase dictionaries
PRODUCT	nintendo ds lite, apple ipod, generation black, ipod nano, apple iphone, gb black, xperia, ipods, verizon media, mac app store, kde, hd video, nokia n8, ipads, iphone/ipod, galaxy tab, samsung galaxy, playstation portable, nintendo ds, vpn
TV-SHOW	pretty little, american skins, nof, order svu, greys, kktny, rhobh, parks & recreation, parks & rec, dawson 's creek, big fat gypsy weddings, big fat gypsy wedding, winter wipeout, jersey shores, idiot abroad, royle, jerseyshore, mr . sunshine, hawaii five-0, new jersey shore
FACILITY	voodoo lounge, grand ballroom, crash mansion, sullivan hall, memorial union, rogers arena, rockwood music hall, amway center, el mocambo, madison square, bridgestone arena, cat club, le poisson rouge, bryant park, mandalay bay, broadway bar, ritz carlton, mgm grand, olympia theatre, consol energy center

- KKTNY = **Kourtney and Kim Take New York**
- RHOBH = **Real Housewives of Beverly Hills**

Evaluation

- Manually Annotated the 2,400 tweets with the 10 entity types
  - Only used for testing purposes
  - No labeled examples for LLDA & Cotraining

Classification Results: 10 Types (Gold Segmentation)



Summary

- Latent Variable Models can be Useful If:
  - Your problem has many classes
  - You don’t have good labeled data
  - Unlabeled data is plentiful
- Automatically induce useful structure
  - Generative models useful for many tasks
  - Principled way to infer latent structure using only a large sample of data