Topic modeling

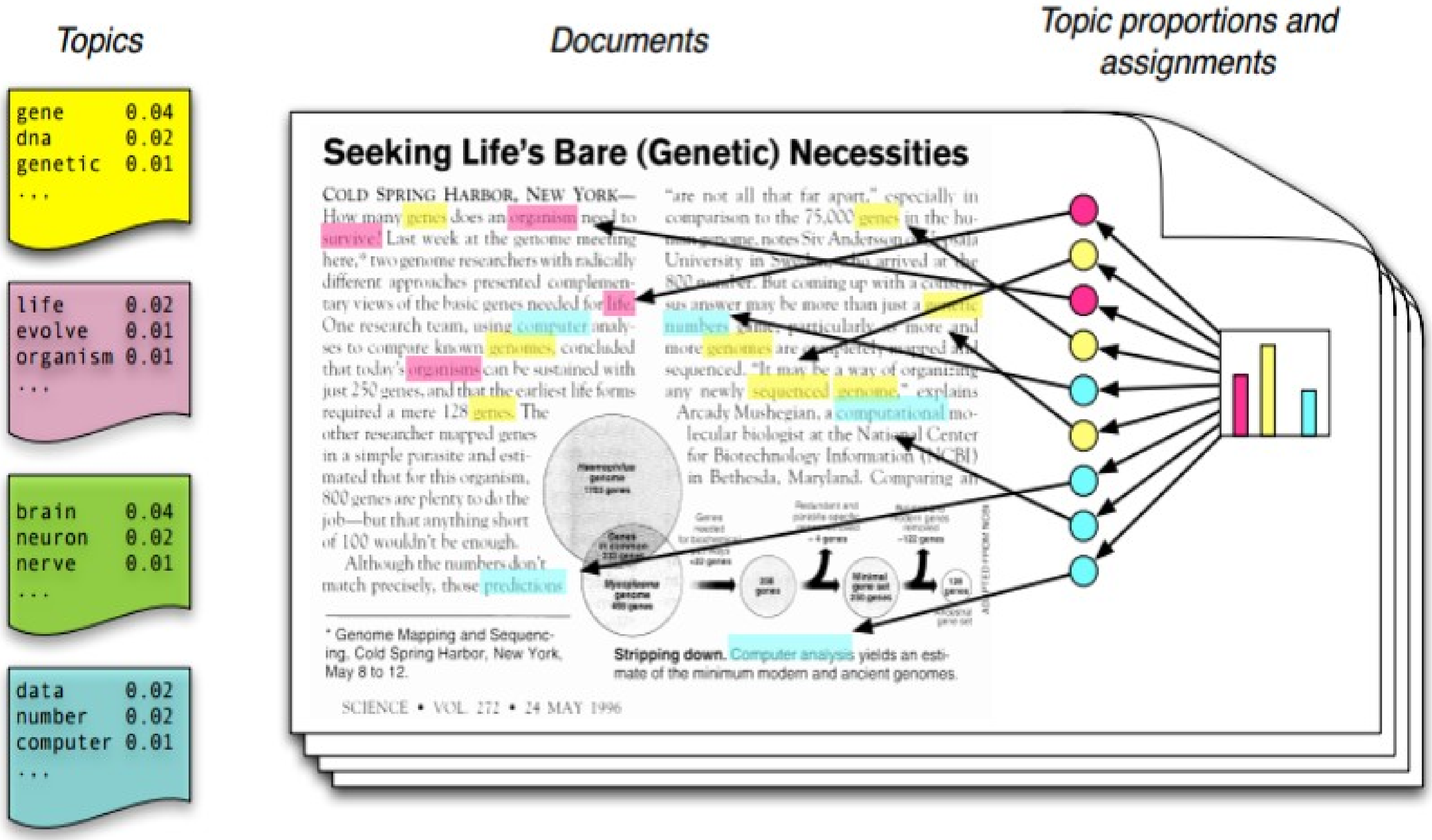
Tag documents automatically

- Supervised learning
  - Document classification
- Unsupervised learning
  - Topic models
- Semi-supervised learning

Tag documents automatically

- Wide range of applications
  - Email filtering
  - Document retrieval
  - Sentiment analysis
  - Language identification
- Wide range of challenges
  - Tag structure?
  - Single tag or multiple tags?
  - How many tags – allow new tags?
  - Short documents

Topic Model



1) associate words (terms) with topics, then

2) associate topics with documents

Latent Dirichlet Allocation

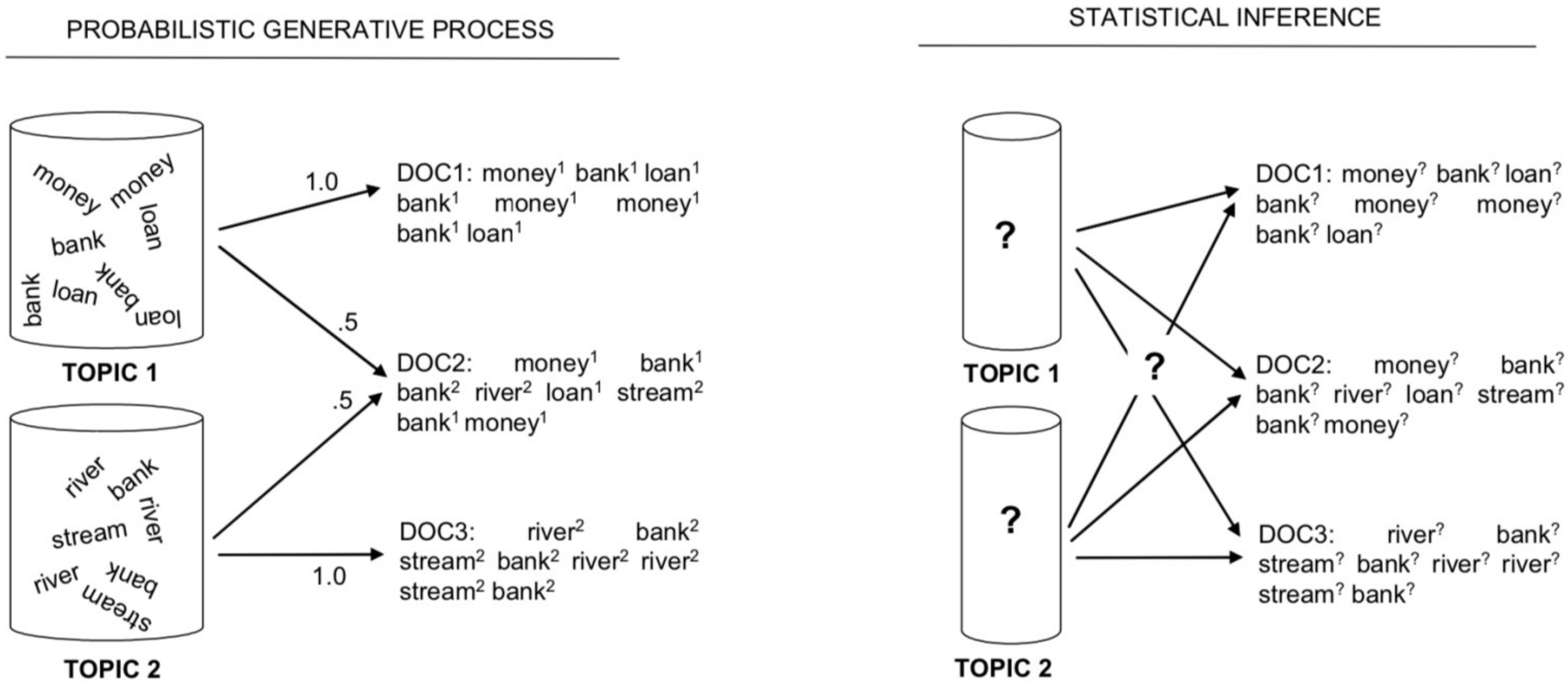
- DM Blei, AY Ng, MI Jordan - Journal of machine Learning research, 2003 - jmlr.org
- Cited by 58742 (Google Scholar, Sep 26, 2025)

- Distributed representations of words and phrases and their compositionality (the original word2vec paper), 2013, 48528 citation
- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (the original BERT paper), 2018, 145045 citation

Learning Task

- Given: observed words in a corpus
- Task: learn what topic model has generated the data (corpus)
- this means we have to infer the
  - probability distribution over words associated with each topic,
  - the distribution over topics for each document, and
  - the topic responsible for generating each word

LDA: a generative model

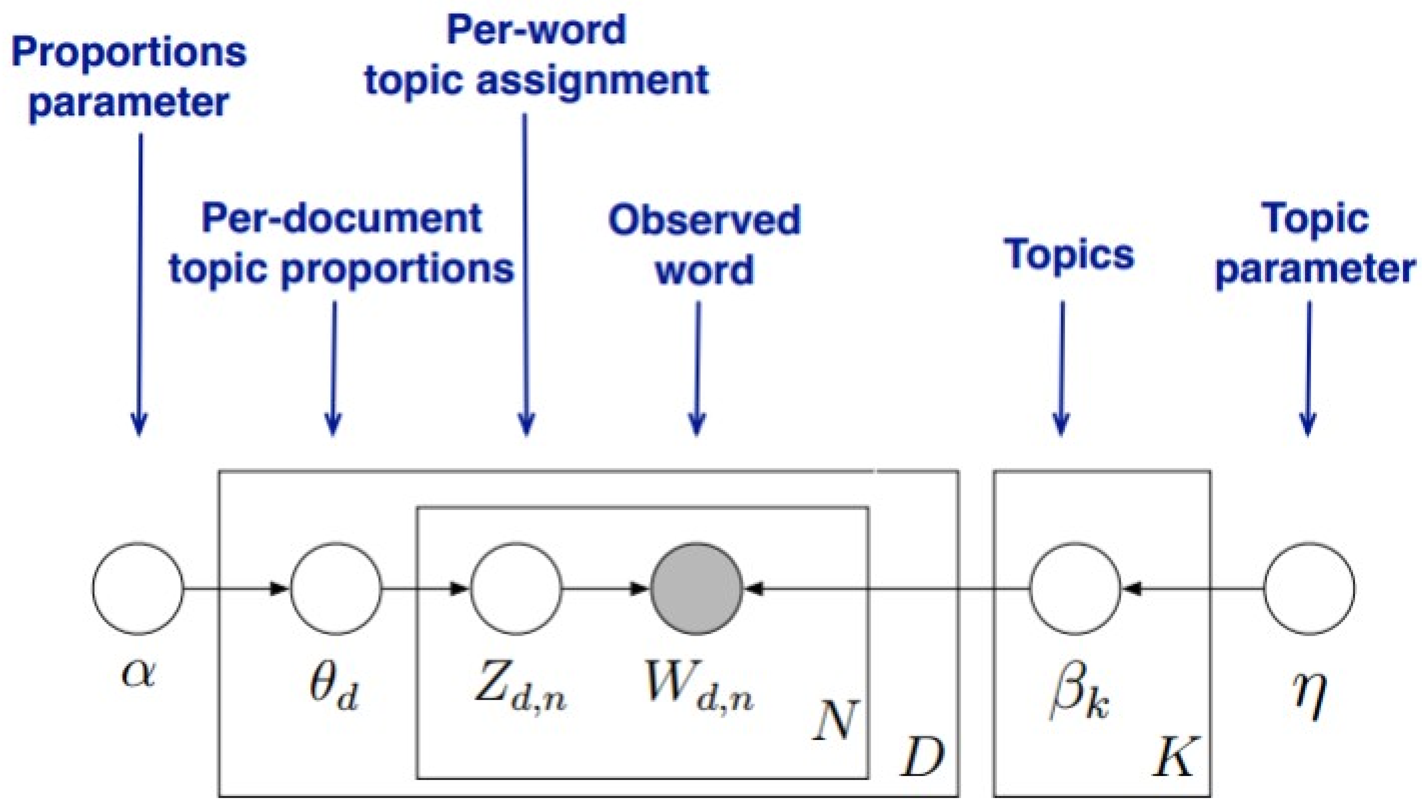


An intuitive example

- Suppose I have a coin
- what is ? Bias
  - E.g.,
- I flip this coin 30 times and observe (#head, #tail)=(20, 10)
- What is the ? Bias after observe experiments

Model Definition

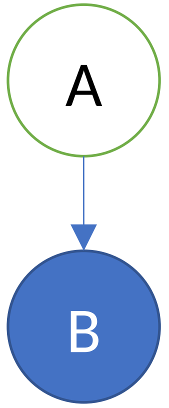
p(topics

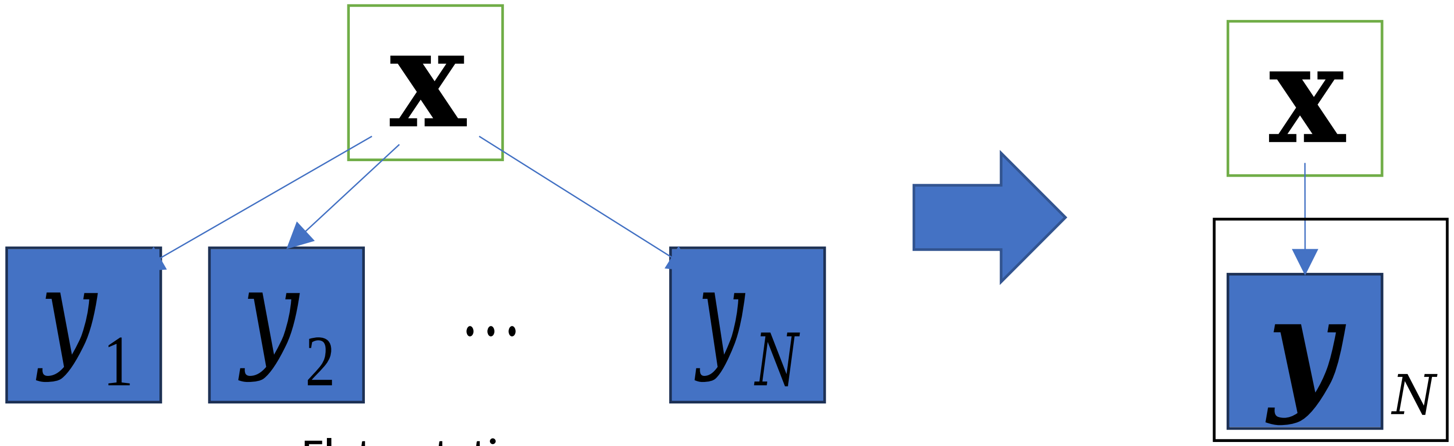


uments)

Probabilistic Graphic Model: Plate Notations

- Represent variables that repeat in a graphical model
- Variables
  - A solid (or shaded) circle means the corresponding variable is *observed*; otherwise it is *hidden*
- Dependency among variables:
  - A Directed Acyclic Graphical (DAG) model
- Using plate notation instead of flat notation

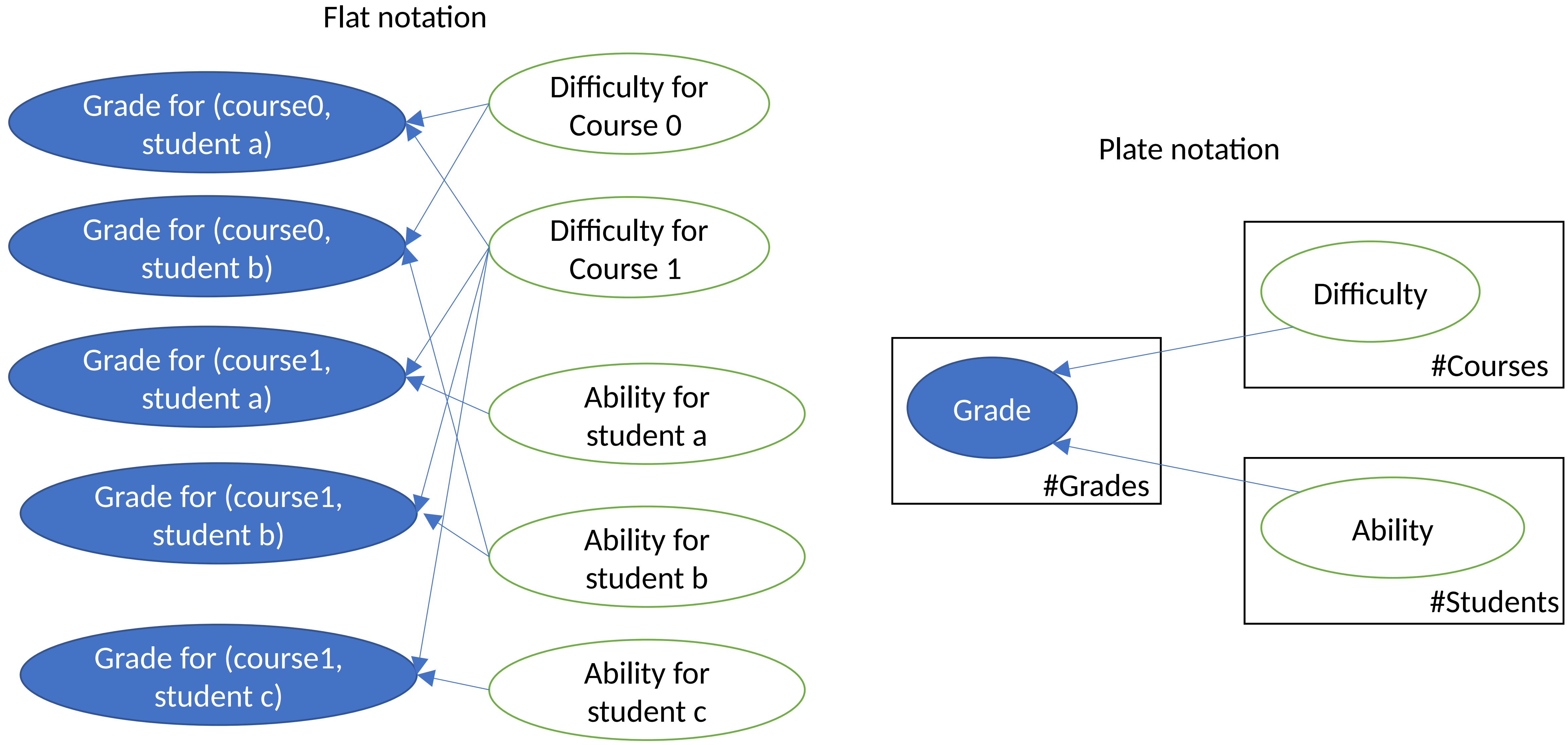




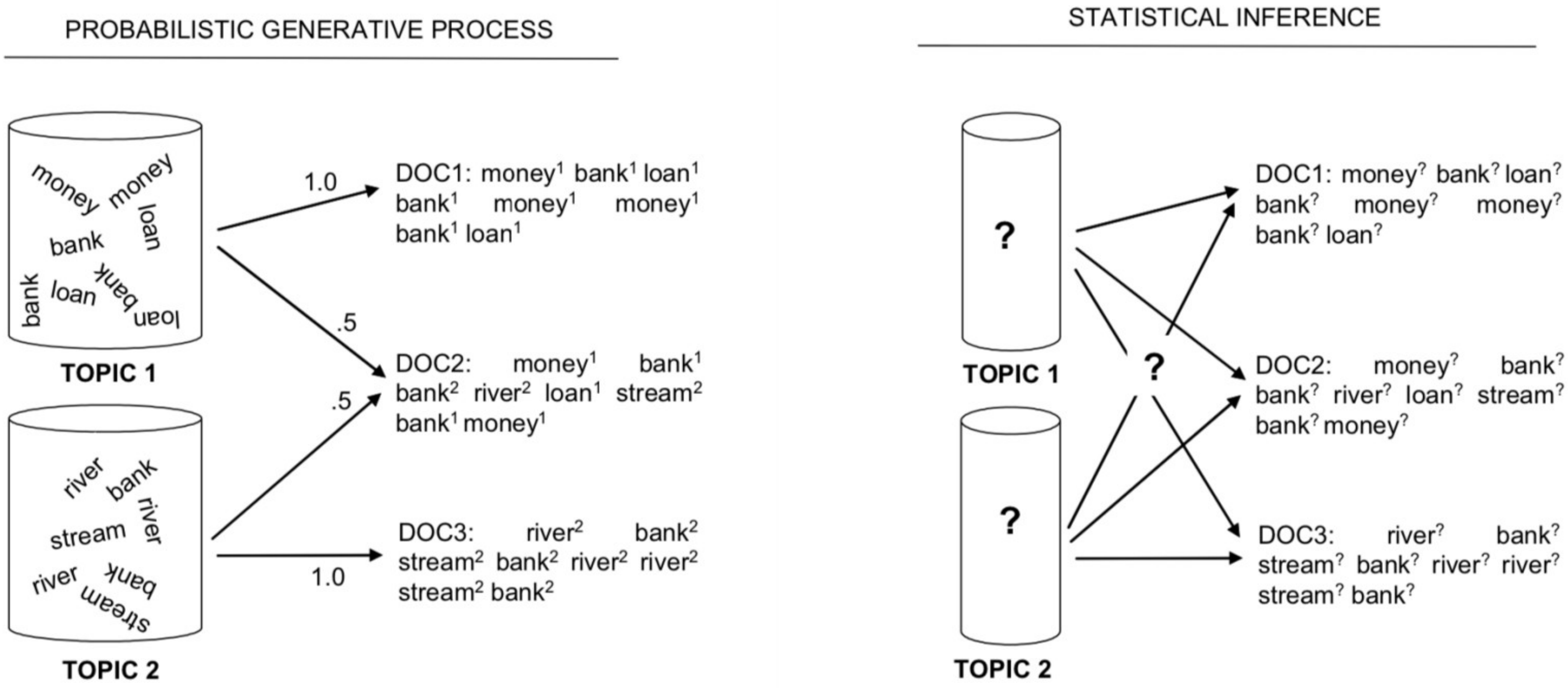
Flat notation

Plate notation

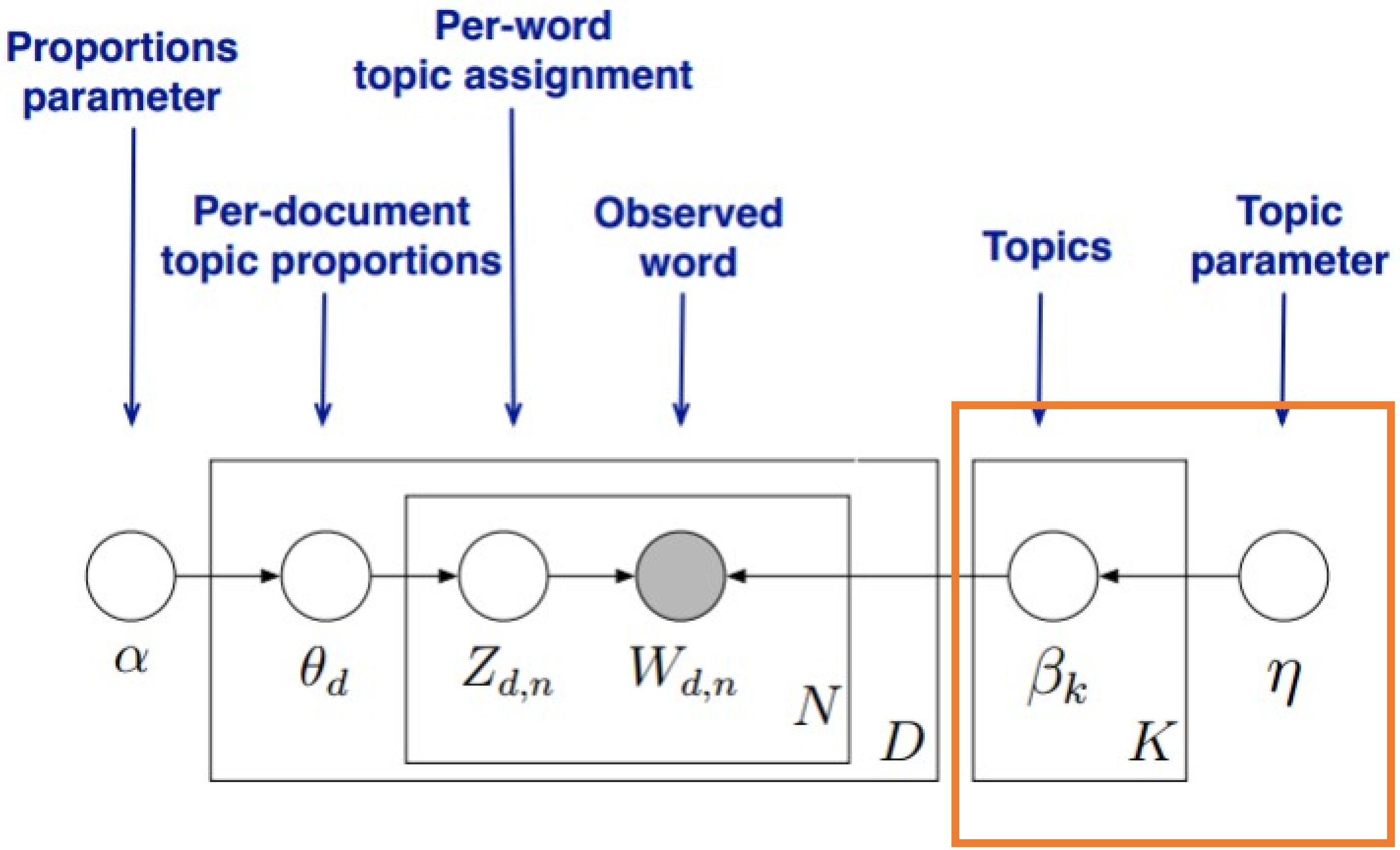
An Example of Plate Notation



LDA: a generative model



Generative process

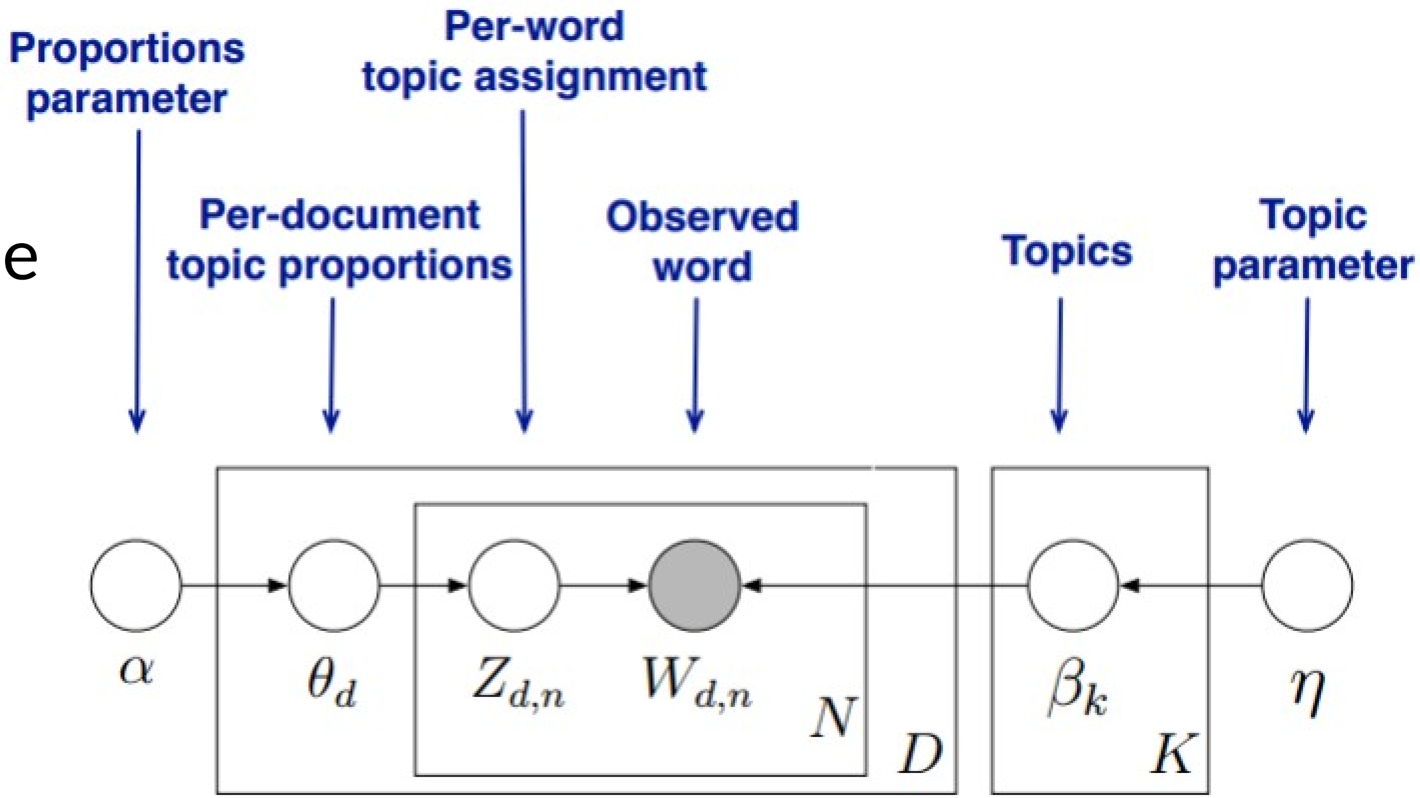


- Draw each topic for

Generative process

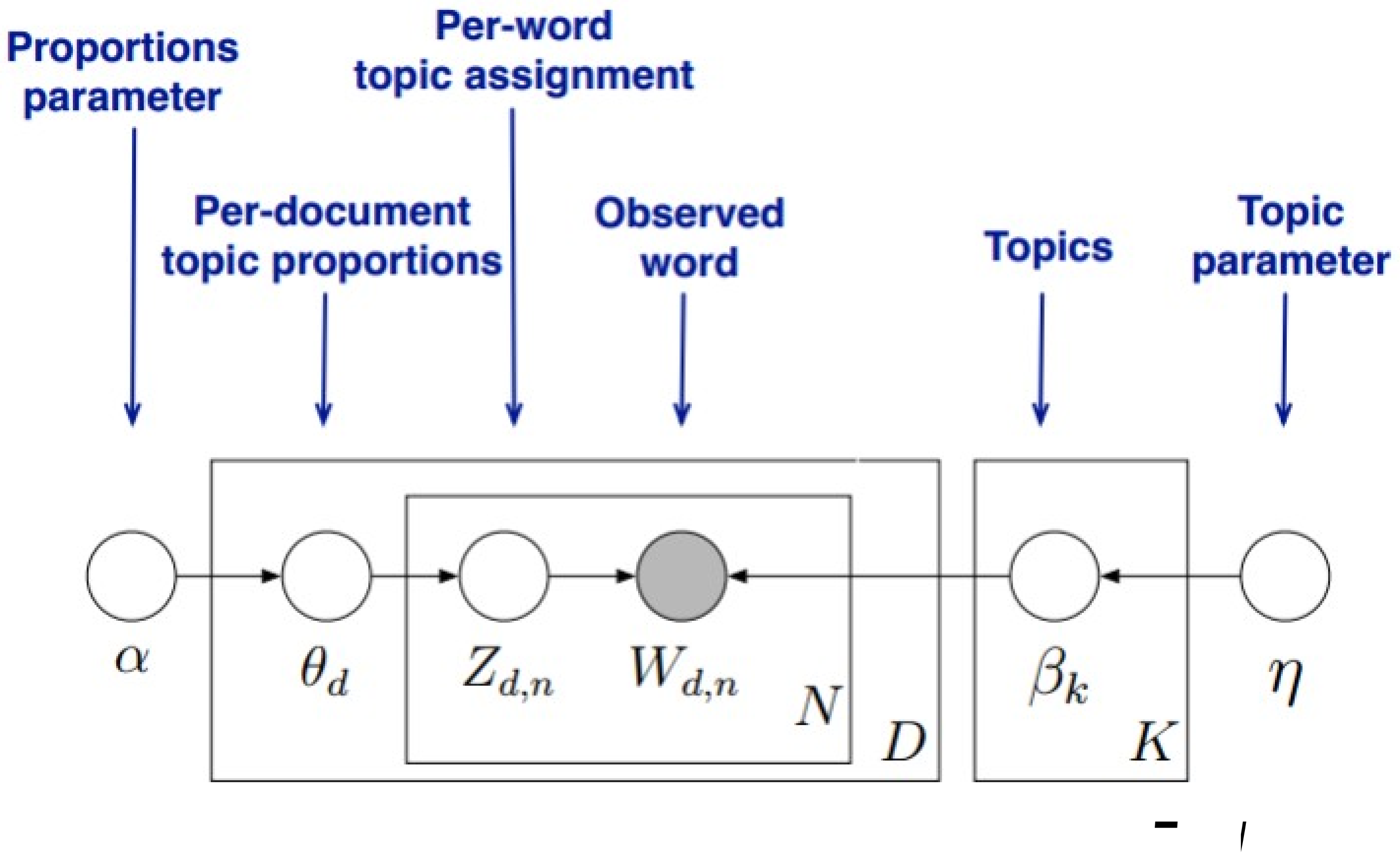
- For each document:

1. Draw topic proportions
2. For each word within the document:
    1) Draw
    2) Draw



Inference

$p(\beta, \theta$



Prior and posterior

- Given evidence, estimate the parameters

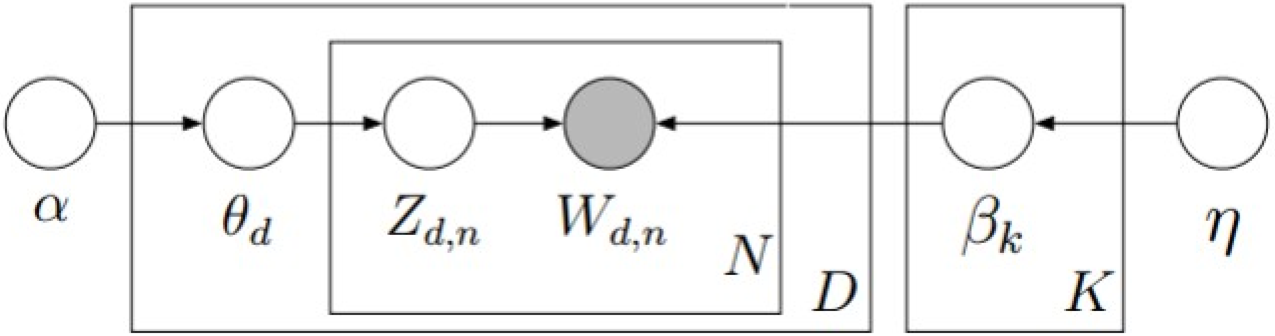
- In the next couple slides, I will use to denote all parameters, to denote all evidence (observed)
- Prior:
- Posterior:

$$p(\theta|x) = \frac{p(x|\theta)}{p(x)} p(\theta)$$

- : likelihood

Conjugate

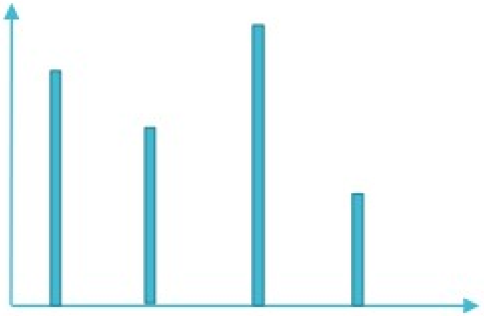
- In Bayesian probability theory, if the posterior distribution is in the same probability distribution family as the prior probability distribution , the prior and posterior are then called **conjugate distributions,** and the prior is called a **conjugate prior** for the likelihood function .



$(\beta_d|\eta) \sim Dir(\beta)$ $\quad$ $(\theta_d|\alpha) \sim Dir(\alpha)$ $\quad$ $Z_{d,n} \sim Multi(\theta_d)$ $\quad$ $W_{d,n} \sim Multi(\beta_{z_{d,n}})$

$p(z_{d,n}|\theta_d) = \theta_{d,z_{d,n}}$ $\quad$ $p(w_{d,n}|z_{d,n}, \beta_{1:K}) = \beta_{z_{d,n},w_{d,n}}$

θ_d:



β:

	Word probabilities for each topic		
Topics			

Multinomial and Dirichlet distributions

- Multinomial: the probability of counts of each side for rolling a k-sided die n times
- Special case here: categorical distribution
  - K>2, n=1
- Dirichlet: modeling a distribution over distributions. Conjugate prior of multinomial

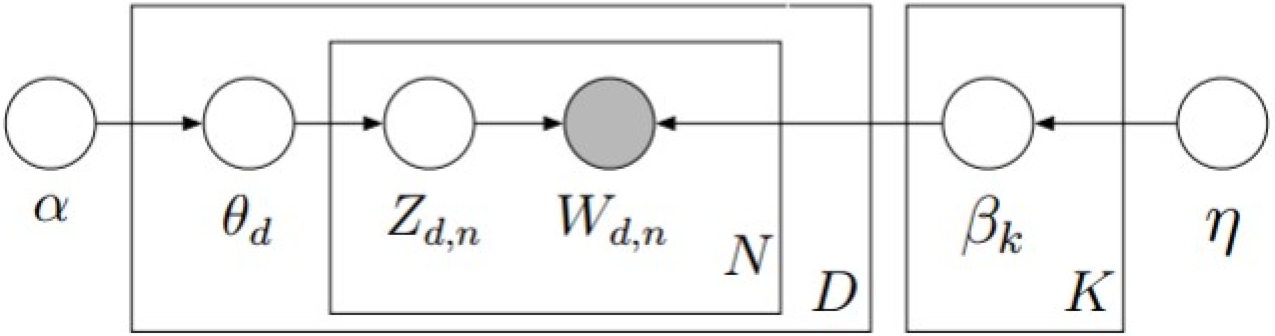
$$P(X_1 = x_1, \dots, X_K = x_K) = \frac{n!}{x_1! \ \dots \ x_K!} p_1^{x_1} \dots p_K^{x_K}$$

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{K} \alpha_i)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} \dots \theta_K^{\alpha_K - 1}$$

$$\begin{array}{rclcl} \theta \mid \boldsymbol{\alpha} & = & (\theta_1, \dots, \theta_K) & \sim & \mathrm{Dir}(K, \boldsymbol{\alpha}) \\ \mathbb{X} \mid \theta & = & (\mathbf{x}_1, \dots, \mathbf{x}_K) & \sim & \mathrm{Cat}(K, \theta) \end{array}$$

then the following holds:

$$\begin{array}{rclcl} \mathbf{c} & = & (c_1, \dots, c_K) & = & \text{number of occurrences of category } i \\ \theta \mid \mathbb{X}, \boldsymbol{\alpha} & \sim & \mathrm{Dir}(K, \mathbf{c} + \boldsymbol{\alpha}) & = & \mathrm{Dir}(K, c_1 + \alpha_1, \dots, c_K + \alpha_K) \end{array}$$



we want to infer the posterior distribution (Bayesian Inference)

Bayesian Inference

- Denote as the collection of model parameters

- Computing the integral in the denominator is impractical
- Solution: Monte Carlo simulation
  - approximate a complex problem by sampling.

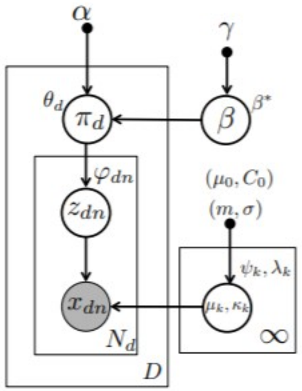
Topic modeling and word embedding

1. Draw $\mathbf{\Sigma}_c \sim \mathcal{W}^{-1}(\mathbf{\Psi}, \nu)$.
2. Draw $\boldsymbol{\mu}_c \sim \mathcal{N}(\boldsymbol{\mu}, \frac{1}{\tau_c}\mathbf{\Sigma}_c)$.
3. For each Gaussian topic $k = 1, 2, \cdots, K$:
   (a) Draw topic covariance $\mathbf{\Sigma}_k \sim \mathcal{W}^{-1}(\mathbf{\Psi}_0, \nu_0)$.
   (b) Draw topic mean $\boldsymbol{\mu}_k \sim \mathcal{N}(\boldsymbol{\mu}_0, \frac{1}{\tau}\mathbf{\Sigma}_k)$.
4. For each document $d = 1, 2, \cdots, D$:
   (a) Draw $\boldsymbol{\eta}_d \sim \mathcal{N}(\boldsymbol{\mu}_c, \mathbf{\Sigma}_c)$.
   (b) For each word index $n = 1, 2, \cdots, N_d$:
      i. Draw a topic $z_{dn} \sim Multinomial(f(\boldsymbol{\eta}_d))$.
      ii. Draw a word $\boldsymbol{w}_{dn} \sim \mathcal{N}(\boldsymbol{\mu}_{z_{dn}}, \mathbf{\Sigma}_{z_{dn}})$.

A Correlated Topic Model Using Word Embeddings, IJCAI'17

1. for $k = 1$ to K
   (a) Draw topic covariance $\mathbf{\Sigma}_k \sim \mathcal{W}^{-1}(\mathbf{\Psi}, \nu)$
   (b) Draw topic mean $\boldsymbol{\mu}_k \sim \mathcal{N}(\boldsymbol{\mu}, \frac{1}{\kappa}\mathbf{\Sigma}_k)$
2. for each document d in corpus D
   (a) Draw topic distribution $\boldsymbol{\theta}_d \sim \mathrm{Dir}(\boldsymbol{\alpha})$
   (b) for each word index n from 1 to N_d
      i. Draw a topic $z_n \sim \mathrm{Categorical}(\boldsymbol{\theta}_d)$
      ii. Draw $\mathbf{v}_{d,n} \sim \mathcal{N}(\boldsymbol{\mu}_{z_n}, \mathbf{\Sigma}_{z_n})$

Gaussian LDA for Topic Models with Word Embeddings, ACL'15



Nonparametric spherical topic modeling with word embeddings, ACL'16

LDA inference

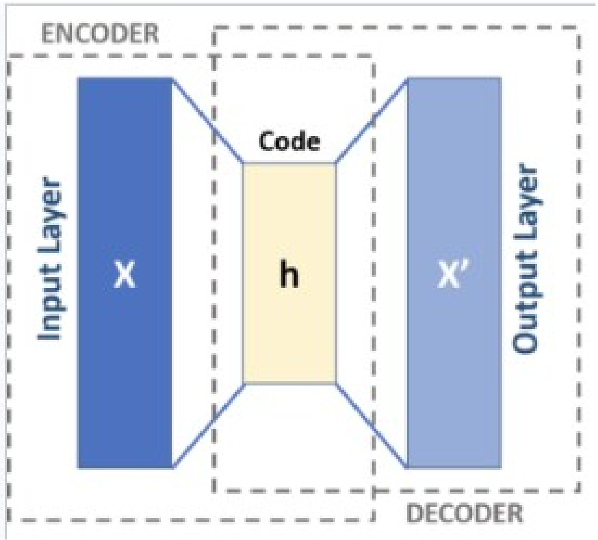
- computational cost of computing the posterior distribution
- Collapsed Gibbs sampling
  - only a small change to the modeling assumptions, requires re-deriving the inference methods
- How about train a black-box inference method?

Background

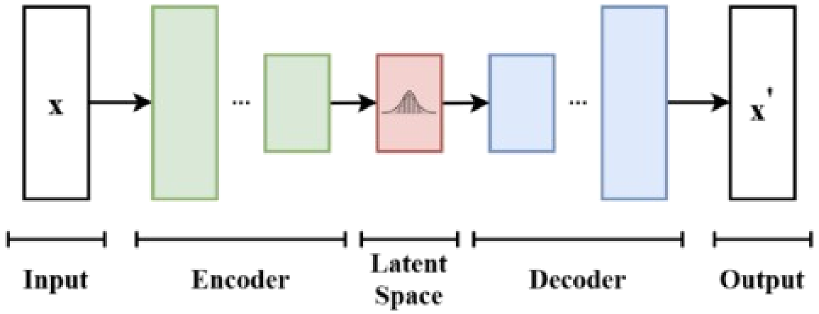
- Autoencoder

$$\min_{\theta,\phi} L(\theta,\phi), \text{where } L(\theta,\phi) = \frac{1}{N}\sum_{i=1}^{N} \|x_i - D_\theta(E_\phi(x_i))\|_2^2$$

  - Compressing the message or reducing dimensionality

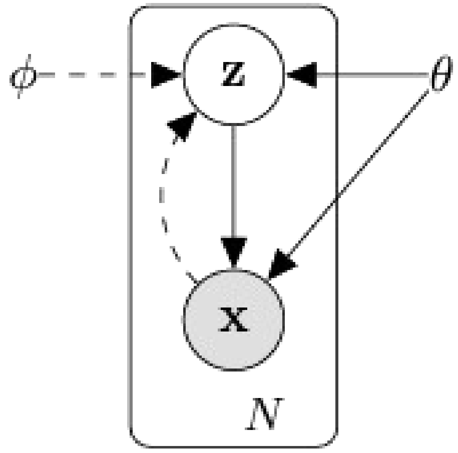


Background



- variational autoencoder (VAE)
  - Architecturally similar to autoencoder
  - significant differences in the goal and mathematical formulation
- Recall:
  - Prior:
  - Likelihood:
  - Posterior: can be intractable
- Latent representation or encoding , which is a random vector jointly-distributed with

Vanilla VAE



Dashed lines: variational approximation to the intractable

- Prior:
- Likelihood: , Gaussian
- is a mixture of Gaussian
- Posterior:
- is not easy to compute, so proximate the posterior

- probabilistic encoder: approximated posterior distribution
- probabilistic decoder: conditional likelihood distribution

Vanilla VAE -- neural net perspective

- Encoder is a neural net
  - Input x
  - Output: a Gaussian probability density
  - sample from this distribution to get noisy values of the representations z
- Decoder is another neural net
  - Input z
  - Output:
- generate new samples that resemble the original input

Optimization

- Evidence lower bound (ELBO)

Optimization

- Minimize ≋ ☾ max

- Gradient descent for backprop will have problem for because of
- Reparameterization trick or stochastic backpropagation
- , then

Back to LDA

- VAE can map a document to a well-behaved approximate posterior distribution using an inference neural network

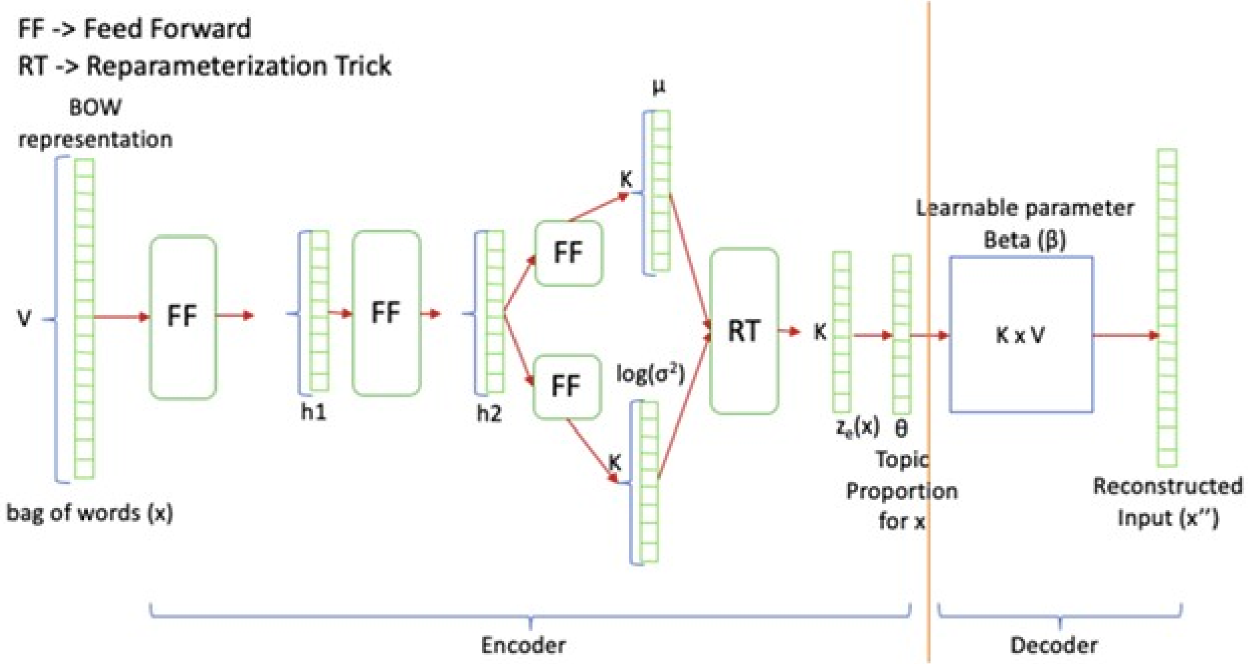


Fig. 1. ProdLDA.

Autoencoding Variational Inference For Topic Models

Two challenges to apply VAE for topic models

- the Dirichlet prior is not a location scale family, which hinders reparameterization
  - location scale family: distributions parametrized by a location parameter and a scale parameter. E.g. normal, uniform
- the encoder network becomes stuck in a bad local optimum in which all topics are identical

Solution

- Dirichlet prior is not a location scale family
- Use an encoder network that approximates the Dirichlet prior with a logistic-normal distribution

where and are the encoder network outputs

- encoder network stuck in bad local optimum
- Adam optimizer, batch normalization and dropout units in the encoder network

BOW or embedding?

- Bag of words will fail in the face of large vocabularies
  - Size of is #vocabularies * #topics
- In practice, to run LDA, severe pruning is needed
  - Remove frequent words
  - Remove very infrequent words
- Word embedding
  - Fix length (usually 100-200 dimension)

Topic Modeling in Embedding Spaces

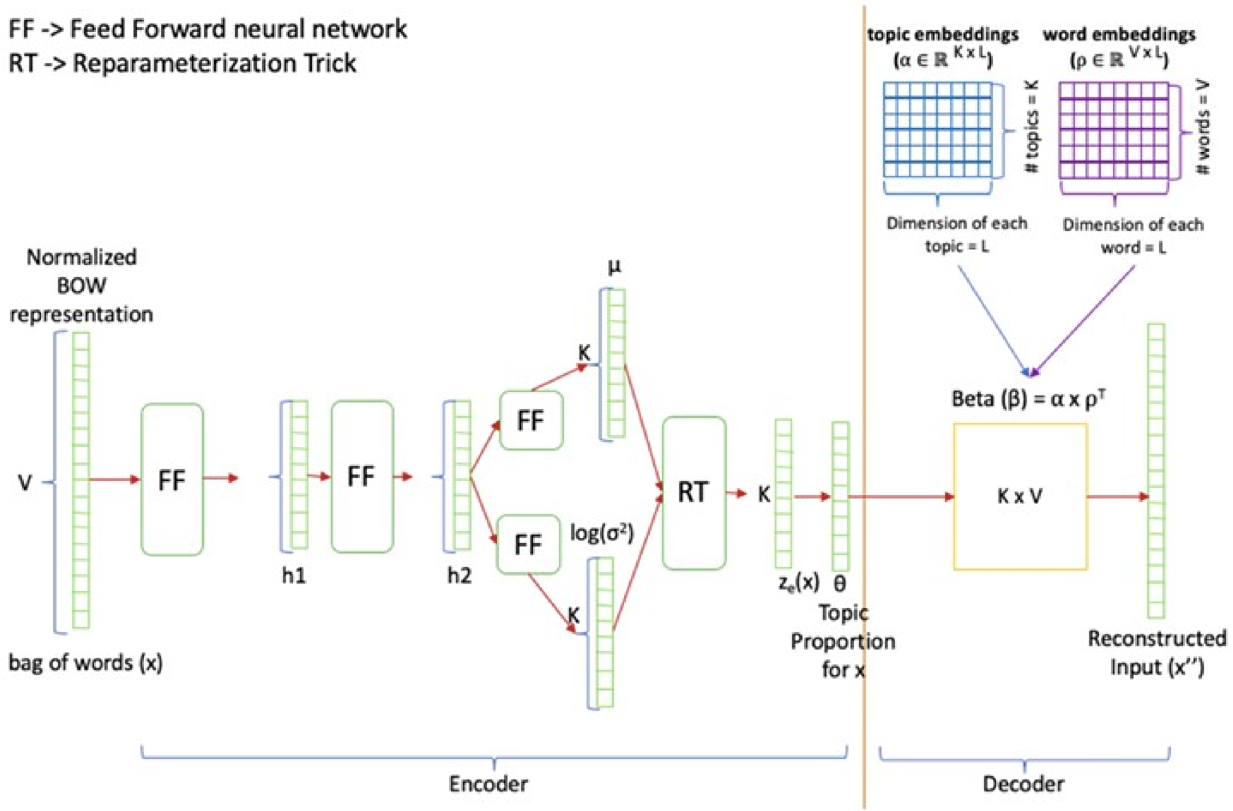


Fig. 2. Embedded Topic Model.

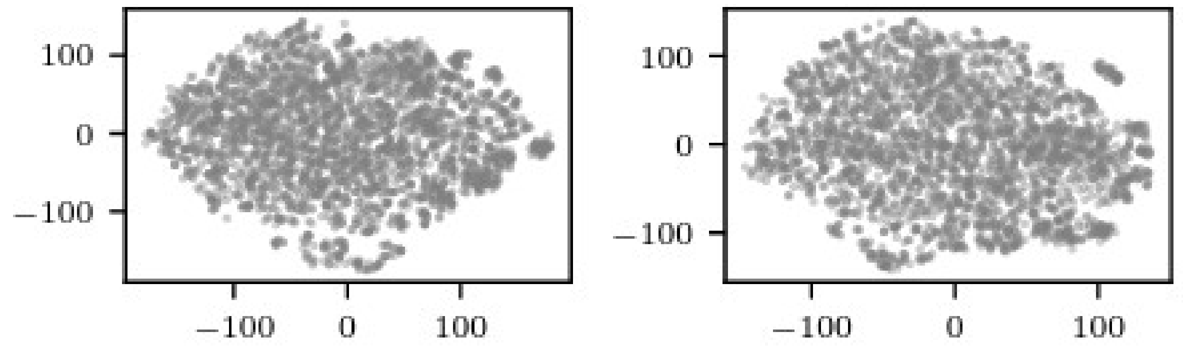
Generative process of ETM

1. Draw topic proportions $\theta_d \sim \mathcal{LN}(0, I)$.
2. For each word n in the document:
   a. Draw topic assignment $z_{dn} \sim \mathrm{Cat}(\theta_d)$.
   b. Draw the word $w_{dn} \sim \mathrm{softmax}(\rho^\top \alpha_{z_{dn}})$.

- Steps 1 and 2a are standard for topic modeling (ProdLDA)
- 2b is different: it uses the embeddings of the vocabulary and the assigned topic embedding to draw the observed word from the assigned topic
- 2b mirrors the CBOW likelihood
  - predicts the center word from (bag of) context words

Topic Discovery via Latent Space Clustering

- Topic modeling is in fact a special type of clustering
- How about we directly cluster on word embedding?



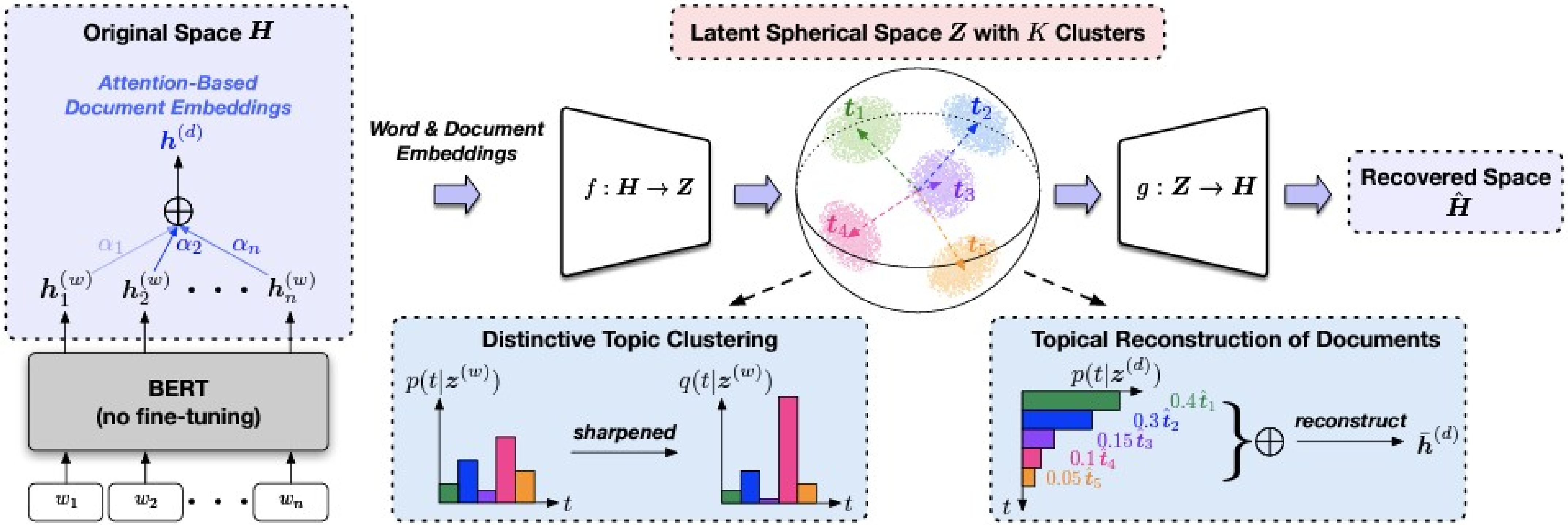
(a) **New York Times.** (b) **Yelp Review.**

Figure 1: Visualization using t-SNE of 3,000 randomly sampled contextualized word embeddings of BERT on (a) NYT and (b) Yelp datasets, respectively. The embedding spaces do not have clearly separated clusters.

Why?

- The MLM pretraining objective of BERT assumes that the learned contextualized embeddings are generated from a Gaussian Mixture Model (GMM) with $|V|$ mixture components where $|V|$ is the vocabulary size of BERT.
- PLM embeddings are usually high dimensional
- Lack of good document representations from PLMs
  - SentenceBERT reported that [CLS] token without fine-tuning is even worse than average GloVe embeddings

TopClus



Clustering loss
A topical reconstruction loss of documents
An embedding space preserving loss