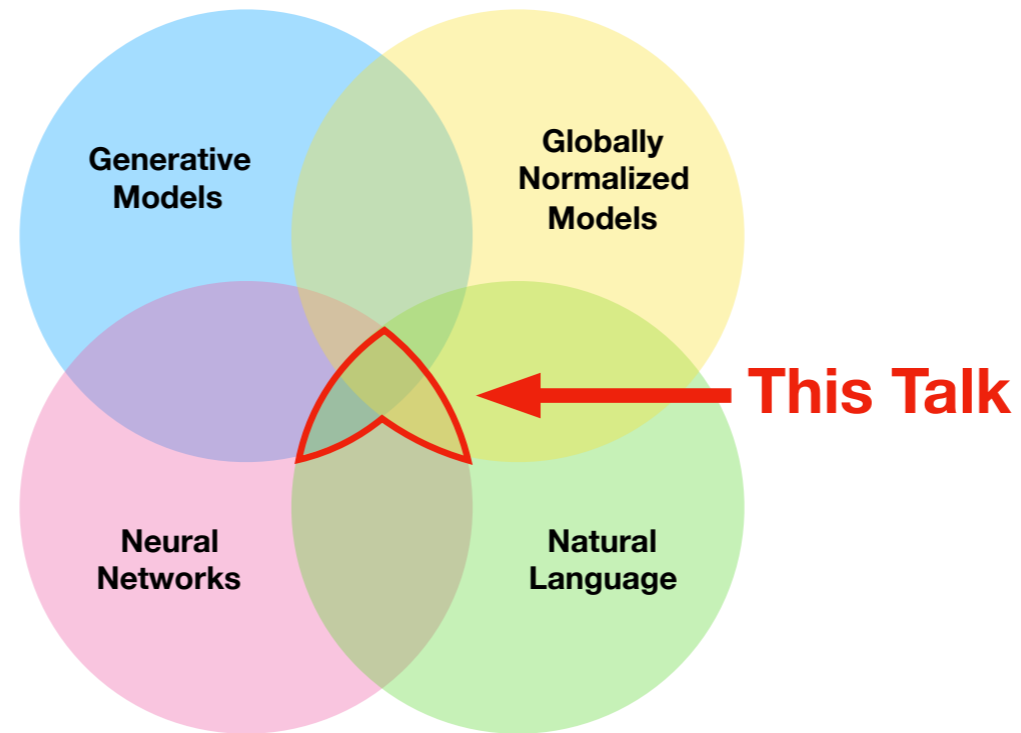


Neural Structured Probabilistic Models: Methods and Applications in NLP

Tom Effland
April 12th, 2019
Dept. of Computer Science, Columbia University

Theme & Scope



2

Concerned here with the intersection of:

Joint distributions involving text and its continuous and/or discrete correlates (e.g. representations or annotations of the text)

Using flexible, compositional neural networks to parameterize these distributions

Methods form core thread, but their wide applicability demonstrated through survey of their uses in NLP

Outline

- **VAEs**

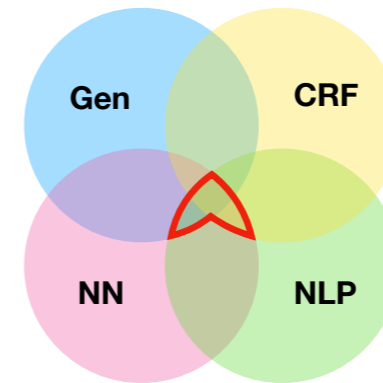
- Continuous Variables
- Optimization Issues: Posterior Collapse
- Topic Modeling
- Discrete Variables and Semi-supervised Learning

- **Neural CRFs**

- Exact Inference
- Approximate Inference

- **VAEs for Discrete Structure: Semi-Supervised Learning**

- **Viewing Attention as a Latent Variable**



Quick roadmap:

- * First we'll discuss Variational Autoencoders and their many flavors in NLP
- * Then we'll discuss neural CRFs for bridging SOTA neural architectures with structured outputs
- * Then we'll discuss intersection of these two concepts for semi-supervised learning in NLP
- * And finally we'll wrap up with a discussion of the connections of attention (now ubiquitous in NLP) to more formal characterizations of latent variables

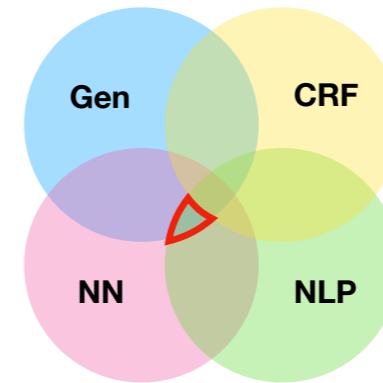
Outline

- **Variational Inference (background)**

- Continuous Variables
- Optimization Issues: Posterior Collapse
- Topic Modeling
- Discrete Variables and Semi-supervised Learning
- Neural CRFs
 - Exact Inference
 - Approximate Inference
- VAEs for Discrete Structure: Semi-Supervised Learning
- Viewing Attention as a Latent Variable

[Kingma & Welling 14]

[Rezende et al. 14]



So quickly we'll set up the variational inference problem

Background: Variational Inference

5

Unsupervised setting:

- * Observe dataset, text always BOW or Seq of Words
- * We'll have a generative model of the text with latent vars ***per datapoint***
- * We'd like to learn model by maximizing the marginal likelihood of the data and do posterior inference
- * But the marginal likelihoods are intractable integral because of some combo of NN or infinite/high dimensional latent var
- * Variational inference reframes inference and optimization of marginal likelihood using an ***approximate posterior q***
- * However, we can rewrite the marginal for any proposal q as ...
- * and we know the KL with posterior is always positive
- * so we can drop it and optimize a lower bound using an approximate posterior

Background: Variational Inference

Data $\mathcal{D} = \{x^k\}_{k=1}^D, x^k \in \{0,1\}^{|V|} \vee x^k \in V^N$

5

Unsupervised setting:

- * Observe dataset, text always BOW or Seq of Words
- * We'll have a generative model of the text with latent vars **per datapoint**
- * We'd like to learn model by maximizing the marginal likelihood of the data and do posterior inference
- * But the marginal likelihoods are intractable integral because of some combo of NN or infinite/high dimensional latent var
- * Variational inference reframes inference and optimization of marginal likelihood using an **approximate posterior q**
- * However, we can rewrite the marginal for any proposal q as ...
- * and we know the KL with posterior is always positive
- * so we can drop it and optimize a lower bound using an approximate posterior

Background: Variational Inference

Data $\mathcal{D} = \{x^k\}_{k=1}^D, x^k \in \{0,1\}^{|V|} \vee x^k \in V^N$

Model $p_\theta(x^k, z^k) = p_\theta(x^k | z^k)p(z^k)$

5

Unsupervised setting:

- * Observe dataset, text always BOW or Seq of Words
- * We'll have a generative model of the text with latent vars **per datapoint**
- * We'd like to learn model by maximizing the marginal likelihood of the data and do posterior inference
- * But the marginal likelihoods are intractable integral because of some combo of NN or infinite/high dimensional latent var
- * Variational inference reframes inference and optimization of marginal likelihood using an **approximate posterior q**
- * However, we can rewrite the marginal for any proposal q as ...
- * and we know the KL with posterior is always positive
- * so we can drop it and optimize a lower bound using an approximate posterior

Background: Variational Inference

Data $\mathcal{D} = \{x^k\}_{k=1}^D, x^k \in \{0,1\}^{|V|} \vee x^k \in V^N$

Model $p_\theta(x^k, z^k) = p_\theta(x^k | z^k)p(z^k)$

Want $\theta^* = \arg \max_{\theta} \sum_{k=1}^D \log p_\theta(x^k)$
 $p_\theta(z^k | x^k) = p_\theta(x^k | z^k)p(z^k) / p_\theta(x^k)$

5

Unsupervised setting:

- * Observe dataset, text always BOW or Seq of Words
- * We'll have a generative model of the text with latent vars **per datapoint**
- * We'd like to learn model by maximizing the marginal likelihood of the data and do posterior inference
- * But the marginal likelihoods are intractable integral because of some combo of NN or infinite/high dimensional latent var
- * Variational inference reframes inference and optimization of marginal likelihood using an **approximate posterior q**
- * However, we can rewrite the marginal for any proposal q as ...
- * and we know the KL with posterior is always positive
- * so we can drop it and optimize a lower bound using an approximate posterior

Background: Variational Inference

Data $\mathcal{D} = \{x^k\}_{k=1}^D, x^k \in \{0,1\}^{|\mathcal{V}|} \vee x^k \in V^N$

Model $p_\theta(x^k, z^k) = p_\theta(x^k | z^k)p(z^k)$

Want $\theta^* = \arg \max_{\theta} \sum_{k=1}^D \log p_\theta(x^k)$ ← Intractable
 $p_\theta(z^k | x^k) = p_\theta(x^k | z^k)p(z^k) / p_\theta(x^k)$

5

Unsupervised setting:

- * Observe dataset, text always BOW or Seq of Words
- * We'll have a generative model of the text with latent vars **per datapoint**
- * We'd like to learn model by maximizing the marginal likelihood of the data and do posterior inference
- * But the marginal likelihoods are intractable integral because of some combo of NN or infinite/high dimensional latent var
- * Variational inference reframes inference and optimization of marginal likelihood using an **approximate posterior q**
- * However, we can rewrite the marginal for any proposal q as ...
- * and we know the KL with posterior is always positive
- * so we can drop it and optimize a lower bound using an approximate posterior

Background: Variational Inference

Data $\mathcal{D} = \{x^k\}_{k=1}^D, x^k \in \{0,1\}^{|V|} \vee x^k \in V^N$

Model $p_\theta(x^k, z^k) = p_\theta(x^k | z^k)p(z^k)$

Want $\theta^* = \arg \max_{\theta} \sum_{k=1}^D \log p_\theta(x^k)$ ← Intractable
 $p_\theta(z^k | x^k) = p_\theta(x^k | z^k)p(z^k) / p_\theta(x^k)$

ELBO: Inference as Optimization

$$\log p_\theta(x^k) = \mathbb{E}_{q_\phi(z^k)}[\log p_\theta(x^k | z^k)] - KL(q_\phi(z^k) || p(z^k)) + KL(q_\phi(z^k) || p_\theta(z^k | x^k))$$

5

Unsupervised setting:

- * Observe dataset, text always BOW or Seq of Words
- * We'll have a generative model of the text with latent vars **per datapoint**
- * We'd like to learn model by maximizing the marginal likelihood of the data and do posterior inference
- * But the marginal likelihoods are intractable integral because of some combo of NN or infinite/high dimensional latent var
- * Variational inference reframes inference and optimization of marginal likelihood using an **approximate posterior q**
- * However, we can rewrite the marginal for any proposal q as ...
- * and we know the KL with posterior is always positive
- * so we can drop it and optimize a lower bound using an approximate posterior

Background: Variational Inference

Data $\mathcal{D} = \{x^k\}_{k=1}^D, x^k \in \{0,1\}^{|\mathcal{V}|} \vee x^k \in V^N$

Model $p_\theta(x^k, z^k) = p_\theta(x^k | z^k)p(z^k)$

Want $\theta^* = \arg \max_{\theta} \sum_{k=1}^D \log p_\theta(x^k)$ ← Intractable
 $p_\theta(z^k | x^k) = p_\theta(x^k | z^k)p(z^k) / p_\theta(x^k)$

ELBO: Inference as Optimization

$$\log p_\theta(x^k) = \mathbb{E}_{q_\phi(z^k)}[\log p_\theta(x^k | z^k)] - \underbrace{KL(q_\phi(z^k) || p(z^k)) + KL(q_\phi(z^k) || p_\theta(z^k | x^k))}_{\geq 0}$$

5

Unsupervised setting:

- * Observe dataset, text always BOW or Seq of Words
- * We'll have a generative model of the text with latent vars **per datapoint**
- * We'd like to learn model by maximizing the marginal likelihood of the data and do posterior inference
- * But the marginal likelihoods are intractable integral because of some combo of NN or infinite/high dimensional latent var
- * Variational inference reframes inference and optimization of marginal likelihood using an **approximate posterior q**
- * However, we can rewrite the marginal for any proposal q as ...
- * and we know the KL with posterior is always positive
- * so we can drop it and optimize a lower bound using an approximate posterior

Background: Variational Inference

Data $\mathcal{D} = \{x^k\}_{k=1}^D, x^k \in \{0,1\}^{|\mathcal{V}|} \vee x^k \in V^N$

Model $p_\theta(x^k, z^k) = p_\theta(x^k | z^k)p(z^k)$

Want $\theta^* = \arg \max_{\theta} \sum_{k=1}^D \log p_\theta(x^k)$ ← Intractable
 $p_\theta(z^k | x^k) = p_\theta(x^k | z^k)p(z^k) / p_\theta(x^k)$

ELBO: Inference as Optimization

$$\begin{aligned} \log p_\theta(x^k) &= \mathbb{E}_{q_\phi(z^k)}[\log p_\theta(x^k | z^k)] - \underbrace{KL(q_\phi(z^k) || p(z^k)) + KL(q_\phi(z^k) || p_\theta(z^k | x^k))}_{\geq 0} \\ &\geq \mathbb{E}_{q_\phi(z^k)}[\log p_\theta(x^k | z^k)] - KL(q_\phi(z^k) || p(z^k)) \\ &= L(\theta, \phi; x^k) \end{aligned}$$

5

Unsupervised setting:

- * Observe dataset, text always BOW or Seq of Words
- * We'll have a generative model of the text with latent vars **per datapoint**
- * We'd like to learn model by maximizing the marginal likelihood of the data and do posterior inference
- * But the marginal likelihoods are intractable integral because of some combo of NN or infinite/high dimensional latent var
- * Variational inference reframes inference and optimization of marginal likelihood using an **approximate posterior q**
- * However, we can rewrite the marginal for any proposal q as ...
- * and we know the KL with posterior is always positive
- * so we can drop it and optimize a lower bound using an approximate posterior

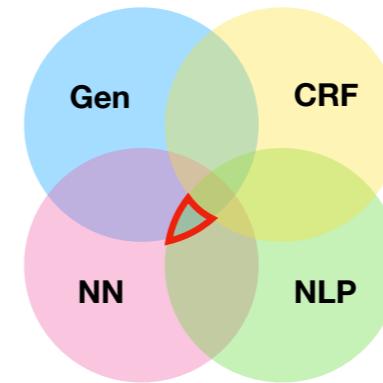
Outline

- **Variational Autoencoder (background)**

[Kingma & Welling 14]

[Rezende et al. 14]

- Continuous Variables
- Optimization Issues: Posterior Collapse
- Topic Modeling
- Discrete Variables and Semi-supervised Learning
- Neural CRFs
 - Exact Inference
 - Approximate Inference
- VAEs for Discrete Structure: Semi-Supervised Learning
- Viewing Attention as a Latent Variable



6

But first, we'll outline VAEs because they're involved in $> 2/3$ of the papers

What we're going to see is they solve two problems:

- * They show how to get low-variance unbiased gradients wrt ϕ
- * They show how to use a "recognition network" for approximate posterior q to allow for fast inference that generalizes to new datapoints

Variational Autoencoders: Contribution (1)

$$L(\theta, \phi; x) = \mathbb{E}_{q_\phi(z)}[\log p_\theta(x|z)] - KL(q_\phi(z) || p(z))$$
$$\approx \frac{1}{M} \sum_{j=1}^M [\log p_\theta(x|z^{(j)})] - KL(q_\phi(z) || p(z))$$

[Kingma & Welling 14]

[Rezende et al. 14]

7

Which brings us to our first two papers, published contemporaneously:

- * Here's our loss function again, which we need to further approximate with monte-carlo because p_θ is expensive to evaluate (a DNN)
- * Note: KL is typically analytically tractable
- * How to optimize? gradient descent would be great (simple, autodiff very powerful framework)
- * We can easily get gradient wrt theta, but have problems with phi...
- * Which brings us to first contribution: for a normally distributed z , we can reparameterize as deterministic function of standard noise
- * So we can rewrite the expectation and now we can get low-variance unbiased gradients wrt phi

Variational Autoencoders: Contribution (1)

$$L(\theta, \phi; x) = \mathbb{E}_{q_\phi(z)}[\log p_\theta(x|z)] - KL(q_\phi(z) || p(z))$$
$$\approx \frac{1}{M} \sum_{j=1}^M [\log p_\theta(x|z^{(j)})] - KL(q_\phi(z) || p(z))$$

How To Optimize? $\nabla_\theta L(\theta, \phi; x)$ ✓

[Kingma & Welling 14]

[Rezende et al. 14]

7

Which brings us to our first two papers, published contemporaneously:

- * Here's our loss function again, which we need to further approximate with monte-carlo because p_θ is expensive to evaluate (a DNN)
- * Note: KL is typically analytically tractable
- * How to optimize? gradient descent would be great (simple, autodiff very powerful framework)
- * We can easily get gradient wrt theta, but have problems with phi...
- * Which brings us to first contribution: for a normally distributed z , we can reparameterize as deterministic function of standard noise
- * So we can rewrite the expectation and now we can get low-variance unbiased gradients wrt phi

Variational Autoencoders: Contribution (1)

$$L(\theta, \phi; x) = \mathbb{E}_{q_\phi(z)}[\log p_\theta(x|z)] - KL(q_\phi(z) || p(z))$$
$$\approx \frac{1}{M} \sum_{j=1}^M [\log p_\theta(x|z^{(j)})] - KL(q_\phi(z) || p(z))$$

How To Optimize? $\nabla_\theta L(\theta, \phi; x)$ ✓ $\nabla_\phi L(\theta, \phi; x)$

[Kingma & Welling 14]

[Rezende et al. 14]

7

Which brings us to our first two papers, published contemporaneously:

- * Here's our loss function again, which we need to further approximate with monte-carlo because p_θ is expensive to evaluate (a DNN)
- * Note: KL is typically analytically tractable
- * How to optimize? gradient descent would be great (simple, autodiff very powerful framework)
- * We can easily get gradient wrt theta, but have problems with phi...
- * Which brings us to first contribution: for a normally distributed z , we can reparameterize as deterministic function of standard noise
- * So we can rewrite the expectation and now we can get low-variance unbiased gradients wrt phi

Variational Autoencoders: Contribution (1)

$$L(\theta, \phi; x) = \mathbb{E}_{q_\phi(z)}[\log p_\theta(x|z)] - KL(q_\phi(z) || p(z))$$
$$\approx \frac{1}{M} \sum_{j=1}^M [\log p_\theta(x|z^{(j)})] - KL(q_\phi(z) || p(z))$$

How To Optimize? $\nabla_\theta L(\theta, \phi; x)$ ✓ $\nabla_\phi L(\theta, \phi; x)$ ✗

[Kingma & Welling 14]

[Rezende et al. 14]

7

Which brings us to our first two papers, published contemporaneously:

- * Here's our loss function again, which we need to further approximate with monte-carlo because p_θ is expensive to evaluate (a DNN)
- * Note: KL is typically analytically tractable
- * How to optimize? gradient descent would be great (simple, autodiff very powerful framework)
- * We can easily get gradient wrt theta, but have problems with phi...
- * Which brings us to first contribution: for a normally distributed z , we can reparameterize as deterministic function of standard noise
- * So we can rewrite the expectation and now we can get low-variance unbiased gradients wrt phi

Variational Autoencoders: Contribution (1)

$$L(\theta, \phi; x) = \mathbb{E}_{q_\phi(z)}[\log p_\theta(x|z)] - KL(q_\phi(z) || p(z))$$
$$\approx \frac{1}{M} \sum_{j=1}^M [\log p_\theta(x|z^{(j)})] - KL(q_\phi(z) || p(z))$$

How To Optimize? $\nabla_\theta L(\theta, \phi; x)$ ✓ $\nabla_\phi L(\theta, \phi; x)$ ✗

Reparameterization Trick! (not always applicable)

[Kingma & Welling 14]

[Rezende et al. 14]

7

Which brings us to our first two papers, published contemporaneously:

- * Here's our loss function again, which we need to further approximate with monte-carlo because p_θ is expensive to evaluate (a DNN)
- * Note: KL is typically analytically tractable
- * How to optimize? gradient descent would be great (simple, autodiff very powerful framework)
- * We can easily get gradient wrt theta, but have problems with phi...
- * Which brings us to first contribution: for a normally distributed z , we can reparameterize as deterministic function of standard noise
- * So we can rewrite the expectation and now we can get low-variance unbiased gradients wrt phi

Variational Autoencoders: Contribution (1)

$$L(\theta, \phi; x) = \mathbb{E}_{q_\phi(z)}[\log p_\theta(x|z)] - KL(q_\phi(z) || p(z))$$
$$\approx \frac{1}{M} \sum_{j=1}^M [\log p_\theta(x|z^{(j)})] - KL(q_\phi(z) || p(z))$$

How To Optimize? $\nabla_\theta L(\theta, \phi; x)$ ✓ $\nabla_\phi L(\theta, \phi; x)$ ✗

Reparameterization Trick! (not always applicable)

$$z \sim \mathcal{N}(\mu, \sigma^2) \Rightarrow z = \mu + \sigma \epsilon, \epsilon \sim \mathcal{N}(0,1)$$

[Kingma & Welling 14]

[Rezende et al. 14]

7

Which brings us to our first two papers, published contemporaneously:

- * Here's our loss function again, which we need to further approximate with monte-carlo because p_θ is expensive to evaluate (a DNN)
- * Note: KL is typically analytically tractable
- * How to optimize? gradient descent would be great (simple, autodiff very powerful framework)
- * We can easily get gradient wrt theta, but have problems with phi...
- * Which brings us to first contribution: for a normally distributed z , we can reparameterize as deterministic function of standard noise
- * So we can rewrite the expectation and now we can get low-variance unbiased gradients wrt phi

Variational Autoencoders: Contribution (1)

$$L(\theta, \phi; x) = \mathbb{E}_{q_\phi(z)}[\log p_\theta(x|z)] - KL(q_\phi(z) || p(z))$$
$$\approx \frac{1}{M} \sum_{j=1}^M [\log p_\theta(x|z^{(j)})] - KL(q_\phi(z) || p(z))$$

How To Optimize? $\nabla_\theta L(\theta, \phi; x)$ ✓ $\nabla_\phi L(\theta, \phi; x)$ ✗

Reparameterization Trick! (not always applicable)

$$z \sim \mathcal{N}(\mu, \sigma^2) \Rightarrow z = \mu + \sigma \epsilon, \epsilon \sim \mathcal{N}(0,1)$$

$$\mathbb{E}_{q_\phi(z)}[\log p_\theta(x|z)] = \mathbb{E}_{q(\epsilon)}[\log p_\theta(x|z_\phi(\epsilon))] \approx \frac{1}{M} \sum_{j=1}^M [\log p_\theta(x|z_\phi(\epsilon^{(j)}))]$$

[Kingma & Welling 14]

[Rezende et al. 14]

7

Which brings us to our first two papers, published contemporaneously:

- * Here's our loss function again, which we need to further approximate with monte-carlo because p_θ is expensive to evaluate (a DNN)
- * Note: KL is typically analytically tractable
- * How to optimize? gradient descent would be great (simple, autodiff very powerful framework)
- * We can easily get gradient wrt theta, but have problems with phi...
- * Which brings us to first contribution: for a normally distributed z , we can reparameterize as deterministic function of standard noise
- * So we can rewrite the expectation and now we can get low-variance unbiased gradients wrt phi

Variational Autoencoders: Contribution (1)

$$L(\theta, \phi; x) = \mathbb{E}_{q_\phi(z)}[\log p_\theta(x|z)] - KL(q_\phi(z) || p(z))$$
$$\approx \frac{1}{M} \sum_{j=1}^M [\log p_\theta(x|z^{(j)})] - KL(q_\phi(z) || p(z))$$

How To Optimize? $\nabla_\theta L(\theta, \phi; x)$ ✓ $\nabla_\phi L(\theta, \phi; x)$ ✓

Reparameterization Trick! (not always applicable)

$$z \sim \mathcal{N}(\mu, \sigma^2) \Rightarrow z = \mu + \sigma \epsilon, \epsilon \sim \mathcal{N}(0,1)$$

$$\mathbb{E}_{q_\phi(z)}[\log p_\theta(x|z)] = \mathbb{E}_{q(\epsilon)}[\log p_\theta(x|z_\phi(\epsilon))] \approx \frac{1}{M} \sum_{j=1}^M [\log p_\theta(x|z_\phi(\epsilon^{(j)}))]$$

[Kingma & Welling 14]

[Rezende et al. 14]

7

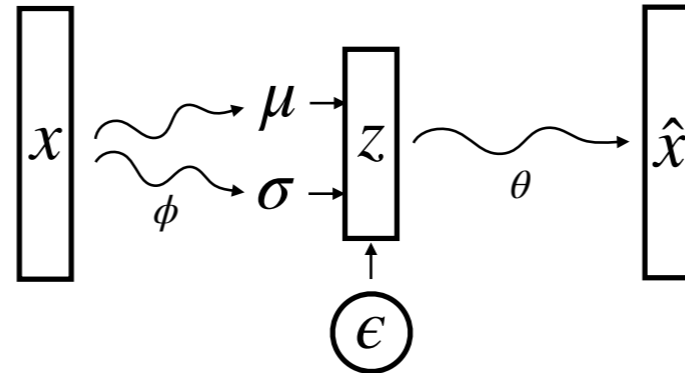
Which brings us to our first two papers, published contemporaneously:

- * Here's our loss function again, which we need to further approximate with monte-carlo because p_θ is expensive to evaluate (a DNN)
- * Note: KL is typically analytically tractable
- * How to optimize? gradient descent would be great (simple, autodiff very powerful framework)
- * We can easily get gradient wrt theta, but have problems with phi...
- * Which brings us to first contribution: for a normally distributed z , we can reparameterize as deterministic function of standard noise
- * So we can rewrite the expectation and now we can get low-variance unbiased gradients wrt phi

Variational Autoencoders: Contribution (2)

Amortized Inference = Inference Network

$$z_{\phi_x}(\epsilon) \rightarrow z_{\phi}(\epsilon, x) = \mu_{\phi}(x) + \sigma_{\phi}(x)\epsilon$$



[Kingma & Welling 14]

[Rezende et al. 14]

8

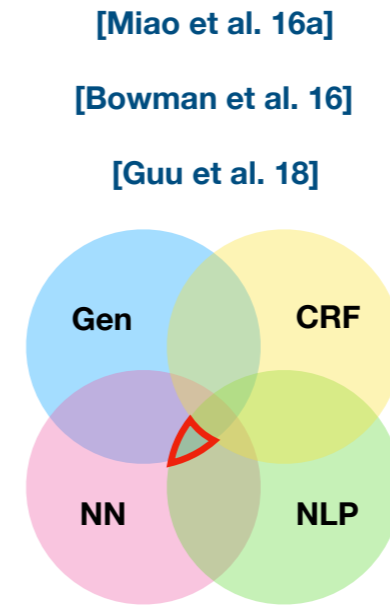
- * Now that we can get gradients wrt phi, why not also predict these params themselves with a DNN,
- * allowing for generalization across instances and fast inference at test time
- * This is the variational autoencoder

Again, they solve two problems:

- * They show how to get low-variance unbiased gradients wrt phi
- * They show how to use a “recognition network” for approximate posterior q to allow for fast inference that generalizes to new datapoints

Outline

- VAEs
 - **Continuous Variables**
 - Optimization Issues: Posterior Collapse
 - Topic Modeling
 - Discrete Variables and Semi-supervised Learning
 - Neural CRFs
 - Exact Inference
 - Approximate Inference
 - VAEs for Discrete Structure: Semi-Supervised Learning
 - Viewing Attention as a Latent Variable

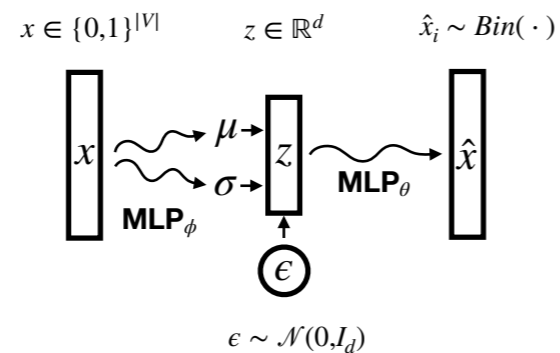


With VAEs in hand, now we'll look at their first applications to NLP

All of these papers use autoencoders with stochastic latent representations for document/sentence modeling, but each model the problem a bit differently

Generating Text from Continuous Latent Space with VAEs

[Miao et al. 16a]



10

First Miao used VAEs for learning dense representations of documents in a bag-of-words model

- * It gets better likelihoods than prior models, such as LDA
- * But since its BOW, can't actually generate any text, really just for inference

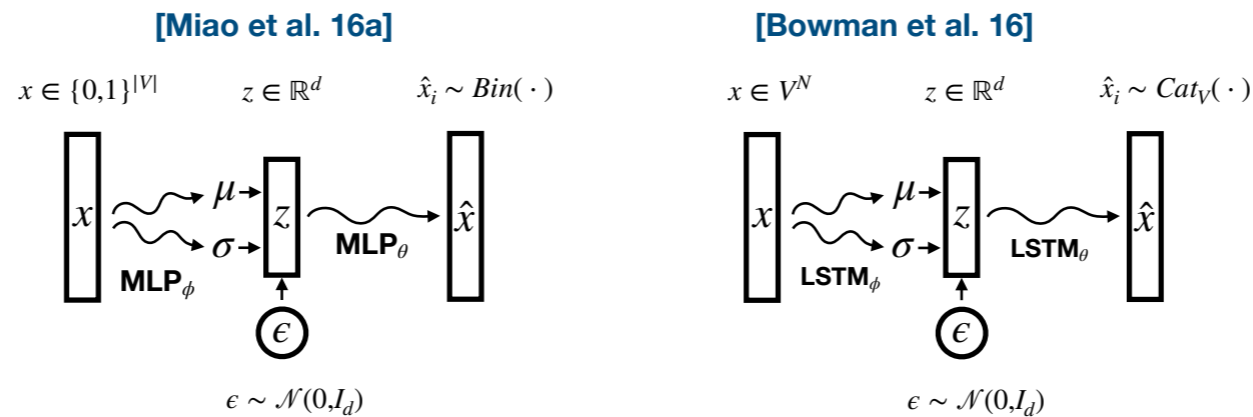
Then Bowman published a VAE for sequence reps of documents using a seq2seq model with a latent gaussian

- * While they were first to do VAE for text generation
- * The results are largely negative — the model is not able to outperform a standard RNN lang model
- * Because of serious optimization issues called posterior collapse, which we'll get into next
- * Their solution is to anneal the KL term, which is necessary to get the model to use z at all

Finally we have Guu, who also use an autoencoder, but there method is semi-parametric

- * Instead of generating a sentence from scratch, they sample a "prototype" sentence from the training data and an "edit" vector, then use a seq2seq model to make simple changes to the prototype for generation
- * This yields good results
- * and interestingly, the approximate posterior is essentially fixed which drives learning of the generative model posterior — basically the reverse of how we'd normally think about it
- * Also, they don't use a normal distribution, they use a product of magnitude (uniform) and direction (vonMises) distributions to get around posterior collapse, which we'll discuss in the next section

Generating Text from Continuous Latent Space with VAEs



10

First Miao used VAEs for learning dense representations of documents in a bag-of-words model

- * It gets better likelihoods than prior models, such as LDA
- * But since its BOW, can't actually generate any text, really just for inference

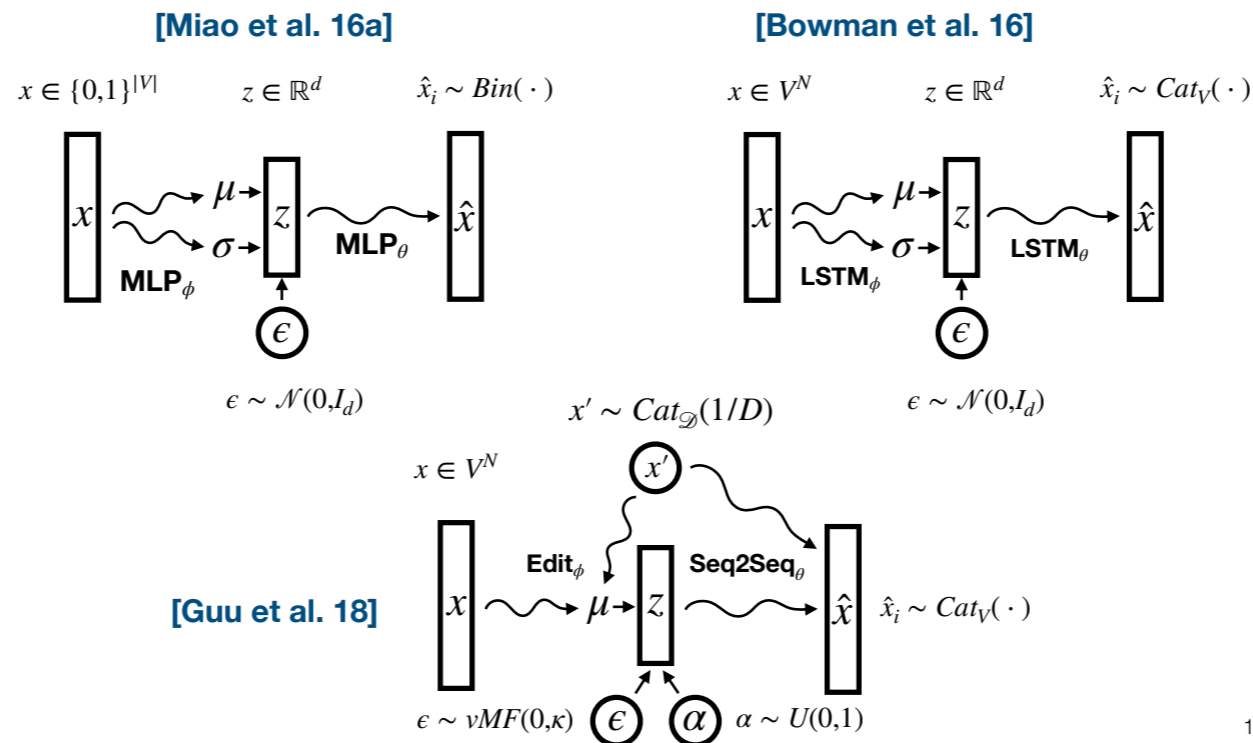
Then Bowman published a VAE for sequence reps of documents using a seq2seq model with a latent gaussian

- * While they were first to do VAE for text generation
- * The results are largely negative — the model is not able to outperform a standard RNN lang model
- * Because of serious optimization issues called posterior collapse, which we'll get into next
- * Their solution is to anneal the KL term, which is necessary to get the model to use z at all

Finally we have Guu, who also use an autoencoder, but there method is semi-parametric

- * Instead of generating a sentence from scratch, they sample a "prototype" sentence from the training data and an "edit" vector, then use a seq2seq model to make simple changes to the prototype for generation
- * This yields good results
- * and interestingly, the approximate posterior is essentially fixed which drives learning of the generative model posterior — basically the reverse of how we'd normally think about it
- * Also, they don't use a normal distribution, they use a product of magnitude (uniform) and direction (vonMises) distributions to get around posterior collapse, which we'll discuss in the next section

Generating Text from Continuous Latent Space with VAEs



10

First Miao used VAEs for learning dense representations of documents in a bag-of-words model

- * It gets better likelihoods than prior models, such as LDA
- * But since its BOW, can't actually generate any text, really just for inference

Then Bowman published a VAE for sequence reps of documents using a seq2seq model with a latent gaussian

- * While they were first to do VAE for text generation
- * The results are largely negative — the model is not able to outperform a standard RNN lang model
- * Because of serious optimization issues called posterior collapse, which we'll get into next
- * Their solution is to anneal the KL term, which is necessary to get the model to use z at all

Finally we have Guu, who also use an autoencoder, but there method is semi-parametric

- * Instead of generating a sentence from scratch, they sample a "prototype" sentence from the training data and an "edit" vector, then use a seq2seq model to make simple changes to the prototype for generation
- * This yields good results
- * and interestingly, the approximate posterior is essentially fixed which drives learning of the generative model posterior — basically the reverse of how we'd normally think about it
- * Also, they don't use a normal distribution, they use a product of magnitude (uniform) and direction (vonMises) distributions to get around posterior collapse, which we'll discuss in the next section

Generating Text from Continuous Latent Space with VAEs

Paper, Task	Contributions	Limitations
[Miao et al. 16a] bag-of-words representation	First document VAE, better likelihood than LDA	No word order information: cannot generate grammatical text
[Bowman et al. 16] sentence generation	First text VAE with word order in generation	Posterior collapse: underfits
[Guu et al. 18] sentence generation	1. Editing prototypes easier than from scratch 2. Fixed inference shapes generative model	Poor generation far from training data because of simple edits from non- parametric samples

11

They represent the docs a bit differently, but really just different flavors of this idea.

Outline

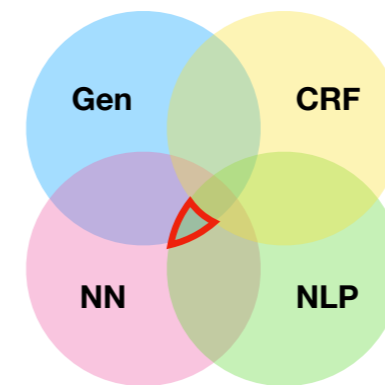
- VAEs
 - Continuous Variables
 - **Optimization Issues: Posterior Collapse**
 - Topic Modeling
 - Discrete Variables and Semi-supervised Learning
- Neural CRFs
 - Exact Inference
 - Approximate Inference
- VAEs for Discrete Structure: Semi-Supervised Learning
- Viewing Attention as a Latent Variable

[Yang et al. 17]

[Xu & Durrett 18]

[Kim et al. 18]

[He et al. 19]



Next we'll dive into the super-prominent optimization issue in VAEs: posterior collapse and discuss approaches to mitigating its deleterious effects

Posterior Collapse

**Autoregressive
Decoder** $p_{\theta}(x_1, \dots, x_N | z) = \prod_{i=1}^N p_{\theta}(x_i | x_{<i}, z)$

13

Most models which try to model actual word order, such as Bowman and Guu, will use an autoregressive RNN Language Model, which factorizes as...

This yields, for a single sample of z , the ELBO ...

Posterior collapse is the phenomena where the model gets trapped a bad local optimum early in training

Where the KL goes to 0 and z contains no information about x

And the generative model converges to an autoregressive that ignores this z

Posterior Collapse

$$\text{Autoregressive Decoder } p_{\theta}(x_1, \dots, x_N | z) = \prod_{i=1}^N p_{\theta}(x_i | x_{<i}, z)$$

$$L(\theta, \phi; x) \approx \sum_{i=1}^N [\log p_{\theta}(x_i | x_{<i}, z)] - KL(q_{\phi}(z) || p(z))$$

13

Most models which try to model actual word order, such as Bowman and Guu, will use an autoregressive RNN Language Model, which factorizes as...

This yields, for a single sample of z , the ELBO ...

Posterior collapse is the phenomena where the model gets trapped a bad local optimum early in training

Where the KL goes to 0 and z contains no information about x

And the generative model converges to an autoregressive that ignores this z

Posterior Collapse

$$\text{Autoregressive Decoder } p_{\theta}(x_1, \dots, x_N | z) = \prod_{i=1}^N p_{\theta}(x_i | x_{<i}, z)$$

**Bad local optima
caused by poor q initialization & powerful decoder**

$$L(\theta, \phi; x) \approx \sum_{i=1}^N [\log p_{\theta}(x_i | x_{<i}, z)] - KL(q_{\phi}(z) || p(z))$$

13

Most models which try to model actual word order, such as Bowman and Guu, will use an autoregressive RNN Language Model, which factorizes as...

This yields, for a single sample of z , the ELBO ...

Posterior collapse is the phenomena where the model gets trapped a bad local optimum early in training

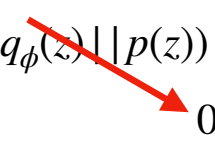
Where the KL goes to 0 and z contains no information about x

And the generative model converges to an autoregressive that ignores this z

Posterior Collapse

Autoregressive Decoder $p_{\theta}(x_1, \dots, x_N | z) = \prod_{i=1}^N p_{\theta}(x_i | x_{<i}, z)$

**Bad local optima
caused by poor q initialization & powerful decoder**

$$L(\theta, \phi; x) \approx \sum_{i=1}^N [\log p_{\theta}(x_i | x_{<i}, z)] - KL(q_{\phi}(z) || p(z))$$


13

Most models which try to model actual word order, such as Bowman and Guu, will use an autoregressive RNN Language Model, which factorizes as...

This yields, for a single sample of z , the ELBO ...

Posterior collapse is the phenomena where the model gets trapped a bad local optimum early in training

Where the KL goes to 0 and z contains no information about x

And the generative model converges to an autoregressive that ignores this z

Posterior Collapse

Autoregressive Decoder $p_{\theta}(x_1, \dots, x_N | z) = \prod_{i=1}^N p_{\theta}(x_i | x_{<i}, z)$

**Bad local optima
caused by poor q initialization & powerful decoder**

$$L(\theta, \phi; x) \approx \sum_{i=1}^N [\log p_{\theta}(x_i | x_{<i}, z)] - KL(q_{\phi}(z) || p(z))$$

$\log p_{\theta}(x_i | x_{<i})$

**Autoregressive language model
(ignores global information)**

Most models which try to model actual word order, such as Bowman and Guu, will use an autoregressive RNN Language Model, which factorizes as...

This yields, for a single sample of z , the ELBO ...

Posterior collapse is the phenomena where the model gets trapped a bad local optimum early in training

Where the KL goes to 0 and z contains no information about x

And the generative model converges to an autoregressive that ignores this z

Mitigating Posterior Collapse

14

One of the first papers to successfully deal with this problem is Yang et al.

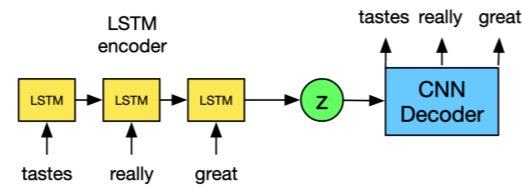
- * They propose to solve it by limiting the dependency structure generative model using dilated convolutions, preventing the model from seeing too much history
- * They find that while this helps and are first to get LL above RNNLM, but there is a clear tradeoff in decoder capacity and use of z

Xu and Durrett take a different tack:

- * They address the problem by switching the latent var distribution to a Fisher vonMises dist, which puts mass on the unit hyper-sphere
- * By doing this, the KL term no longer depends on μ and effectively becomes a tunable hyperparameter of the model, allowing for tuning the balance of contribution between reconstruction and KL to LL of sentences
- * This works quite well in practice, but now we've introduced another hyperparameter, κ , that needs tuning

Mitigating Posterior Collapse

[Yang et al. 17]



14

One of the first papers to successfully deal with this problem is Yang et al.

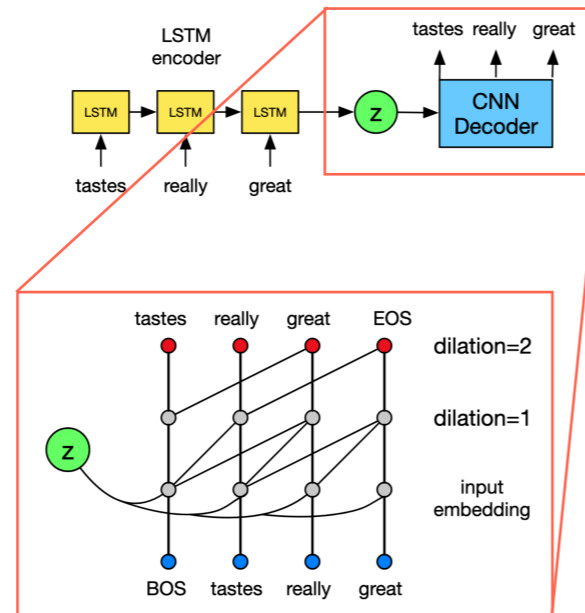
- * They propose to solve it by limiting the dependency structure generative model using dilated convolutions, preventing the model from seeing too much history
- * They find that while this helps and are first to get LL above RNNLM, but there is a clear tradeoff in decoder capacity and use of z

Xu and Durrett take a different tack:

- * They address the problem by switching the latent var distribution to a Fisher vonMises dist, which puts mass on the unit hyper-sphere
- * By doing this, the KL term no longer depends on mu and effectively becomes a tunable hyperparameter of the model, allowing for tuning the balance of contribution between reconstruction and KL to LL of sentences
- * This works quite well in practice, but now we've introduced another hyperparameter, kappa, that needs tuning

Mitigating Posterior Collapse

[Yang et al. 17]



14

One of the first papers to successfully deal with this problem is Yang et al.

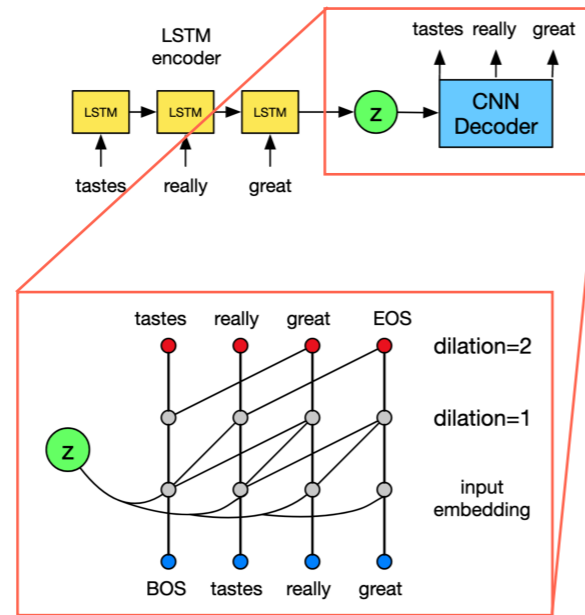
- * They propose to solve it by limiting the dependency structure generative model using dilated convolutions, preventing the model from seeing too much history
- * They find that while this helps and are first to get LL above RNNLM, but there is a clear tradeoff in decoder capacity and use of z

Xu and Durrett take a different tack:

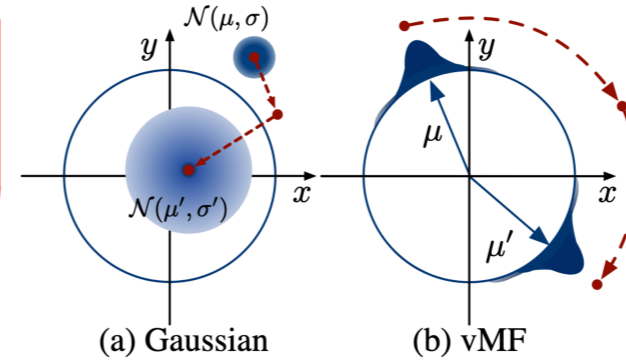
- * They address the problem by switching the latent var distribution to a Fisher vonMises dist, which puts mass on the unit hyper-sphere
- * By doing this, the KL term no longer depends on mu and effectively becomes a tunable hyperparameter of the model, allowing for tuning the balance of contribution between reconstruction and KL to LL of sentences
- * This works quite well in practice, but now we've introduced another hyperparameter, kappa, that needs tuning

Mitigating Posterior Collapse

[Yang et al. 17]



[Xu & Durrett 18]



14

One of the first papers to successfully deal with this problem is Yang et al.

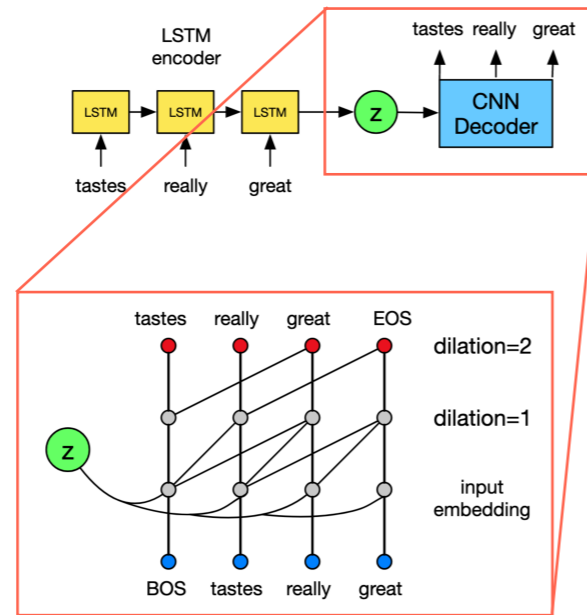
- * They propose to solve it by limiting the dependency structure generative model using dilated convolutions, preventing the model from seeing too much history
- * They find that while this helps and are first to get LL above RNNLM, but there is a clear tradeoff in decoder capacity and use of z

Xu and Durrett take a different tack:

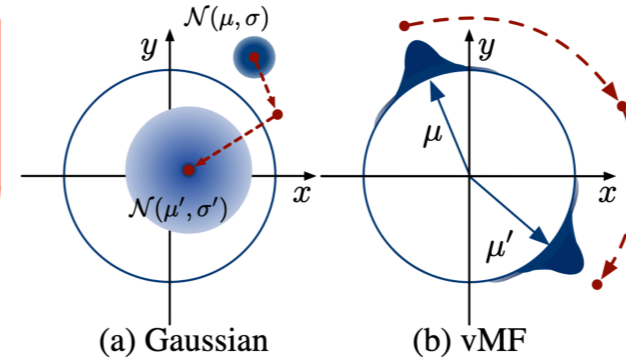
- * They address the problem by switching the latent var distribution to a Fisher vonMises dist, which puts mass on the unit hyper-sphere
- * By doing this, the KL term no longer depends on μ and effectively becomes a tunable hyperparameter of the model, allowing for tuning the balance of contribution between reconstruction and KL to LL of sentences
- * This works quite well in practice, but now we've introduced another hyperparameter, κ , that needs tuning

Mitigating Posterior Collapse

[Yang et al. 17]



[Xu & Durrett 18]



$$KL(q_{\kappa}(z_{\phi}(\alpha)) || p(\alpha)) \perp \phi$$

One of the first papers to successfully deal with this problem is Yang et al.

- * They propose to solve it by limiting the dependency structure generative model using dilated convolutions, preventing the model from seeing too much history
- * They find that while this helps and are first to get LL above RNNLM, but there is a clear tradeoff in decoder capacity and use of z

Xu and Durrett take a different tack:

- * They address the problem by switching the latent var distribution to a Fisher vonMises dist, which puts mass on the unit hyper-sphere
- * By doing this, the KL term no longer depends on mu and effectively becomes a tunable hyperparameter of the model, allowing for tuning the balance of contribution between reconstruction and KL to LL of sentences
- * This works quite well in practice, but now we've introduced another hyperparameter, kappa, that needs tuning

Mitigating Posterior Collapse

Simultaneous Updates $\phi_t^*, \theta_t^* = \arg \max_{\phi, \theta} L(\theta, \phi; X)$

Inner optimization, tighter lower bound ↓
 $\phi_t^* = \arg \max_{\phi} L(\theta_{t-1}^*, \phi; X), \theta_t^* = \arg \max_{\theta} L(\theta, \phi_t^*; X)$

15

Two more recent approaches approach the problem from a different perspective: changing the optimization problem

They both argue that the issue is the proposed variational parameters from the inference network are suboptimal and propose to solve it by optimizing the variational parameters in an inner loop

Kim et al solve this by using the inference net params as initializations for traditional meanfield stochastic variational inference, then solve that optimization in an inner loop with SGD and backprop through the entire process

- * This empirically helps significantly, and they are the first to get non-zero KLs in latent gaussian and SOTA lang model perplexities without compromising the generator
- * But their method is incredibly slow (10-15x slowdown)

He et al propose a much simpler modification:

- * They simply optimize the inference network to convergence on an inner loop **only for the first few epochs of training**
- * This leads to even better scores at only 3-5x slowdown

Mitigating Posterior Collapse

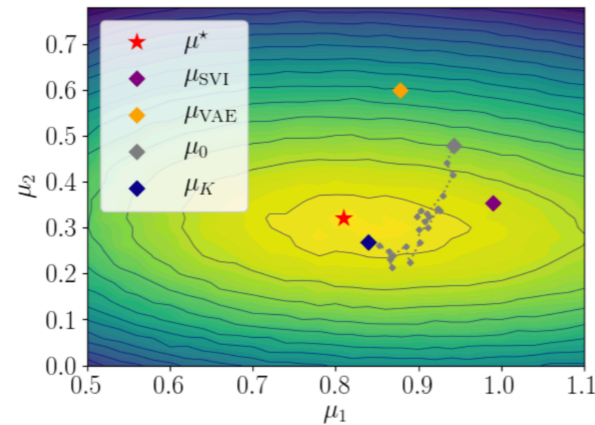
Simultaneous Updates $\phi_t^*, \theta_t^* = \arg \max_{\phi, \theta} L(\theta, \phi; X)$

Inner optimization, tighter lower bound



$\phi_t^* = \arg \max_{\phi} L(\theta_{t-1}^*, \phi; X), \theta_t^* = \arg \max_{\theta} L(\theta, \phi_t^*; X)$

[Kim et al. 18]



15

Two more recent approaches approach the problem from a different perspective: changing the optimization problem

They both argue that the issue is the proposed variational parameters from the inference network are suboptimal and propose to solve it by optimizing the variational parameters in an inner loop

Kim et al solve this by using the inference net params as initializations for traditional meanfield stochastic variational inference, then solve that optimization in an inner loop with SGD and backprop through the entire process

- * This empirically helps significantly, and they are the first to get non-zero KLs in latent gaussian and SOTA lang model perplexities without compromising the generator
- * But their method is incredibly slow (10-15x slowdown)

He et al propose a much simpler modification:

- * They simply optimize the inference network to convergence on an inner loop **only for the first few epochs of training**
- * This leads to even better scores at only 3-5x slowdown

Mitigating Posterior Collapse

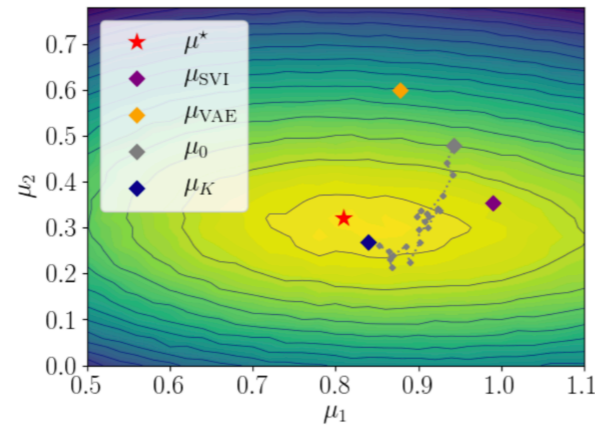
Simultaneous Updates $\phi_t^*, \theta_t^* = \arg \max_{\phi, \theta} L(\theta, \phi; X)$

Inner optimization, tighter lower bound

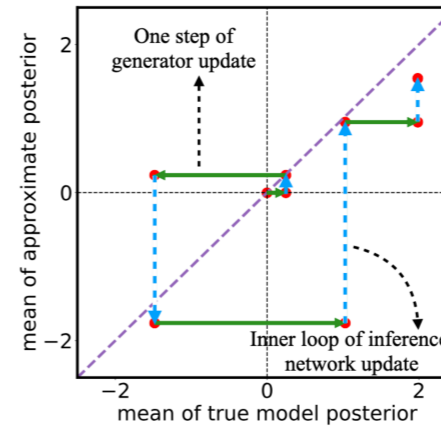


$\phi_t^* = \arg \max_{\phi} L(\theta_{t-1}^*, \phi; X), \theta_t^* = \arg \max_{\theta} L(\theta, \phi_t^*; X)$

[Kim et al. 18]



[He et al. 19]



Two more recent approaches approach the problem from a different perspective: changing the optimization problem

They both argue that the issue is the proposed variational parameters from the inference network are suboptimal and propose to solve it by optimizing the variational parameters in an inner loop

Kim et al solve this by using the inference net params as initializations for traditional meanfield stochastic variational inference, then solve that optimization in an inner loop with SGD and backprop through the entire process

- * This empirically helps significantly, and they are the first to get non-zero KLs in latent gaussian and SOTA lang model perplexities without compromising the generator
- * But their method is incredibly slow (10-15x slowdown)

He et al propose a much simpler modification:

- * They simply optimize the inference network to convergence on an inner loop **only for the first few epochs of training**
- * This leads to even better scores at only 3-5x slowdown

Mitigating Posterior Collapse

Paper, Task	Contributions	Limitations
[Yang et al. 17] sentence generation	First text VAE that outperforms autoregressive RNN-LM	Must restrict generative architecture to use limited historical context
[Xu & Durrett 18] sentence generation	Exchange latent Gaussian for vonMises-Fisher leads to better performance w/o collapse	Must treat latent variance as hyperparameter , else collapse comes back. Also, comparison is unfair
[Kim et al. 18] sentence & image generation	Use amortized prediction as initialization for SVI and backprop through it all	Very slow training (>10x vanilla VAE), difficult to implement
[He et al. 19] sentence & image generation	Aggressively optimize inference network until it matches model posterior	Still requires an inner optimization during early training stages

16

Posterior collapse is a massive issue for VAEs in NLP because of the strong autoregressive decoder that can easily do well in optimization w/o using the noisy latent variables

The first two methods propose to solve it by restricting the generative family/architecture in some way, but this trades away modeling flexibility

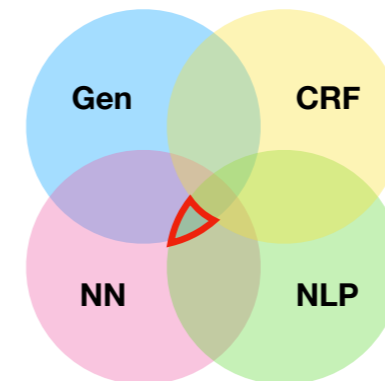
The second two propose a more principled approach that doesn't limit the model, but the trade off here now is that training is a bit slower.

Outline

- VAEs
 - Continuous Variables
 - Optimization Issues: Posterior Collapse
- **Topic Modeling**
 - Discrete Variables and Semi-supervised Learning
- Neural CRFs
 - Exact Inference
 - Approximate Inference
- VAEs for Discrete Structure: Semi-Supervised Learning
- Viewing Attention as a Latent Variable

[Srivastava & Sutton 17]

[Miao et al. 17]



17

All of the previous approaches were concerned with generation using continuous latent variables.

The next *topic* is using VAEs for topic models.

VAEs for topic models are better than traditional close-form update approaches for two reasons:

- * Neural nets can easily cope with additional context information, allowing custom flavors of topic models to be rapidly developed
- * Amortized inference in topic models allows for quick inference at test time

VAEs for Topic Models

18

Srivastava and Sutton provide the first successful VAE for LDA.

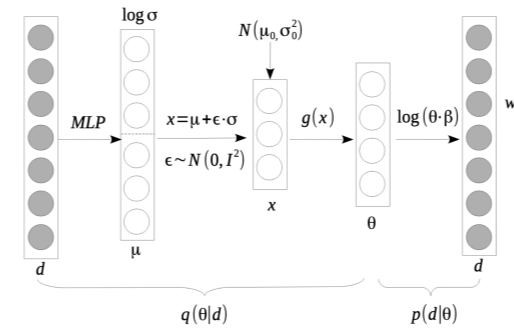
- * They do this on the collapsed model by constructing a Laplace approximation to the Dirichlet prior and inferring/sampling topic distributions θ using a logistic normal.
- * The approach works, but in practice they find that optimization is brittle and must be carefully tuned to avoid collapse of the inferred θ s to either 1-hot or uniform optima

Miao et al extend this model by constructing a potentially infinite topic model

- * They do this by using the stick-breaking construction of topic proportions in θ and parameterize the construction with an RNN. The whole thing is differentiable which is very cool
- * One issue is that they must decide the number of “active” topics by have the RNN propose an extra topic on each minibatch and measuring if there’s a change in the ELBO w/ and w/o the additional topic — they do not “infer” the number of topics for each document

VAEs for Topic Models

[Srivastava & Sutton 17]



18

Srivastava and Sutton provide the first successful VAE for LDA.

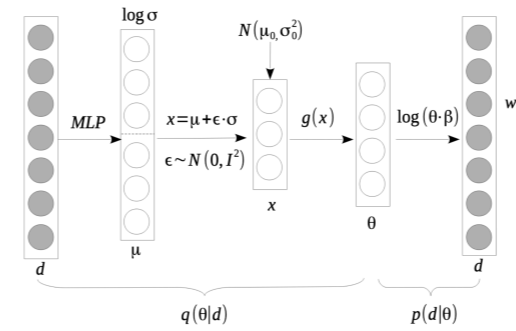
- * They do this on the collapsed model by constructing a Laplace approximation to the Dirichlet prior and inferring/sampling topic distributions θ using a logistic normal.
- * The approach works, but in practice they find that optimization is brittle and must be carefully tuned to avoid collapse of the inferred θ s to either 1-hot or uniform optima

Miao et al extend this model by constructing a potentially infinite topic model

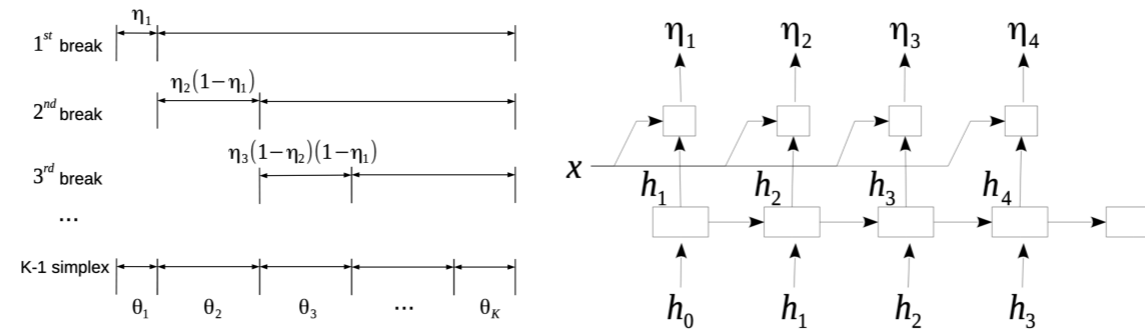
- * They do this by using the stick-breaking construction of topic proportions in θ and parameterize the construction with an RNN. The whole thing is differentiable which is very cool
- * One issue is that they must decide the number of “active” topics by having the RNN propose an extra topic on each minibatch and measuring if there’s a change in the ELBO w/ and w/o the additional topic — they do not “infer” the number of topics for each document

VAEs for Topic Models

[Srivastava & Sutton 17]



[Miao et al. 17]



Srivastava and Sutton provide the first successful VAE for LDA.

- * They do this on the collapsed model by constructing a Laplace approximation to the Dirichlet prior and inferring/sampling topic distributions theta using a logistic normal.
- * The approach works, but in practice they find that optimization is brittle and must be carefully tuned to avoid collapse of the inferred thetas to either 1-hot or uniform optima

Miao et al. extend this model by constructing a potentially infinite topic model

- * They do this by using the stick-breaking construction of topic proportions in theta and parameterize the construction with an RNN. The whole thing is differentiable which is very cool
- * One issue is that they must decide the number of “active” topics by having the RNN propose an extra topic on each minibatch and measuring if there’s a change in the ELBO w/ and w/o the additional topic — they do not “infer” the number of topics for each document

VAEs for Topic Models

Paper, Task	Contributions	Limitations
[Srivastava & Sutton 17] bag-of-words representation	Showed how to successfully train reparameterized, amortized inference for LDA	Brittle — Optimization must be carefully tuned to avoid posterior collapse
[Miao et al. 17] bag-of-words representation	Extends VAE-LDA to predict variable number of topics	Number of topics is not inferred — requires repeated evaluations of likelihood for stopping criterion

19

Amortized inference for topic models is an interesting application of VAEs because they are widely used models and alleviating the need for optimization at test time makes them much more applicable.

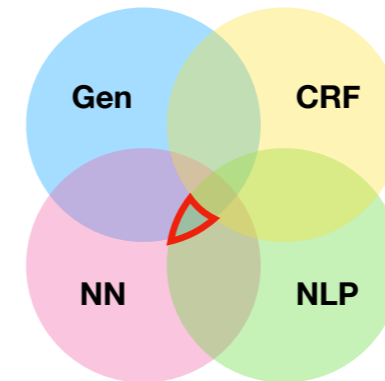
Further, the use of NN parameterizations allows for quick, flexible alterations of the model (such as conditioning on other context) that previously was difficult using mean field with coordinate ascent.

Outline

- VAEs
 - Continuous Variables
 - Optimization Issues: Posterior Collapse
 - Topic Modeling
- **Discrete Variables and Semi-supervised Learning**
- Neural CRFs
 - Exact Inference
 - Approximate Inference
- VAEs for Discrete Structure: Semi-Supervised Learning
- Viewing Attention as a Latent Variable

[Wen et al. 17]

[Yang et al. 17] (again)



20

Ok, so far our latent variables have all been continuous
(the topic model papers dealt marginalized the discrete latent variables out)

Next we'll discuss two papers that use discrete latent variables, that correspond to some classification of the input, and they can derive learning signal for the classifier on unlabeled data using the ELBO in addition to supervised data for better performance

We'll see that you make minor changes the variational objective to cope with observed data

Classification as Inference in Semi-supervised VAEs

Data $\mathcal{D} = \{x^k\}_{k=1}^{D_U} \cup \{(x^k, y^k)\}_{k=1}^{D_L}$

21

The semi-supervised VAE setup is as follows:

- * Now we have additional data that's labeled
- * And our model now depends on a per-instance discrete variable

Then we can write down a pair of ELBOs, one for each situation

- * This yields the final update, which has an extra cross entropy term for optimizing $q(y)$ on the labeled data, otherwise there's no update for $q(y)$ on the labeled data

Classification as Inference in Semi-supervised VAEs

Data $\mathcal{D} = \{x^k\}_{k=1}^{D_U} \cup \{(x^k, y^k)\}_{k=1}^{D_L}$

Model $p_\theta(x^k, z^k, y^k) = p_\theta(x^k | z^k, y^k) p(z^k) p(y^k)$

21

The semi-supervised VAE setup is as follows:

- * Now we have additional data that's labeled
- * And our model now depends on a per-instance discrete variable

Then we can write down a pair of ELBOs, one for each situation

- * This yields the final update, which has an extra cross entropy term for optimizing $q(y)$ on the labeled data, otherwise there's no update for $q(y)$ on the labeled data

Classification as Inference in Semi-supervised VAEs

Data $\mathcal{D} = \{x^k\}_{k=1}^{D_U} \cup \{(x^k, y^k)\}_{k=1}^{D_L}$

Model $p_\theta(x^k, z^k, y^k) = p_\theta(x^k | z^k, y^k) p(z^k) p(y^k)$

ELBO (per k)

$$J_U(\theta, \phi; x) = \mathbb{E}_{q_\phi(z|y)q_\phi(y)}[\log p_\theta(x|y, z)] - KL(q_\phi(z|y) || p(z)) - KL(q_\phi(y) || p(y))$$

21

The semi-supervised VAE setup is as follows:

- * Now we have additional data that's labeled
- * And our model now depends on a per-instance discrete variable

Then we can write down a pair of ELBOs, one for each situation

- * This yields the final update, which has an extra cross entropy term for optimizing $q(y)$ on the labeled data, otherwise there's no update for $q(y)$ on the labeled data

Classification as Inference in Semi-supervised VAEs

Data $\mathcal{D} = \{x^k\}_{k=1}^{D_U} \cup \{(x^k, y^k)\}_{k=1}^{D_L}$

Model $p_\theta(x^k, z^k, y^k) = p_\theta(x^k | z^k, y^k) p(z^k) p(y^k)$

ELBO (per k)

$$J_U(\theta, \phi; x) = \mathbb{E}_{q_\phi(z|y)q_\phi(y)}[\log p_\theta(x|y, z)] - KL(q_\phi(z|y) || p(z)) - KL(q_\phi(y) || p(y))$$

$$J_L(\theta, \phi; x, y) = \mathbb{E}_{q_\phi(z|y, x)}[\log p_\theta(x|y, z)] - KL(q_\phi(z|y) || p(z)) + \log p(y)$$

21

The semi-supervised VAE setup is as follows:

- * Now we have additional data that's labeled
- * And our model now depends on a per-instance discrete variable

Then we can write down a pair of ELBOs, one for each situation

- * This yields the final update, which has an extra cross entropy term for optimizing $q(y)$ on the labeled data, otherwise there's no update for $q(y)$ on the labeled data

Classification as Inference in Semi-supervised VAEs

Data $\mathcal{D} = \{x^k\}_{k=1}^{D_U} \cup \{(x^k, y^k)\}_{k=1}^{D_L}$

Model $p_\theta(x^k, z^k, y^k) = p_\theta(x^k | z^k, y^k)p(z^k)p(y^k)$

ELBO (per k)

$$J_U(\theta, \phi; x) = \mathbb{E}_{q_\phi(z|y)q_\phi(y)}[\log p_\theta(x|y, z)] - KL(q_\phi(z|y) || p(z)) - KL(q_\phi(y) || p(y))$$

$$J_L(\theta, \phi; x, y) = \mathbb{E}_{q_\phi(z|y, x)}[\log p_\theta(x|y, z)] - KL(q_\phi(z|y) || p(z)) + \log p(y)$$

$$J(\theta, \phi; \mathcal{D}) = \mathbb{E}_{(x) \sim \mathcal{D}_U}[J_U] + \mathbb{E}_{(x, y) \sim \mathcal{D}_L}[J_L] + \alpha \mathbb{E}_{(x, y) \sim \mathcal{D}_U}[\log q(y|x)]$$

21

The semi-supervised VAE setup is as follows:

- * Now we have additional data that's labeled
- * And our model now depends on a per-instance discrete variable

Then we can write down a pair of ELBOs, one for each situation

- * This yields the final update, which has an extra cross entropy term for optimizing $q(y)$ on the labeled data, otherwise there's no update for $q(y)$ on the labeled data

Classification as Inference in Semi-supervised VAEs

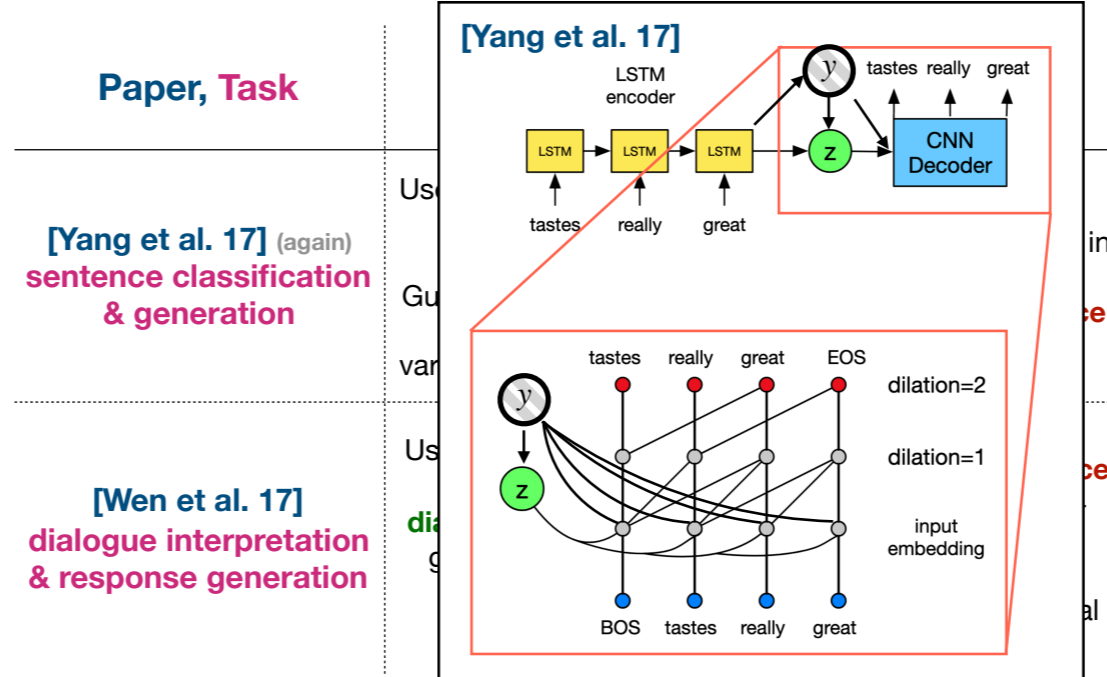
Paper, Task	Contributions	Limitations
[Yang et al. 17] (again) sentence classification & generation	Use semi-supervised VAE for learning text-classification and Gumbel-Softmax samples for reduced gradient variance over REINFORCE	Observe large trade-off in classification and generation performance
[Wen et al. 17] dialogue interpretation & response generation	Use semi-supervised VAE for learning to infer dialogue intentions then generate responses in complex neural architecture	Variational objective yields poor performance — model must be fine-tuned with RL on task success objective. Even then, only marginal improvement

22

The idea of treating annotations as latent variables and doing semi-supervised with a VAE is a very interesting one

- * Yang et al do this for classification, although they find there is a tradeoff in classification and generation performance — as modelers, we must decide which is more important to us
- * Wen et al do something similar by embedding classification of user “intention” as a submodule in a complex neural conversational agent.
- * They however use REINFORCE and find that optimization of the variational objective alone leads to performance well below state of the art

Classification as Inference in Semi-supervised VAEs



The idea of treating annotations as latent variables and doing semi-supervised with a VAE is a very interesting one

- * Yang et al do this for classification, although they find there is a tradeoff in classification and generation performance — as modelers, we must decide which is more important to us
- * Wen et al do something similar by embedding classification of user “intention” as a submodule in a complex neural conversational agent.
- * They however use REINFORCE and find that optimization of the variational objective alone leads to performance well below state of the art

Classification as Inference in Semi-supervised VAEs

Paper, Task	Contributions	Limitations
[Yang et al. 17] (again) sentence classification & generation	Use semi-supervised VAE for learning text-classification and Gumbel-Softmax samples for reduced gradient variance over REINFORCE	Observe large trade-off in classification and generation performance
[Wen et al. 17] dialogue interpretation & response generation	Use semi-supervised VAE for learning to infer dialogue intentions then generate responses in complex neural architecture	Variational objective yields poor performance — model must be fine-tuned with RL on task success objective. Even then, only marginal improvement

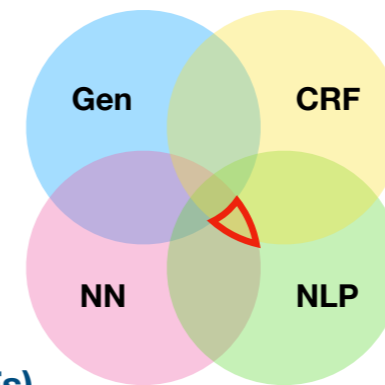
22

The idea of treating annotations as latent variables and doing semi-supervised with a VAE is a very interesting one

- * Yang et al do this for classification, although they find there is a tradeoff in classification and generation performance — as modelers, we must decide which is more important to us
- * Wen et al do something similar by embedding classification of user “intention” as a submodule in a complex neural conversational agent.
- * They however use REINFORCE and find that optimization of the variational objective alone leads to performance well below state of the art

Outline

- VAEs
 - Continuous Variables
 - Optimization Issues: Posterior Collapse
 - Topic Modeling
 - Discrete Variables and Semi-supervised Learning
- **Neural Conditional Random Fields (CRFs)**
 - **Exact Inference**
 - Approximate Inference
 - VAEs for Discrete Structure: Semi-Supervised Learning
 - Viewing Attention as a Latent Variable



[Lample et al. 16]

[Greenberg et al. 18]

[Durrett & Klein 15]

[Kitaev & Klein 18]

Now we'll take a brief tour through applications of undirected graphical models — conditional random fields with neural factors.

Structured outputs are common in many NLP tasks, such as tagging and parsing, and CRFs represent a strong class of models for enforcing structural constraints in these problems, while neural nets have become standard for their representational capacity. The next set of papers illustrate that these two modeling techniques are mutually beneficial

Neural CRFs: Sequences

$$p(y_{1:N}|x) = \frac{\exp\{\psi_{\theta}(x, y_{1:N})\}}{\sum_{y'_{1:N} \in \mathcal{Y}} \exp\{\psi_{\theta}(x, y'_{1:N})\}} = \frac{\exp\{\psi_{\theta}(x, y_{1:N})\}}{Z(\theta, x)}$$

24

A prime example of neural CRF use in NLP are tagging problems, where we have specific dependencies between neighboring tags, since they follow a grammar.

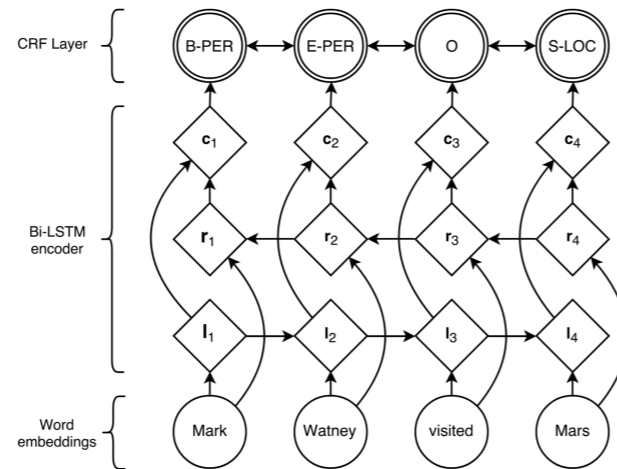
Lample et al were the first to illustrate the mutual benefits of a neural sequence CRF for named entity recognition, learning language-agnostic orthographic features (what names look like at the character level) completely from data and using a CRF layer on top to alleviate decoding errors made by independent-output neural models

More recently, Greenberg et al showed that data from multiple potentially overlapping annotation datasets (for differing tasks) exhibiting partially overlapping label sets could be combined by optimizing the marginal likelihoods of the observed labels. This wouldn't work in an independent model, since unobserved tags have no effect on neighboring tag — the model could never learn to predict the O tag

Neural CRFs: Sequences

$$p(y_{1:N}|x) = \frac{\exp\{\psi_{\theta}(x, y_{1:N})\}}{\sum_{y'_{1:N} \in \mathcal{Y}} \exp\{\psi_{\theta}(x, y'_{1:N})\}} = \frac{\exp\{\psi_{\theta}(x, y'_{1:N})\}}{Z(\theta, x)}$$

[Lample et al. 16]



24

A prime example of neural CRF use in NLP are tagging problems, where we have specific dependencies between neighboring tags, since they follow a grammar.

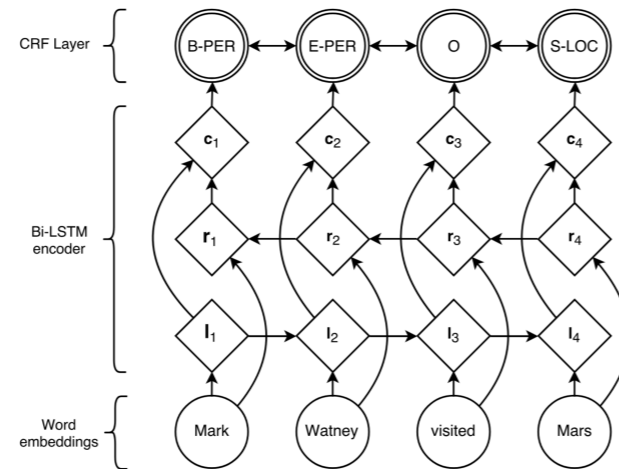
Lample et al were the first to illustrate the mutual benefits of a neural sequence CRF for named entity recognition, learning language-agnostic orthographic features (what names look like at the character level) completely from data and using a CRF layer on top to alleviate decoding errors made by independent-output neural models

More recently, Greenberg et al showed that data from multiple potentially overlapping annotation datasets (for differing tasks) exhibiting partially overlapping label sets could be combined by optimizing the marginal likelihoods of the observed labels. This wouldn't work in an independent model, since unobserved tags have no effect on neighboring tag — the model could never learn to predict the O tag

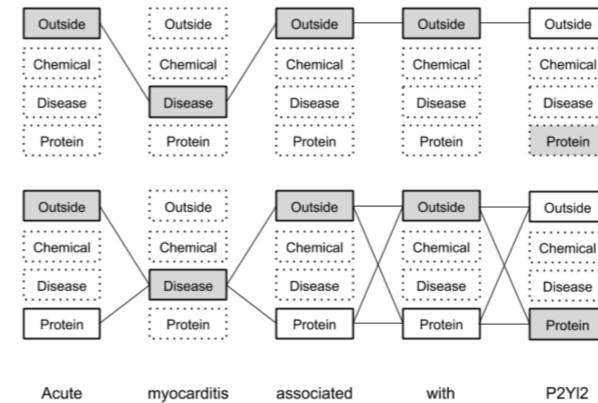
Neural CRFs: Sequences

$$p(y_{1:N}|x) = \frac{\exp\{\psi_{\theta}(x, y_{1:N})\}}{\sum_{y'_{1:N} \in \mathcal{Y}} \exp\{\psi_{\theta}(x, y'_{1:N})\}} = \frac{\exp\{\psi_{\theta}(x, y'_{1:N})\}}{Z(\theta, x)}$$

[Lample et al. 16]



[Greenberg et al. 18]



A prime example of neural CRF use in NLP are tagging problems, where we have specific dependencies between neighboring tags, since they follow a grammar.

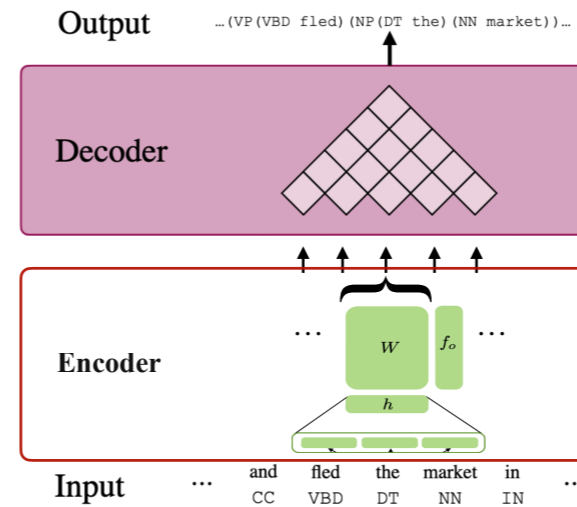
Lample et al were the first to illustrate the mutual benefits of a neural sequence CRF for named entity recognition, learning language-agnostic orthographic features (what names look like at the character level) completely from data and using a CRF layer on top to alleviate decoding errors made by independent-output neural models

More recently, Greenberg et al showed that data from multiple potentially overlapping annotation datasets (for differing tasks) exhibiting partially overlapping label sets could be combined by optimizing the marginal likelihoods of the observed labels. This wouldn't work in an independent model, since unobserved tags have no effect on neighboring tag — the model could never learn to predict the O tag

Neural CRFs: Trees

$$p(T|x) = \frac{\exp\{\psi_\theta(x, T)\}}{\sum_{T \in \mathcal{T}} \exp\{\psi_\theta(x, T)\}} = \frac{\exp\{\psi_\theta(x, T)\}}{Z(\theta, x)}$$

[Durrett & Klein 15]



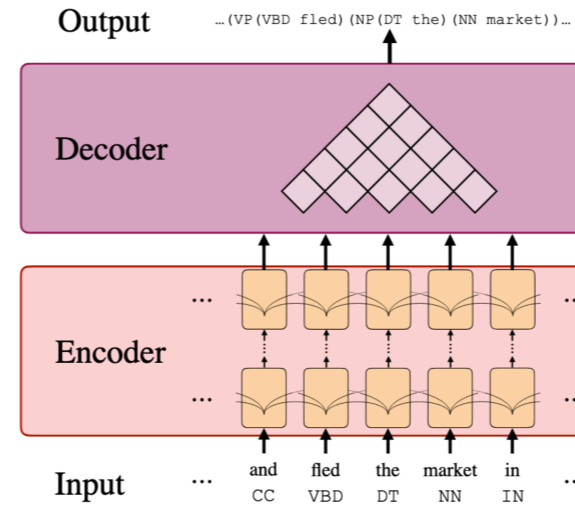
A similar story can be told with tree-parsing. Durrett and Klein were the first to introduce neural crf parsing and their relatively simple learned MLP features outperformed hand-crafted features.

Years later, Kitaev and Klein showed that a dramatic increase in performance can be obtained just by switching out the encoder layer with state-of-the-art sentence encoding architectures, illustrating that the fusion of CRFs with neural nets are essentially decoupled.

Neural CRFs: Trees

$$p(T|x) = \frac{\exp\{\psi_\theta(x, T)\}}{\sum_{T \in \mathcal{T}} \exp\{\psi_\theta(x, T)\}} = \frac{\exp\{\psi_\theta(x, T)\}}{Z(\theta, x)}$$

[Durrett & Klein 15]
[Kitaev & Klein 18]



A similar story can be told with tree-parsing. Durrett and Klein were the first to introduce neural crf parsing and their relatively simple learned MLP features outperformed hand-crafted features.

Years later, Kitaev and Klein showed that a dramatic increase in performance can be obtained just by switching out the encoder layer with state-of-the-art sentence encoding architectures, illustrating that the fusion of CRFs with neural nets are essentially decoupled.

Exact Neural CRFs

Paper, Task	Contributions	Limitations
[Lample et al. 16] named entity recognition	First to show that tagging CRF can be combined with powerful neural features	Tagging CRF known to be suboptimal for segmentation
[Greenberg et al. 18] named entity recognition	Combine annotations from multiple datasets using CRF	Requires unequal tagsets for different datasets or it will not learn to predict "O"
[Durrett & Klein 15] constituency-syntax parsing	First to use tree CRF with neural features	Uses simplistic MLP features
[Kitaev & Klein 18] constituency-syntax parsing	Same CRF, but huge improvements using SOTA neural text encoding architecture	Further improvements can be obtained with BERT encoder pretraining

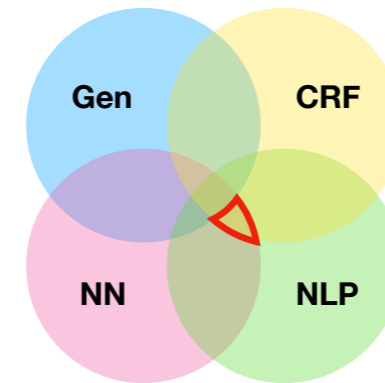
26

In NLP our output is often heavily structured, and conditional random fields are a great tool for representing the dependencies among outputs,

* Further they are completely amenable to having their potentials predicted using neural networks and are end-to-end trainable thanks to the differentiability of the sum-product algorithm

Outline

- VAEs
 - Continuous Variables
 - Optimization Issues: Posterior Collapse
 - Topic Modeling
 - Discrete Variables and Semi-supervised Learning
- Neural CRFs
 - Exact Inference
 - **Approximate Inference**
- VAEs for Discrete Structure: Semi-Supervised Learning
- Viewing Attention as a Latent Variable



[Ganea & Hofmann 17]

[Andor et al. 16]

Often we want to learn models with globally normalized structures that do not permit tractable exact inference.

These next two papers illustrate how to exploit approximate inference in neural and still learn the models end-to-end.

Handling Intractable CRFs

$Z(\theta, x)$ is often intractable for interesting joint factorizations

Credit: [<https://www.cs.cmu.edu/~epxing/Class/10715/lectures/lecture12-CRF.pdf>]

28

Unless the CRF exhibits a factorization that allows for computation of the partition function in polynomial time with dynamic programming, we must approximate it.

Ganea and Hofmann demonstrated one way of doing this in their neural entity-linking model.

- * The model incorporates pair-wise plausibility factors between all entities, allowing for a global disambiguation and learn the model by using truncated loopy belief propagation and optimizing the approximate marginals.
- * LBP is differentiable and so the model can be learned end-to-end.
- * LBP however has quadratic runtime which makes this model extremely slow for documents of appreciable size.

Andor et al use a different approach to mitigate the well-known issue of label bias in locally normalized structured output models, with direct application to dependency parsing.

- * They first train the model with the local objective, but then continue training using beam-search on the unnormalized scores and approximate the partition function using the mass on the beam
- * They find that this significantly reduces label-bias.

Handling Intractable CRFs

$Z(\theta, x)$ is often intractable for interesting joint factorizations

[Ganea & Hofmann 17]
$$p_{\theta}(e_1, \dots, e_M | x) = \exp\left\{ \sum_{i=1}^M [\Psi_{\theta}(e_i) + \sum_{j<i} \Phi_{\theta}(e_i, e_j)] \right\} / Z(\theta, x)$$

Credit: [<https://www.cs.cmu.edu/~epxing/Class/10715/lectures/lecture12-CRF.pdf>]

28

Unless the CRF exhibits a factorization that allows for computation of the partition function in polynomial time with dynamic programming, we must approximate it.

Ganea and Hofmann demonstrated one way of doing this in their neural entity-linking model.

- * The model incorporates pair-wise plausibility factors between all entities, allowing for a global disambiguation and learn the model by using truncated loopy belief propagation and optimizing the approximate marginals.
- * LBP is differentiable and so the model can be learned end-to-end.
- * LBP however has quadratic runtime which makes this model extremely slow for documents of appreciable size.

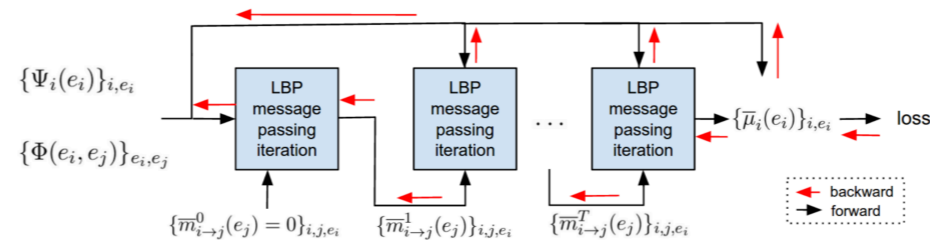
Andor et al use a different approach to mitigate the well-known issue of label bias in locally normalized structured output models, with direct application to dependency parsing.

- * They first train the model with the local objective, but then continue training using beam-search on the unnormalized scores and approximate the partition function using the mass on the beam
- * They find that this significantly reduces label-bias.

Handling Intractable CRFs

$Z(\theta, x)$ is often intractable for interesting joint factorizations

[Ganea & Hofmann 17]
$$p_{\theta}(e_1, \dots, e_M | x) = \exp\left\{ \sum_{i=1}^M [\Psi_{\theta}(e_i) + \sum_{j<i} \Phi_{\theta}(e_i, e_j)] \right\} / Z(\theta, x)$$



Credit: <https://www.cs.cmu.edu/~epxing/Class/10715/lectures/lecture12-CRF.pdf>

Unless the CRF exhibits a factorization that allows for computation of the partition function in polynomial time with dynamic programming, we must approximate it.

Ganea and Hofmann demonstrated one way of doing this in their neural entity-linking model.

- * The model incorporates pair-wise plausibility factors between all entities, allowing for a global disambiguation and learn the model by using truncated loopy belief propagation and optimizing the approximate marginals.
- * LBP is differentiable and so the model can be learned end-to-end.
- * LBP however has quadratic runtime which makes this model extremely slow for documents of appreciable size.

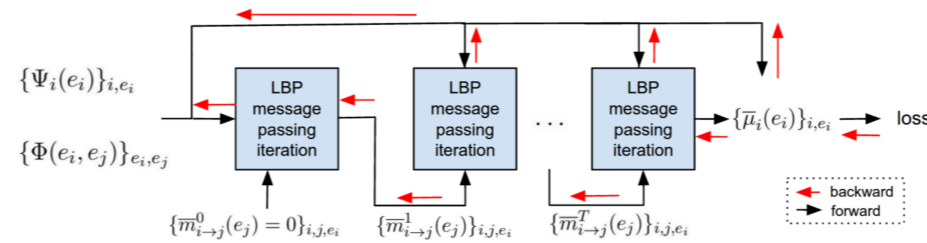
Andor et al use a different approach to mitigate the well-known issue of label bias in locally normalized structured output models, with direct application to dependency parsing.

- * They first train the model with the local objective, but then continue training using beam-search on the unnormalized scores and approximate the partition function using the mass on the beam
- * They find that this significantly reduces label-bias.

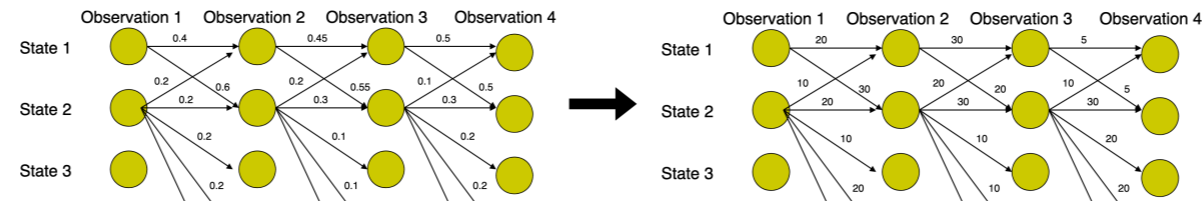
Handling Intractable CRFs

$Z(\theta, x)$ is often intractable for interesting joint factorizations

[Ganea & Hofmann 17]
$$p_{\theta}(e_1, \dots, e_M | x) = \exp\left\{ \sum_{i=1}^M [\Psi_{\theta}(e_i) + \sum_{j<i} \Phi_{\theta}(e_i, e_j)] \right\} / Z(\theta, x)$$



[Andor et al. 16]



Credit: <https://www.cs.cmu.edu/~epxing/Class/10715/lectures/lecture12-CRF.pdf>

Unless the CRF exhibits a factorization that allows for computation of the partition function in polynomial time with dynamic programming, we must approximate it.

Ganea and Hofmann demonstrated one way of doing this in their neural entity-linking model.

- * The model incorporates pair-wise plausibility factors between all entities, allowing for a global disambiguation and learn the model by using truncated loopy belief propagation and optimizing the approximate marginals.
- * LBP is differentiable and so the model can be learned end-to-end.
- * LBP however has quadratic runtime which makes this model extremely slow for documents of appreciable size.

Andor et al use a different approach to mitigate the well-known issue of label bias in locally normalized structured output models, with direct application to dependency parsing.

- * They first train the model with the local objective, but then continue training using beam-search on the unnormalized scores and approximate the partition function using the mass on the beam
- * They find that this significantly reduces label-bias.

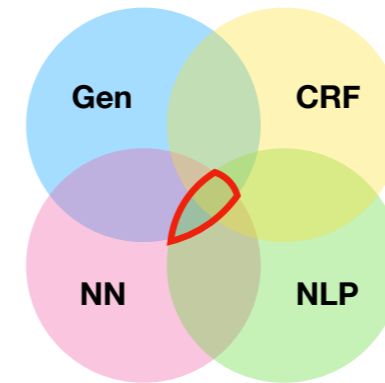
Approximate Neural CRFs

Paper, Task	Contributions	Limitations
[Ganea & Hofmann 17] entity linking	Backprop through loopy belief-propagation to handle intractable CRF	LBP has quadratic runtime, severely slowing down training and limiting size of possible candidate set
[Andor et al. 16] dependency-syntax parsing, part-of-speech tagging, sentence compression	Mitigate label-bias problem using intractable CRF and beam-search to approximate the partition function	Inexact: hoping beam approximates partition, difficult to implement

Both of these papers provide alternative approaches to learning intractable neural crfs without sacrificing end-to-end training by backproping through the approximations.

Outline

- VAEs
 - Continuous Variables
 - Optimization Issues: Posterior Collapse
 - Topic Modeling
 - Discrete Variables and Semi-supervised Learning
- Neural CRFs
 - Exact Inference
 - Approximate Inference
- **VAEs for Discrete Structure: Semi-Supervised Learning**
 - Viewing Attention as a Latent Variable



[Miao & Blunsom 16b]

[Yin et al. 18]

[Zhang et al. 17]

[Corro & Titov 19]

Ok, now that we've discussed VAEs and Neural CRFs, we're ready to discuss methods that for semi-supervised learning of discrete **structured** models.

Structured prediction is a crucial subfield of NLP as complex tasks have considerable dependencies among their outputs, while the complexity of the annotations also makes them more costly to obtain.

What we'll see is that we can take our conditional structured output model and embed it as inference in a generative model.

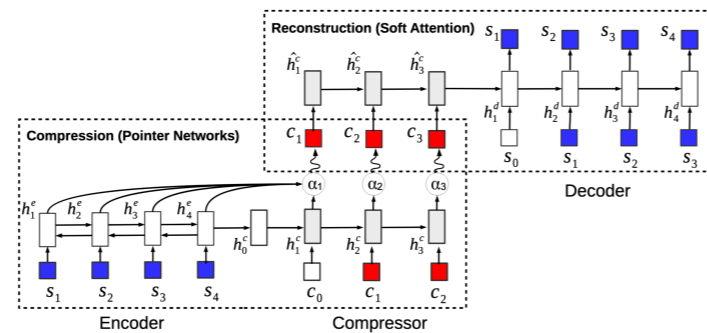
This allows for us to learn on additional unlabeled data by deriving learning signal from reconstructing the input and regularizing inference to the prior (which can be learned)

Semi-Supervised Learning for Structured Prediction

[Miao & Blunsom 16b]

$$p_{\theta}(c_1, \dots, c_M, s_1, \dots, s_N)$$

$$= \prod_{i=1}^M p_{\theta}(c_i | c_{<i}) \prod_{j=1}^N p_{\theta}(s_j | s_{<j}, c_{1:M})$$



31

First, Miao and Blunsom introduced a VAE model of sentence compression and generation where the latent variables are entire discrete sequences (the compressions).

- * Their semi-supervised approach first trained both the inference, generative, and **prior** language model on the supervised data
- * They then continued training on unlabeled data, optimizing the inference model using the REINFORCE gradient estimator
- * What's interesting here is their use of a prior compression language model — this can be seen as an empirical bayesian prior

Yin et al do a very similar thing for learning to predict semantic parses from natural language by first converting the tree problem to a sequence problem by linearizing the trees.

- * They then follow Miao and Blunsom in training

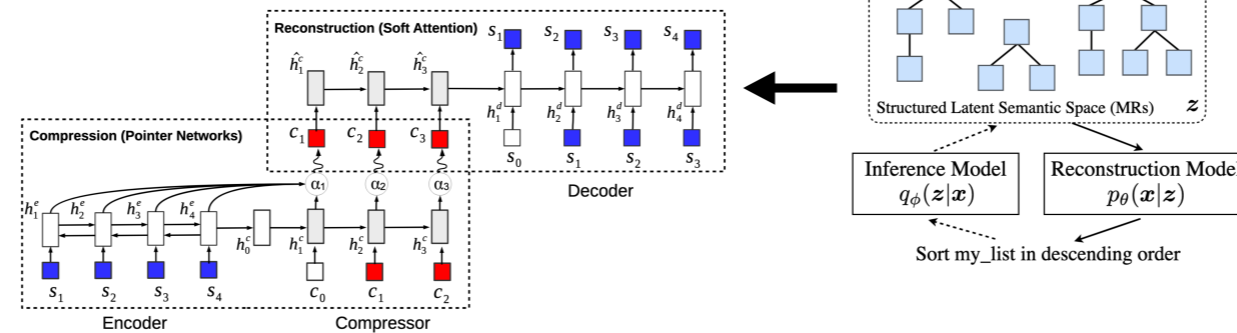
Semi-Supervised Learning for Structured Prediction

[Miao & Blunsom 16b]

[Yin et al. 18]

$$p_{\theta}(c_1, \dots, c_M, s_1, \dots, s_N)$$

$$= \prod_{i=1}^M p_{\theta}(c_i | c_{<i}) \prod_{j=1}^N p_{\theta}(s_j | s_{<j}, c_{1:M})$$



31

First, Miao and Blunsom introduced a VAE model of sentence compression and generation where the latent variables are entire discrete sequences (the compressions).

- * Their semi-supervised approach first trained both the inference, generative, and **prior** language model on the supervised data
- * They then continued training on unlabeled data, optimizing the inference model using the REINFORCE gradient estimator
- * What's interesting here is their use of a prior compression language model — this can be seen as an empirical bayesian prior

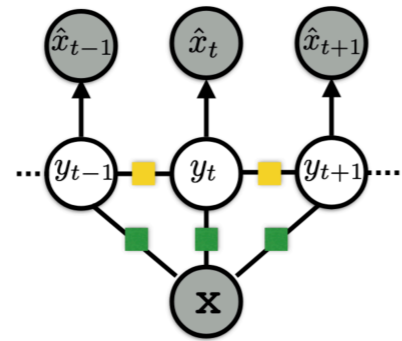
Yin et al do a very similar thing for learning to predict semantic parses from natural language by first converting the tree problem to a sequence problem by linearizing the trees.

- * They then follow Miao and Blunsom in training

Semi-Supervised Learning for Structured Prediction

[Zhang et al. 17]

$$p_{\theta}(\hat{x}_{1:N} | x_{1:N}) \\ = \sum_{y_{1:N} \in \mathcal{Y}} [p_{\theta}(y_{1:N} | x_{1:N}) \prod_{i=1}^N p_{\theta}(\hat{x}_i | y_i)]$$



32

The previous two papers were for locally normalized latent sequences, but these next two embed CRFs as the inference models in VAE-like objectives.

Zhang et al don't quite formulate the problem as VAE, they instead consider a conditional model for reconstructing the input through an unobserved CRF — there is no “prior” on the tag sequences.

* They then use an extremely simple model words given tags, which allows them to calculate the marginal probability of reconstruction given the input using the forward algorithm.

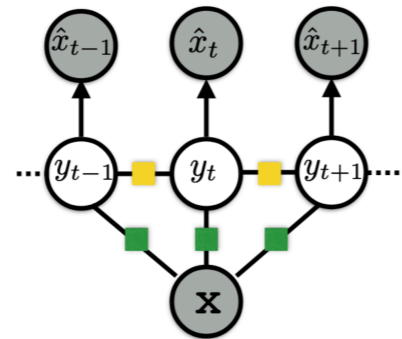
* They do this for semi-supervised part of speech tagging and see decent improvements

More recently, Corro and Titov use the generalized perturb-and-map to get samples of dependency parse trees through the Eisner CRF algorithm by adding independent gumbel noise to the CRF factors and relaxing the argmax to a softmax, which yields “soft” dependency trees as samples. They then embed this as the inference network in a VAE and find that it, like Zhang, yields considerable improvements over a supervised model.

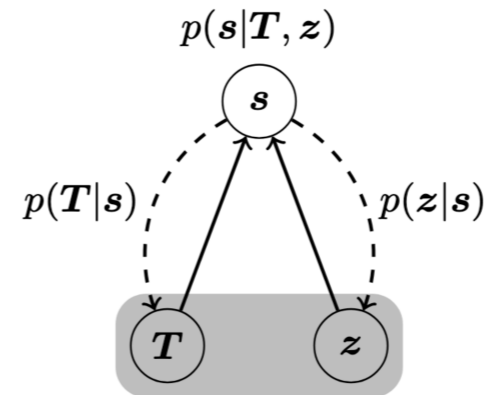
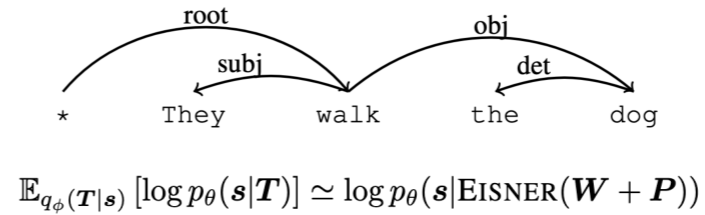
Semi-Supervised Learning for Structured Prediction

[Zhang et al. 17]

$$p_{\theta}(\hat{x}_{1:N} | x_{1:N}) = \sum_{y_{1:N} \in \mathcal{Y}} [p_{\theta}(y_{1:N} | x_{1:N}) \prod_{i=1}^N p_{\theta}(\hat{x}_i | y_i)]$$



[Corro & Titov 19]



32

The previous two papers were for locally normalized latent sequences, but these next two embed CRFs as the inference models in VAE-like objectives.

Zhang et al don't quite formulate the problem as VAE, they instead consider a conditional model for reconstructing the input through an unobserved CRF — there is no “prior” on the tag sequences.

* They then use an extremely simple model words given tags, which allows them to calculate the marginal probability of reconstruction given the input using the forward algorithm.

* They do this for semi-supervised part of speech tagging and see decent improvements

More recently, Corro and Titov use the generalized perturb-and-map to get samples of dependency parse trees through the Eisner CRF algorithm by adding independent gumbel noise to the CRF factors and relaxing the argmax to a softmax, which yields “soft” dependency trees as samples. They then embed this as the inference network in a VAE and find that it, like Zhang, yields considerable improvements over a supervised model.

Semi-Supervised Learning for Structured Prediction

Paper, Task	Contributions	Limitations
[Miao & Blunsom 16b] sentence (de)compression	Embed seq2seq model in VAE and learn compression as latent variable sequence	Use of REINFORCE to yields limited improvement from unsupervised data
[Yin et al. 18] program semantic parsing	Semi-supervised VAE training for semantic parsing	Requires linearization of the tree, which ignores some of the problem structure
[Zhang et al. 17] part of speech tagging	Embed sequence CRF as inference network with simple generative model for tractable EM training	Generative model must be restricted to make objective tractable
[Corro & Titov 19] dependency-syntax parsing	Semi-supervised training of tree parsing, relaxed perturb-and-map samples through dynamic program	Requires architecture that can cope with "soft" dependency trees

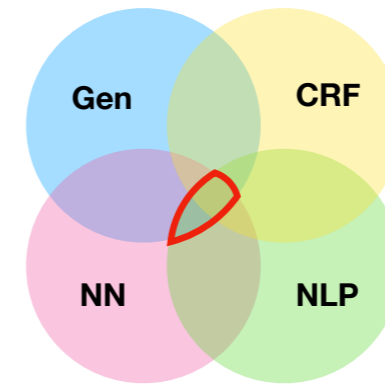
33

The first two papers show how to train VAEs for semi-supervised learning with locally normalized inference distributions, while the second two show how to embed CRFs as latent variables and learn end-to-end

I think semi-supervised learning with structured latent variables is an exciting direction in the field.

Outline

- VAEs
 - Continuous Variables
 - Optimization Issues: Posterior Collapse
 - Topic Modeling
 - Discrete Variables and Semi-supervised Learning
- Neural CRFs
 - Exact Inference
 - Approximate Inference
- VAEs for Discrete Structure: Semi-Supervised Learning
- **Viewing Attention as a Latent Variable**



[Lei et al. 16]

[Deng et al. 18]

[Le & Titov 18]

[Kim et al. 17]

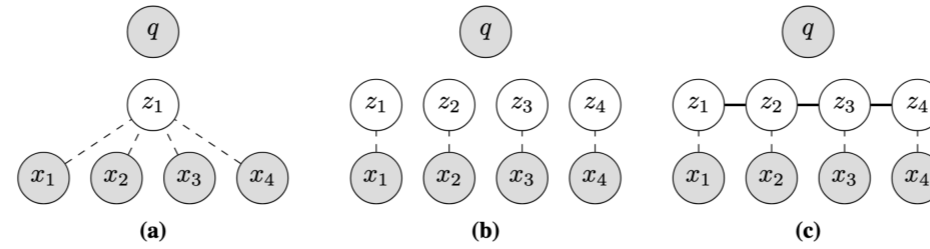
[Strubell et al. 18]

Ok, finally we'll discuss the intersection of attention mechanisms, which have become a workhorse mechanism in NLP and their interpretation as latent variables.

What we'll find is that attention and discrete latent variables have a lot in common — they both predict discrete distributions over sets of objects — and so we can use what we know about graphical models to enhance attention

A Latent View of Attention

[Kim et al. 17]



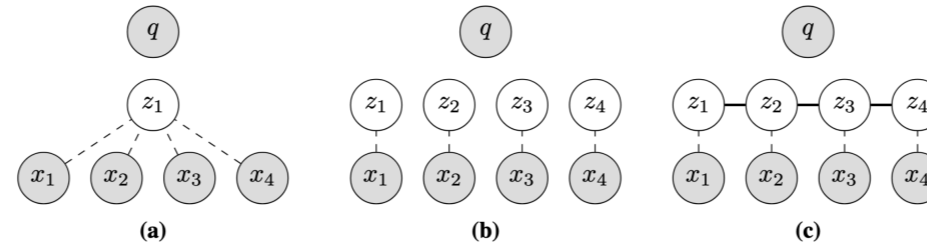
Kim et al extend the typical categorical attention mechanism to use marginal probabilities of structured distributions as the attention scores.

* This allows for neighboring attentions to be correlated, or, when using dependency syntax CRF (not shown), for word representations to be influenced by their most likely syntactic dependency parents in the sentence

A Latent View of Attention

[Kim et al. 17]

$$p(z_1 = i) = \frac{\exp\{v_q^\top W h_{x_i}\}}{\sum_{j=1}^N \exp\{v_q^\top W h_{x_j}\}}$$



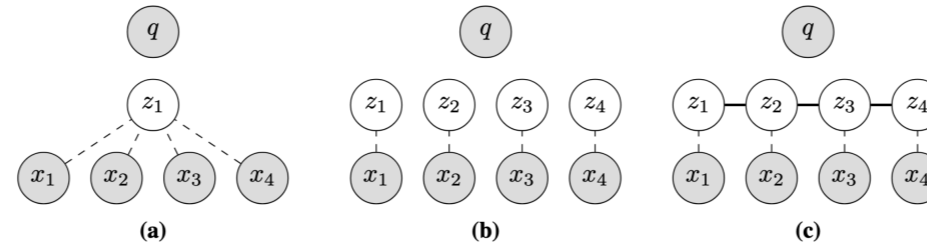
Kim et al extend the typical categorical attention mechanism to use marginal probabilities of structured distributions as the attention scores.

* This allows for neighboring attentions to be correlated, or, when using dependency syntax CRF (not shown), for word representations to be influenced by their most likely syntactic dependency parents in the sentence

A Latent View of Attention

[Kim et al. 17]

$$p(z_1 = i) = \frac{\exp\{v_q^\top W h_{x_i}\}}{\sum_{j=1}^N \exp\{v_q^\top W h_{x_j}\}} \quad c = \mathbb{E}_{p_\theta(z|x,q)}[h_x]$$



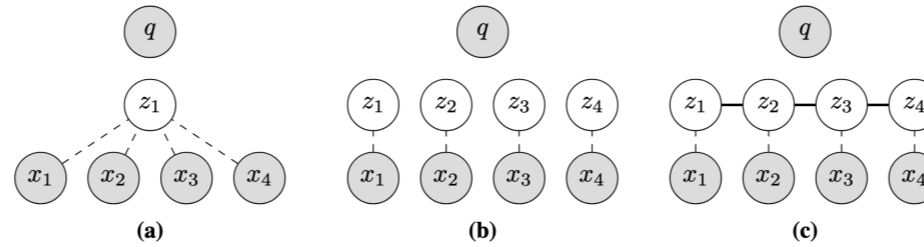
Kim et al extend the typical categorical attention mechanism to use marginal probabilities of structured distributions as the attention scores.

* This allows for neighboring attentions to be correlated, or, when using dependency syntax CRF (not shown), for word representations to be influenced by their most likely syntactic dependency parents in the sentence

A Latent View of Attention

[Kim et al. 17]

$$p(z_i = i) = \frac{\exp\{v_q^\top W h_{x_i}\}}{\sum_{j=1}^N \exp\{v_q^\top W h_{x_j}\}} \quad c = \mathbb{E}_{p_\theta(z|x,q)}[h_x]$$



$$p(z_i = 1) = \sigma(v_q^\top W h_{x_i} + b)$$

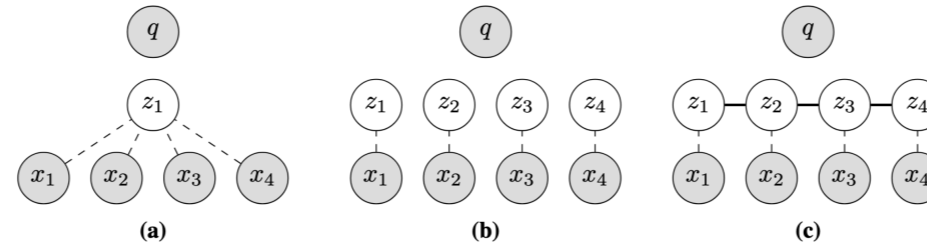
Kim et al extend the typical categorical attention mechanism to use marginal probabilities of structured distributions as the attention scores.

* This allows for neighboring attentions to be correlated, or, when using dependency syntax CRF (not shown), for word representations to be influenced by their most likely syntactic dependency parents in the sentence

A Latent View of Attention

[Kim et al. 17]

$$p(z_1 = i) = \frac{\exp\{v_q^\top W h_{x_i}\}}{\sum_{j=1}^N \exp\{v_q^\top W h_{x_j}\}} \quad c = \mathbb{E}_{p_\theta(z|x,q)}[h_x]$$



$$p(z_i = 1) = \sigma(v_q^\top W h_{x_i} + b)$$

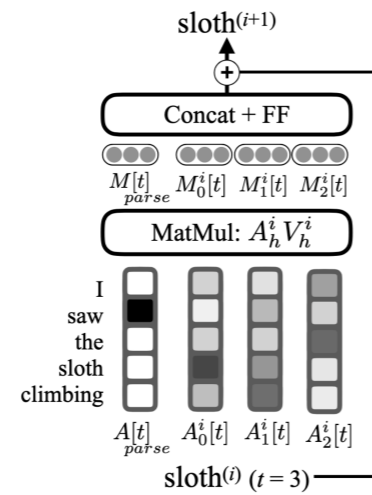
$$p(z_i = 1) \propto \exp\{v_q^\top W h_{x_i} + \log \alpha_{i-1,1} + \log \beta_{i+1,1}\}$$

Kim et al extend the typical categorical attention mechanism to use marginal probabilities of structured distributions as the attention scores.

* This allows for neighboring attentions to be correlated, or, when using dependency syntax CRF (not shown), for word representations to be influenced by their most likely syntactic dependency parents in the sentence

A Latent View of Attention

[Strubell et al. 18]
$$p(x_i = j) = \frac{\exp\{h_{x_i}^\top A h_{x_j}\}}{\sum_{k=1}^N \exp\{h_{x_i}^\top A h_{x_k}\}}$$

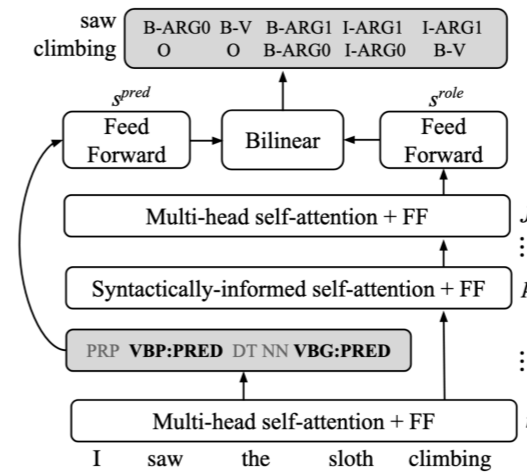
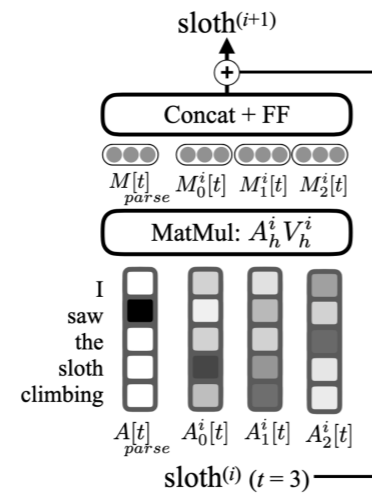


Similarly, Strubell et al, which was the best paper at the most recent EMNLP.

- * Like Kim's tree-based attention, they identified a correspondence between self-attention and dependency-syntax.
- * They then directly supervised one of the attention heads in a transformer architecture to attend to syntactic parents, which provided huge improvements in representation learning for semantic role-labeling.

A Latent View of Attention

[Strubell et al. 18]
$$p(x_i = j) = \frac{\exp\{h_{x_i}^\top A h_{x_j}\}}{\sum_{k=1}^N \exp\{h_{x_i}^\top A h_{x_k}\}}$$



Similarly, Strubell et al, which was the best paper at the most recent EMNLP.

- * Like Kim's tree-based attention, they identified a correspondence between self-attention and dependency-syntax.
- * They then directly supervised one of the attention heads in a transformer architecture to attend to syntactic parents, which provided huge improvements in representation learning for semantic role-labeling.

A Latent View of Attention

[Lei et al. 16]

$$p_{\theta}(y, z_{1:N} | x_{1:N}) = p_{\theta}(y | x_z) p_{\theta}(z_{1:N} | x_{1:N})$$

Review

the beer was n't what i expected, and i'm not sure it's "true to style", but i thought it was delicious. **a very pleasant ruby red-amber color** with a relatively brilliant finish, but a limited amount of carbonation, from the look of it. aroma is what i think an amber ale should be - a nice blend of caramel and happiness bound together.

Ratings

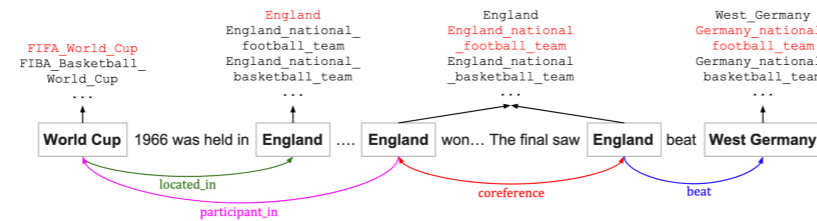
Look: 5 stars

Smell: 4 stars

Lei et al. embedding a stochastic binary attention as a bottleneck in document aspect classification which forced the model to focus on important signal only, but they don't treat it as a formal latent variable in VAE sense

A Latent View of Attention

[Le & Titov 18]



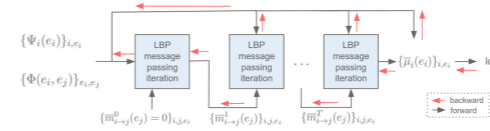
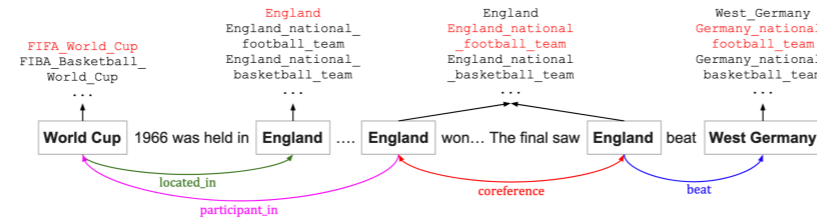
Le and Titov extend the approximate CRF entity linking model of Ganea and Hofmann by encoding potential relations between entity mentions as latent variables in the model.

* Instead of giving all pairs equal weight, this mechanism effectively weighs certain pairs more highly to favor the influence of pairs which are likely related in the text to the final disambiguation score

A Latent View of Attention

[Le & Titov 18]

[Ganea & Hofmann 17]



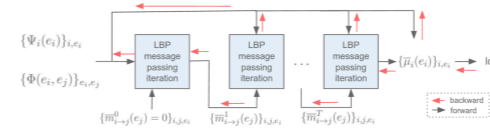
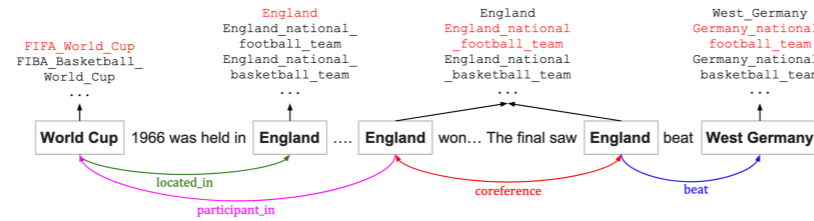
Le and Titov extend the approximate CRF entity linking model of Ganea and Hofmann by encoding potential relations between entity mentions as latent variables in the model.

* Instead of giving all pairs equal weight, this mechanism effectively weighs certain pairs more highly to favor the influence of pairs which are likely related in the text to the final disambiguation score

A Latent View of Attention

[Le & Titov 18]

[Ganea & Hofmann 17]



$$p_{\theta}(e_1, \dots, e_M | x) = \exp\left\{ \sum_{i=1}^M [\Psi_{\theta}(e_i) + \sum_{j<i} \Phi_{\theta}(e_i, e_j)] \right\} / Z(\theta, x)$$

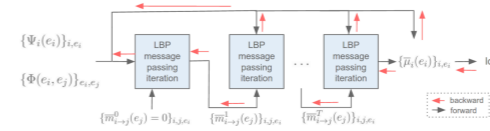
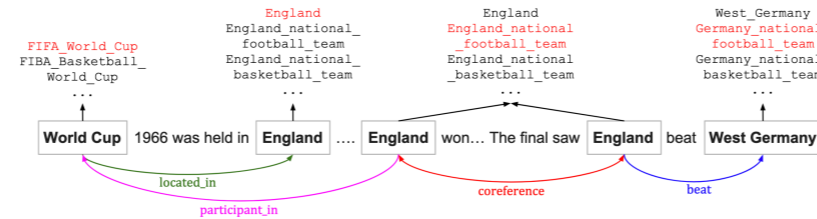
Le and Titov extend the approximate CRF entity linking model of Ganea and Hofmann by encoding potential relations between entity mentions as latent variables in the model.

* Instead of giving all pairs equal weight, this mechanism effectively weighs certain pairs more highly to favor the influence of pairs which are likely related in the text to the final disambiguation score

A Latent View of Attention

[Le & Titov 18]

[Ganea & Hofmann 17]



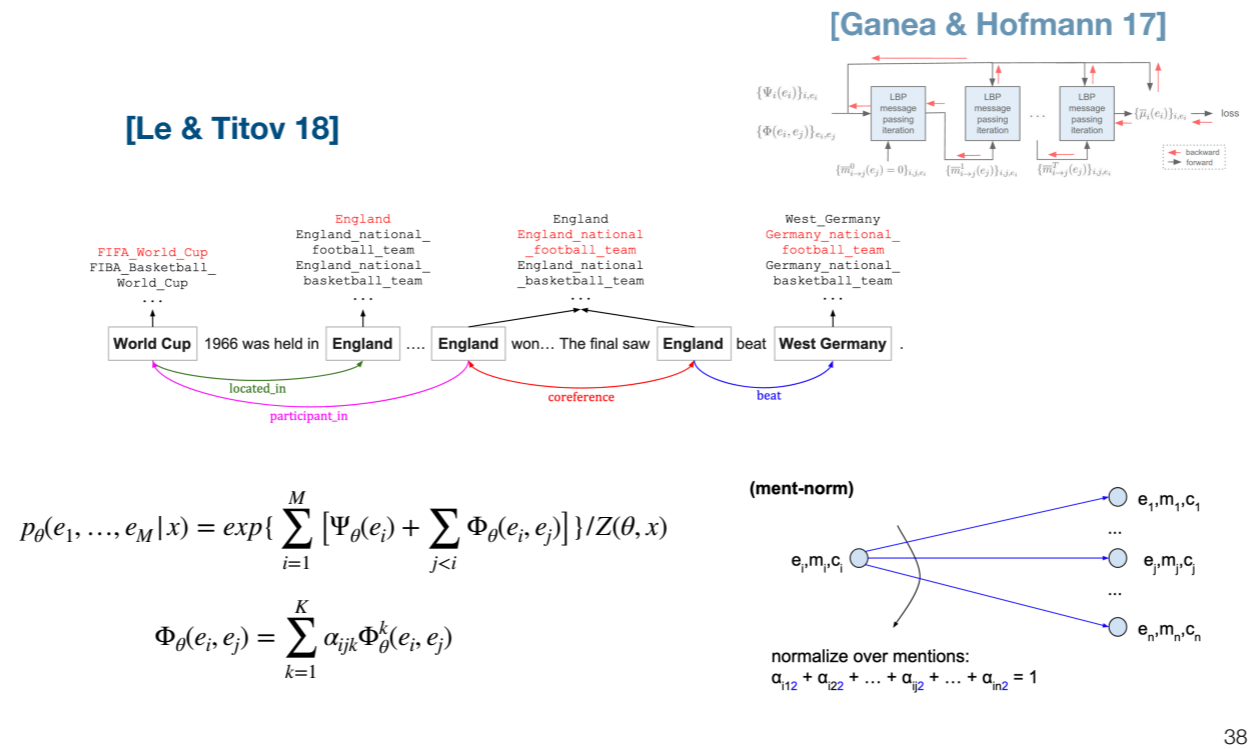
$$p_{\theta}(e_1, \dots, e_M | x) = \exp\left\{ \sum_{i=1}^M [\Psi_{\theta}(e_i) + \sum_{j<i} \Phi_{\theta}(e_i, e_j)] \right\} / Z(\theta, x)$$

$$\Phi_{\theta}(e_i, e_j) = \sum_{k=1}^K \alpha_{ijk} \Phi_{\theta}^k(e_i, e_j)$$

Le and Titov extend the approximate CRF entity linking model of Ganea and Hofmann by encoding potential relations between entity mentions as latent variables in the model.

* Instead of giving all pairs equal weight, this mechanism effectively weighs certain pairs more highly to favor the influence of pairs which are likely related in the text to the final disambiguation score

A Latent View of Attention



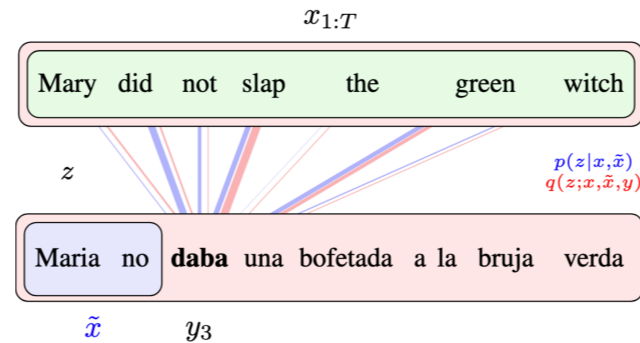
Le and Titov extend the approximate CRF entity linking model of Ganea and Hofmann by encoding potential relations between entity mentions as latent variables in the model.

* Instead of giving all pairs equal weight, this mechanism effectively weighs certain pairs more highly to favor the influence of pairs which are likely related in the text to the final disambiguation score

A Latent View of Attention

[Deng et al. 18]

$$p_{\theta}(y_i, z_i | \tilde{x}, x_{1:N}) = p_{\theta}(y_i | z_i, \tilde{x}, x) p_{\theta}(z_i | \tilde{x}, x_{1:N})$$



39

Lastly, Deng et al model attention in machine translation formally as a latent variable in a VAE.

Viewing attention this way allows them to use an inference network that could consider the entire output when providing attention samples, in particular it can look at the word **to be generated** when computing attention.

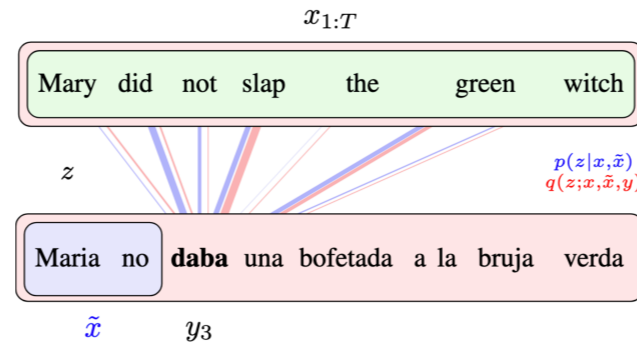
* This provides more signal to the model samples and the KL term in the vae drives the forward attention towards this approximate posterior attention

* An interesting byproduct of this approach is that they also get a posterior inference model which can take a translation pair and infer alignments between them.

A Latent View of Attention

[Deng et al. 18]

$$p_{\theta}(y_i, z_i | \tilde{x}, x_{1:N}) = p_{\theta}(y_i | z_i, \tilde{x}, x) p_{\theta}(z_i | \tilde{x}, x_{1:N})$$



$$\log p_{\theta}(y_i | \tilde{x}, x_{1:N}) \geq \mathbb{E}_{q_{\phi}(z_i | y_i, \tilde{x}, x_{1:N})} [p_{\theta}(y_i | z_i, \tilde{x}, x)] - KL(q_{\phi}(z_i) || p_{\theta}(z_i))$$

Lastly, Deng et al model attention in machine translation formally as a latent variable in a VAE.

Viewing attention this way allows them to use an inference network that could consider the entire output when providing attention samples, in particular it can look at the word **to be generated** when computing attention.

- * This provides more signal to the model samples and the KL term in the vae drives the forward attention towards this approximate posterior attention
- * An interesting byproduct of this approach is that they also get a posterior inference model which can take a translation pair and infer alignments between them.

A Latent View of Attention (cont'd)

Paper, Task	Contributions	Limitations
[Kim et al. 17] machine translation, question answering, & natural language inference	Correlated attentions from tractable CRFs	Attentions are restricted to product of marginals , not joint distributions
[Strubell et al. 18] semantic role labeling	Supervise the attentions with separate labeled data, improving performance	Not guaranteed to produce proper dependency trees , missing structural constraints

40

These mechanisms also have much in common with graphical models — they induce unsupervised distributions over sets of objects. This is important because attention mechanisms have become a mainstay in neural architectures for NLP, in part because they improve performance, in part because they provide some level of interpretability.

A Latent View of Attention

Paper, Task	Contributions	Limitations
[Lei et al. 16] document aspect classification	Use hard binary attention as bottleneck for classification	Requires many rational samples per update for convergence
[Le & Titov 18] entity linking	Model latent relations between entities in joint disambiguation as an attention improves performance	Latent relations are unsupervised and therefore not grounded to known relations in the KG
[Deng et al. 18] machine translation	Attention as approximate posterior to condition on extra output during training	Difficult to optimize the model successfully

These papers illustrate that attention mechanisms can benefit greatly from latent-variable approaches, whether it by structured inference, cross-entropy supervision, stochasticity, reweighing the scoring factors, or using posterior inference to improve forward attention.

Conclusions

- Deep learning + graphical models: **mutually beneficial**
- **Wide applicability** in NLP
- **Semi-supervised learning** by embedding classification/structured prediction **as inference in generative model**
- **Attention** can be **improved by** ideas from **structured/latent variable models**

42

To wrap up, I hope I've convinced you that deep learning, generative models, and structured models all have a lot to offer each other in the field of NLP.

I particularly think the semi-supervised learning for neural structured outputs is an exciting direction for the field!

Thanks!

Questions?

Backup Slides

Mitigating Posterior Collapse

45

One of the first papers to successfully deal with this problem is Yang et al.

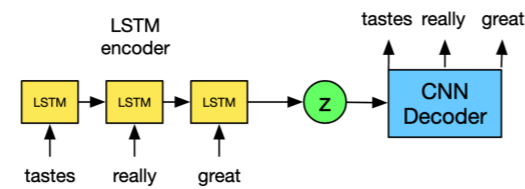
- * They propose to solve it by limiting the dependency structure generative model using dilated convolutions, preventing the model from seeing too much history
- * They find that while this helps and are first to get LL above RNNLM, but there is a clear tradeoff in decoder capacity and use of z

Xu and Durrett take a different tack:

- * They address the problem by switching the latent var distribution to a Fisher vonMises dist, which puts mass on the unit hyper-sphere
- * By doing this, the KL term no longer depends on μ and effectively becomes a tunable hyperparameter of the model, allowing for tuning the balance of contribution between reconstruction and KL to LL of sentences
- * This works quite well in practice, but now we've introduced another hyperparameter, κ , that needs tuning

Mitigating Posterior Collapse

[Yang et al. 17]



45

One of the first papers to successfully deal with this problem is Yang et al.

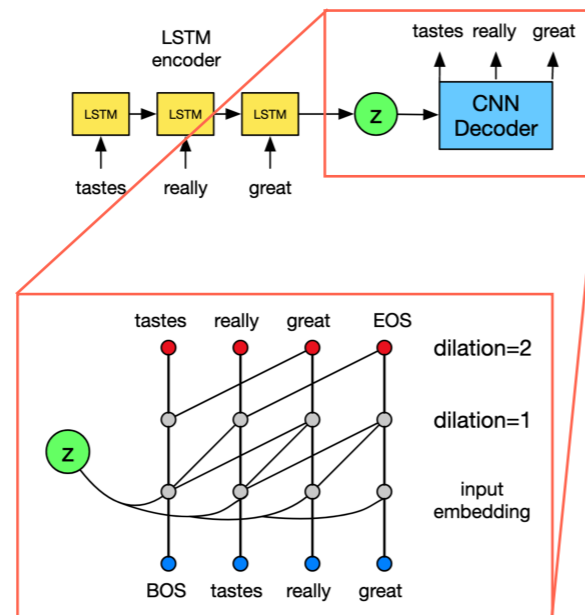
- * They propose to solve it by limiting the dependency structure generative model using dilated convolutions, preventing the model from seeing too much history
- * They find that while this helps and are first to get LL above RNNLM, but there is a clear tradeoff in decoder capacity and use of z

Xu and Durrett take a different tack:

- * They address the problem by switching the latent var distribution to a Fisher vonMises dist, which puts mass on the unit hyper-sphere
- * By doing this, the KL term no longer depends on mu and effectively becomes a tunable hyperparameter of the model, allowing for tuning the balance of contribution between reconstruction and KL to LL of sentences
- * This works quite well in practice, but now we've introduced another hyperparameter, kappa, that needs tuning

Mitigating Posterior Collapse

[Yang et al. 17]



45

One of the first papers to successfully deal with this problem is Yang et al.

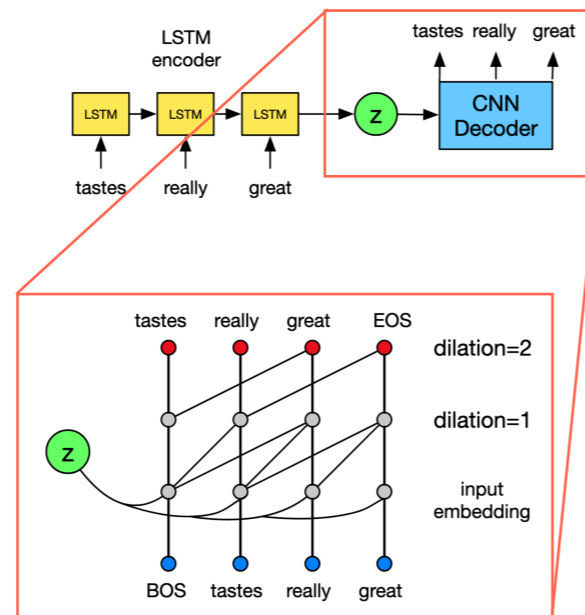
- * They propose to solve it by limiting the dependency structure generative model using dilated convolutions, preventing the model from seeing too much history
- * They find that while this helps and are first to get LL above RNNLM, but there is a clear tradeoff in decoder capacity and use of z

Xu and Durrett take a different tack:

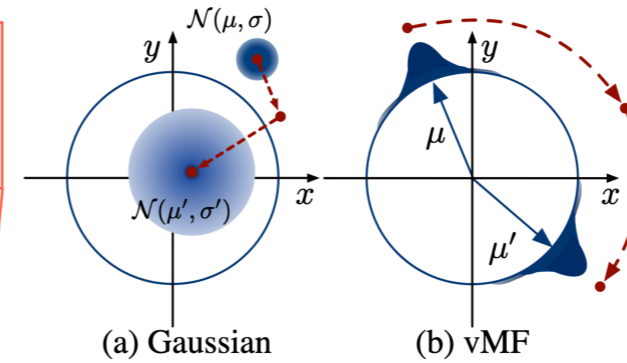
- * They address the problem by switching the latent var distribution to a Fisher vonMises dist, which puts mass on the unit hyper-sphere
- * By doing this, the KL term no longer depends on mu and effectively becomes a tunable hyperparameter of the model, allowing for tuning the balance of contribution between reconstruction and KL to LL of sentences
- * This works quite well in practice, but now we've introduced another hyperparameter, kappa, that needs tuning

Mitigating Posterior Collapse

[Yang et al. 17]



[Xu & Durrett 18]



45

One of the first papers to successfully deal with this problem is Yang et al.

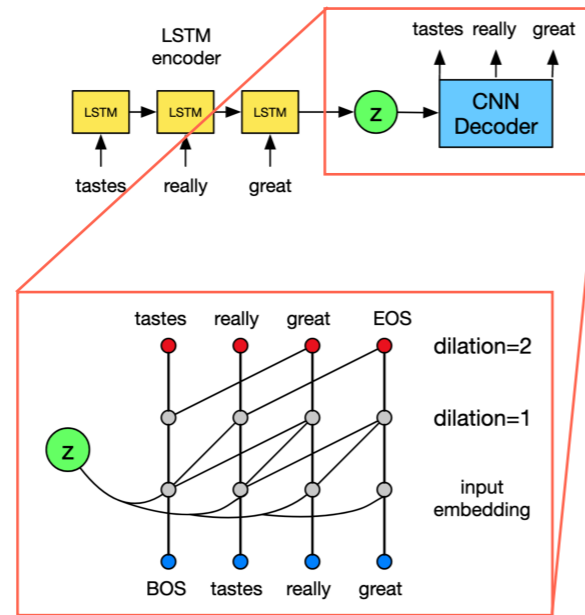
- * They propose to solve it by limiting the dependency structure generative model using dilated convolutions, preventing the model from seeing too much history
- * They find that while this helps and are first to get LL above RNNLM, but there is a clear tradeoff in decoder capacity and use of z

Xu and Durrett take a different tack:

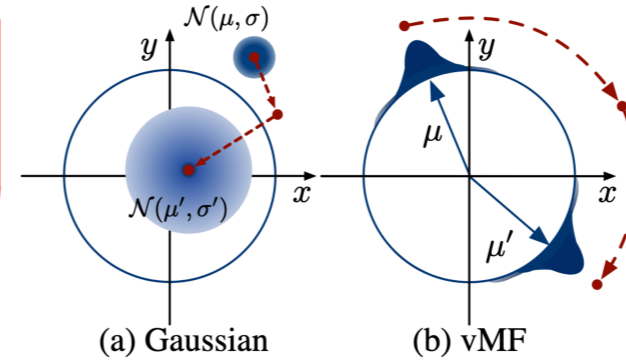
- * They address the problem by switching the latent var distribution to a Fisher vonMises dist, which puts mass on the unit hyper-sphere
- * By doing this, the KL term no longer depends on μ and effectively becomes a tunable hyperparameter of the model, allowing for tuning the balance of contribution between reconstruction and KL to LL of sentences
- * This works quite well in practice, but now we've introduced another hyperparameter, κ , that needs tuning

Mitigating Posterior Collapse

[Yang et al. 17]



[Xu & Durrett 18]



$$L(\theta, \phi; x) = \mathbb{E}_{q(\alpha)}[\log p_{\theta}(x | z_{\phi}(\alpha; \kappa))] - KL(q_{\kappa}(z_{\phi}(\alpha)) || p(\alpha)),$$

45

One of the first papers to successfully deal with this problem is Yang et al.

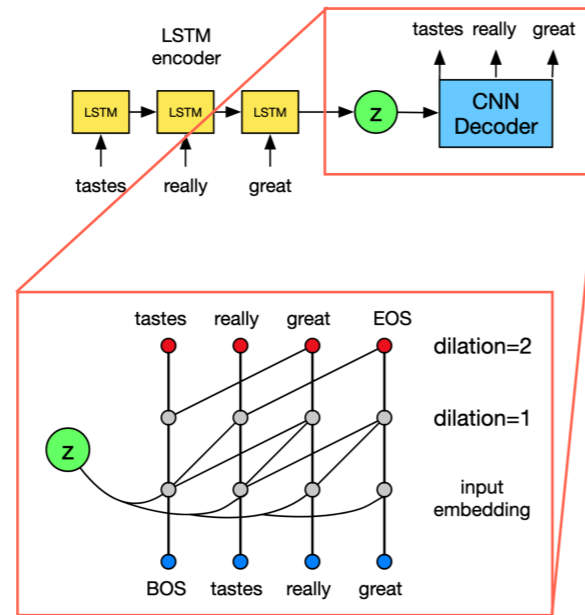
- * They propose to solve it by limiting the dependency structure generative model using dilated convolutions, preventing the model from seeing too much history
- * They find that while this helps and are first to get LL above RNNLM, but there is a clear tradeoff in decoder capacity and use of z

Xu and Durrett take a different tack:

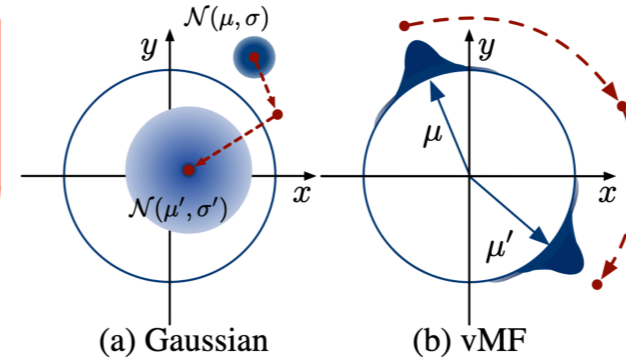
- * They address the problem by switching the latent var distribution to a Fisher vonMises dist, which puts mass on the unit hyper-sphere
- * By doing this, the KL term no longer depends on μ and effectively becomes a tunable hyperparameter of the model, allowing for tuning the balance of contribution between reconstruction and KL to LL of sentences
- * This works quite well in practice, but now we've introduced another hyperparameter, κ , that needs tuning

Mitigating Posterior Collapse

[Yang et al. 17]



[Xu & Durrett 18]



$$L(\theta, \phi; x) = \mathbb{E}_{q(\alpha)}[\log p_{\theta}(x | z_{\phi}(\alpha; \kappa))] - KL(q_{\kappa}(z_{\phi}(\alpha)) || p(\alpha)),$$

$$KL(q_{\kappa}(z_{\phi}(\alpha)) || p(\alpha)) \perp \phi$$

45

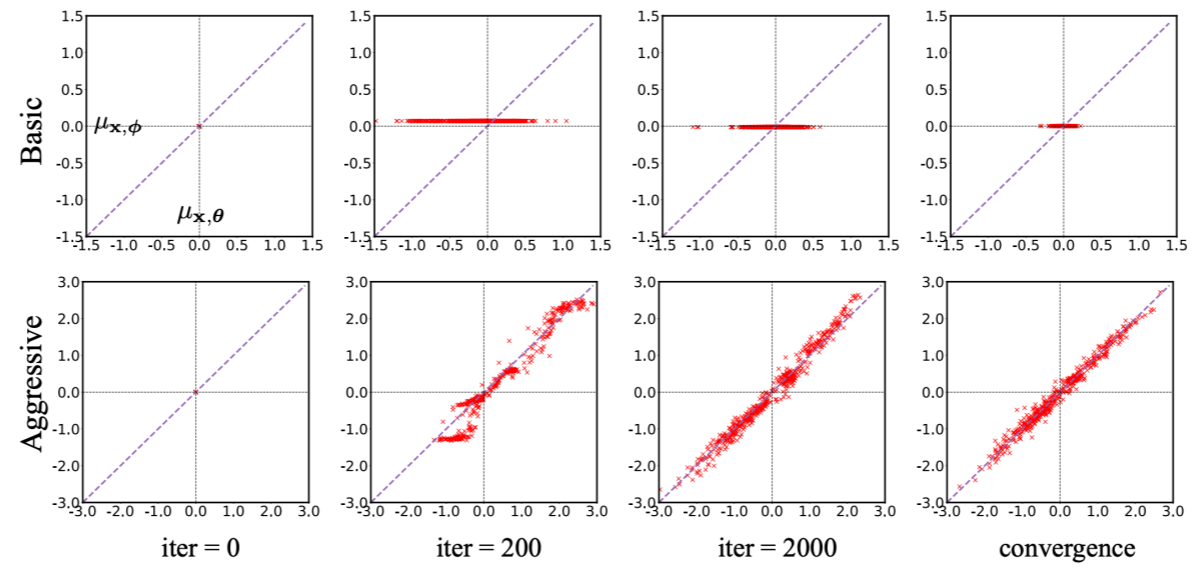
One of the first papers to successfully deal with this problem is Yang et al.

- * They propose to solve it by limiting the dependency structure generative model using dilated convolutions, preventing the model from seeing too much history
- * They find that while this helps and are first to get LL above RNNLM, but there is a clear tradeoff in decoder capacity and use of z

Xu and Durrett take a different tack:

- * They address the problem by switching the latent var distribution to a Fisher vonMises dist, which puts mass on the unit hyper-sphere
- * By doing this, the KL term no longer depends on mu and effectively becomes a tunable hyperparameter of the model, allowing for tuning the balance of contribution between reconstruction and KL to LL of sentences
- * This works quite well in practice, but now we've introduced another hyperparameter, kappa, that needs tuning

Lagging Inference Networks



Lagging Inference Networks

Algorithm 1 VAE training with controlled aggressive inference network optimization.

```
1:  $\theta, \phi \leftarrow$  Initialize parameters
2:  $aggressive \leftarrow$  TRUE
3: repeat
4:   if  $aggressive$  then
5:     repeat  $\triangleright$  [aggressive updates]
6:        $\mathbf{X} \leftarrow$  Random data minibatch
7:       Compute gradients  $\mathbf{g}_\phi \leftarrow \nabla_\phi \mathcal{L}(\mathbf{X}; \theta, \phi)$ 
8:       Update  $\phi$  using gradients  $\mathbf{g}_\phi$ 
9:     until convergence
10:     $\mathbf{X} \leftarrow$  Random data minibatch
11:    Compute gradients  $\mathbf{g}_\theta \leftarrow \nabla_\theta \mathcal{L}(\mathbf{X}; \theta, \phi)$ 
12:    Update  $\theta$  using gradients  $\mathbf{g}_\theta$ 
13:  else  $\triangleright$  [basic VAE training]
14:     $\mathbf{X} \leftarrow$  Random data minibatch
15:    Compute gradients  $\mathbf{g}_{\theta, \phi} \leftarrow \nabla_{\phi, \theta} \mathcal{L}(\mathbf{X}; \theta, \phi)$ 
16:    Update  $\theta, \phi$  using  $\mathbf{g}_{\theta, \phi}$ 
17:  end if
18:  Update  $aggressive$  as discussed in Section 4.2
19: until convergence
```

$$I_q = \mathbb{E}_{\mathbf{x} \sim p_d(\mathbf{x})} [D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))] - D_{\text{KL}}(q_\phi(\mathbf{z})||p(\mathbf{z})),$$

Topic Models

48

A quick recap of standard LDA topic models.

We're given a hyperparameter α that governs topic sparsity and a set K of topic word distributions as parameters

- * Then for each document, we draw its topic distribution (a simplex vector) from a dirichlet prior
- * Then for each word in the doc, we draw a discrete topic choice z_n
- * Then we draw a word from the topic selected by z_n

The marginal probability of the data is given below. Note that we can marginalize out the discrete topic choices easily, yielding what's called the "collapsed" model. Now we only need to do inference on the topic proportions θ .

Topic Models

input: topic sparsity prior α

48

A quick recap of standard LDA topic models.

We're given a hyperparameter α that governs topic sparsity and a set K of topic word distributions as parameters

- * Then for each document, we draw its topic distribution (a simplex vector) from a dirichlet prior
- * Then for each word in the doc, we draw a discrete topic choice z_n
- * Then we draw a word from the topic selected by z_n

The marginal probability of the data is given below. Note that we can marginalize out the discrete topic choices easily, yielding what's called the "collapsed" model. Now we only need to do inference on the topic proportions θ .

Topic Models

input: topic sparsity prior α
word distributions

48

A quick recap of standard LDA topic models.

We're given a hyperparameter alpha that governs topic sparsity and a set K of topic word distributions as parameters

- * Then for each document, we draw its topic distribution (a simplex vector) from a dirichlet prior
- * Then for each word in the doc, we draw a discrete topic choice z_n
- * Then we draw a word from the topic selected by z_n

The marginal probability of the data is given below. Note that we can marginalize out the discrete topic choices easily, yielding what's called the "collapsed" model. Now we only need to do inference on the topic proportions theta.

Topic Models

input: topic sparsity prior α
word distributions
for each document D

$$\theta \sim \text{Dirichlet}(\alpha)$$

48

A quick recap of standard LDA topic models.

We're given a hyperparameter alpha that governs topic sparsity and a set K of topic word distributions as parameters

- * Then for each document, we draw its topic distribution (a simplex vector) from a dirichlet prior
- * Then for each word in the doc, we draw a discrete topic choice z_n
- * Then we draw a word from the topic selected by z_n

The marginal probability of the data is given below. Note that we can marginalize out the discrete topic choices easily, yielding what's called the "collapsed" model. Now we only need to do inference on the topic proportions theta.

Topic Models

input: topic sparsity prior α
word distributions
for each document D
Draw topic distribution $\theta \sim \text{Dirichlet}(\alpha)$

48

A quick recap of standard LDA topic models.

We're given a hyperparameter alpha that governs topic sparsity and a set K of topic word distributions as parameters

- * Then for each document, we draw its topic distribution (a simplex vector) from a dirichlet prior
- * Then for each word in the doc, we draw a discrete topic choice z_n
- * Then we draw a word from the topic selected by z_n

The marginal probability of the data is given below. Note that we can marginalize out the discrete topic choices easily, yielding what's called the "collapsed" model. Now we only need to do inference on the topic proportions theta.

Topic Models

```
input: topic sparsity prior  $\alpha$   
        word distributions  
for each document D  
    Draw topic distribution  $\theta \sim \text{Dirichlet}(\alpha)$   
    for each word in D  
         $z_n \sim \text{Categorical}(\theta)$ 
```

48

A quick recap of standard LDA topic models.

We're given a hyperparameter alpha that governs topic sparsity and a set K of topic word distributions as parameters

- * Then for each document, we draw its topic distribution (a simplex vector) from a dirichlet prior
- * Then for each word in the doc, we draw a discrete topic choice z_n
- * Then we draw a word from the topic selected by z_n

The marginal probability of the data is given below. Note that we can marginalize out the discrete topic choices easily, yielding what's called the "collapsed" model. Now we only need to do inference on the topic proportions theta.

Topic Models

```
input: topic sparsity prior  $\alpha$   
word distributions  
for each document D  
  Draw topic distribution  $\theta \sim \text{Dirichlet}(\alpha)$   
  for each word in D  
    Sample topic  $z_n \sim \text{Categorical}(\theta)$ 
```

48

A quick recap of standard LDA topic models.

We're given a hyperparameter alpha that governs topic sparsity and a set K of topic word distributions as parameters

- * Then for each document, we draw its topic distribution (a simplex vector) from a dirichlet prior
- * Then for each word in the doc, we draw a discrete topic choice z_n
- * Then we draw a word from the topic selected by z_n

The marginal probability of the data is given below. Note that we can marginalize out the discrete topic choices easily, yielding what's called the "collapsed" model. Now we only need to do inference on the topic proportions theta.

Topic Models

```
input: topic sparsity prior  $\alpha$   
        word distributions  
for each document D  
    Draw topic distribution  $\theta \sim \text{Dirichlet}(\alpha)$   
    for each word in D  
        Sample topic  $z_n \sim \text{Categorical}(\theta)$   
        Sample word  $w_n \sim \text{Categorical}(\beta_{z_n})$ 
```

48

A quick recap of standard LDA topic models.

We're given a hyperparameter alpha that governs topic sparsity and a set K of topic word distributions as parameters

- * Then for each document, we draw its topic distribution (a simplex vector) from a dirichlet prior
- * Then for each word in the doc, we draw a discrete topic choice z_n
- * Then we draw a word from the topic selected by z_n

The marginal probability of the data is given below. Note that we can marginalize out the discrete topic choices easily, yielding what's called the "collapsed" model. Now we only need to do inference on the topic proportions theta.

Topic Models

input: topic sparsity prior α
word distributions $\beta_{1:K}$
for each document D
Draw topic distribution $\theta \sim \text{Dirichlet}(\alpha)$
for each word in D
Sample topic $z_n \sim \text{Categorical}(\theta)$
Sample word $w_n \sim \text{Categorical}(\beta_{z_n})$

$$\begin{aligned} p(w_{1:N} | \alpha, \beta) &= \int_{\theta} \left(\prod_{n=1}^N \sum_{z_n=1}^k p(w_n | z_n, \beta_{z_n}) p(z_n | \theta) \right) p(\theta | \alpha) d\theta \\ &= \int_{\theta} \left(\prod_{n=1}^N p(w_n | \beta, \theta) \right) p(\theta | \alpha) d\theta \end{aligned}$$

48

A quick recap of standard LDA topic models.

We're given a hyperparameter alpha that governs topic sparsity and a set K of topic word distributions as parameters

- * Then for each document, we draw its topic distribution (a simplex vector) from a dirichlet prior
- * Then for each word in the doc, we draw a discrete topic choice z_n
- * Then we draw a word from the topic selected by z_n

The marginal probability of the data is given below. Note that we can marginalize out the discrete topic choices easily, yielding what's called the "collapsed" model. Now we only need to do inference on the topic proportions theta.

Autoencoding Variational Inference For Topic Models

Dirichlet Laplace Approximation

$$\mu_{1k} = \log \alpha_k - \frac{1}{K} \sum_i \log \alpha_i$$

$$\Sigma_{1kk} = \frac{1}{\alpha_k} \left(1 - \frac{2}{K}\right) + \frac{1}{K^2} \sum_i \frac{1}{\alpha_k}.$$

ELBO

$$L(\Theta) = \sum_{d=1}^D \left[- \left(\frac{1}{2} \left\{ \text{tr}(\Sigma_1^{-1} \Sigma_0) + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \Sigma_1^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) - K + \log \frac{|\Sigma_1|}{|\Sigma_0|} \right\} \right) + \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)} \left[\mathbf{w}_d^\top \log \left(\sigma(\boldsymbol{\beta}) \sigma(\boldsymbol{\mu}_0 + \Sigma_0^{1/2} \boldsymbol{\epsilon}) \right) \right] \right],$$

Autoencoding Variational Inference For Topic Models

# topics	ProdLDA VAE	LDA VAE	LDA DMFVI	LDA Collapsed Gibbs	NVDM
50	0.24	0.11	0.11	0.17	0.08
200	0.19	0.11	0.06	0.14	0.06

Table 1: Average topic coherence on the 20 Newsgroups dataset. Higher is better.

# topics	ProdLDA VAE	LDA VAE	LDA DMFVI	LDA Collapsed Gibbs	NVDM
50	0.14	0.07	-	0.04	0.07
200	0.12	0.05	-	0.06	0.05

Table 2: Average topic coherence on the RCV1 dataset. Higher is better. Results not reported for LDA DMFVI, as inference failed to converge in 24 hours.

# topics	ProdLDA VAE	LDA VAE	LDA DMFVI	LDA Collapsed Gibbs	NVDM
50	1172	1059	1046	728	837
200	1168	1128	1195	688	884

Table 3: Perplexity scores for 20 Newsgroups. Lower is better.

Discovering Discrete Latent Topics with Neural Variational Inference

Algorithm 1 Unbounded Recurrent Neural Topic Model

0: Initialise Θ and Φ ; Set active topic number i

1: **repeat**

2: **for** $s \in$ minibatches S **do**

3: **for** $k \in [1, i]$ **do**

4: Compute topic vector $t_k = \text{RNN}_{\text{Topic}}(t_{k-1})$

5: Compute topic distribution $\beta_k = \text{softmax}(v \cdot t_k^T)$

6: **end for**

7: **for** $d \in D_s$ **do**

8: Sample topic proportion $\hat{\theta} \sim G_{\text{RSB}}(\theta | \mu(d), \sigma^2(d))$

9: **for** $w \in$ document d **do**

10: Compute log-likelihood $\log p(w | \hat{\theta}, \beta)$

11: **end for**

12: Compute lowerbound \mathcal{L}_d^{i-1} and \mathcal{L}_d^i

13: Compute gradients $\nabla_{\Theta, \Phi} \mathcal{L}_d^i$ and update

14: **end for**

15: Compute likelihood increase \mathcal{I}

16: **if** $\mathcal{I} > \gamma$ **then**

17: Increase active topic number $i = i + 1$

18: **end if**

19: **end for**

20: **until** Convergence

$$\mathcal{L}_d^i \approx \sum_{n=1}^N \left[\log p(w_n | \beta^i, \hat{\theta}^i) \right] - D_{KL} [q(x|d) || p(x)]$$

$$\mathcal{I} = \sum_d^D [\mathcal{L}_d^i - \mathcal{L}_d^{i-1}] / \sum_d^D [\mathcal{L}_d^i]$$

Discovering Discrete Latent Topics with Neural Variational Inference

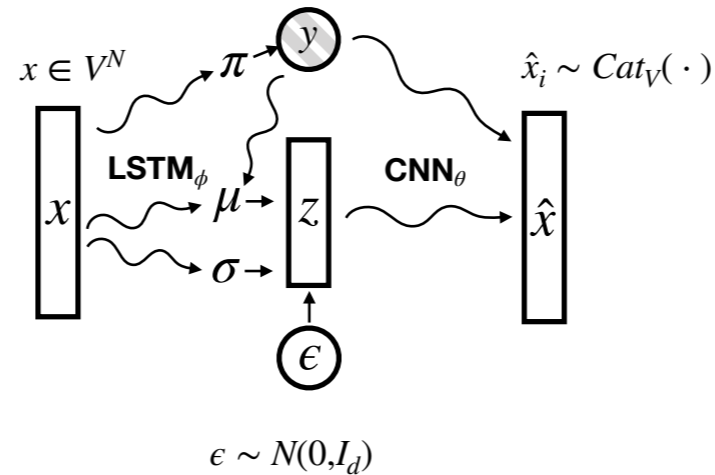
Finite Topic Model	MXM		20News		RCV1	
	50	200	50	200	50	200
GSM	306	272	822	830	717	602
GSB	309	296	838	826	788	634
RSB	311	297	835	822	750	628
OnlineLDA (Hoffman et al., 2010)	312	342	893	1015	1062	1058
NVLDA (Srivastava & Sutton, 2016)	330	357	1073	993	791	797
Unbounded Topic Model	MXM	20News	RCV1			
RSB-TF	303	825	622			
HDP (Wang et al., 2011)	370	937	918			

Finite Document Model	MXM		20News		RCV1	
	50	200	50	200	50	200
GSM	270	267	787	829	653	521
GSB	285	275	816	815	712	544
RSB	286	283	785	792	662	534
NVDM (Miao et al., 2016)	345	345	837	873	717	588
ProdLDA (Srivastava & Sutton, 2016)	319	326	1009	989	780	788
Unbounded Document Model	MXM	20News	RCV1			
RSB-TF	285	788	532			

Classification as Inference in Semi-supervised VAEs

[Yang et al. 17] (again)

$$y \sim \text{Cat}_{\mathcal{Y}}(\theta) \vee \text{Cat}_{\mathcal{Y}}(\pi_{\phi}(x))$$



53

We can visualize this in a VAE architecture by looking back at Yang et al 17 — they also evaluate their dilated CNN decoder on semi-supervised classification

Here now our $q(z)$ depends on y , which may be fixed or sampled.

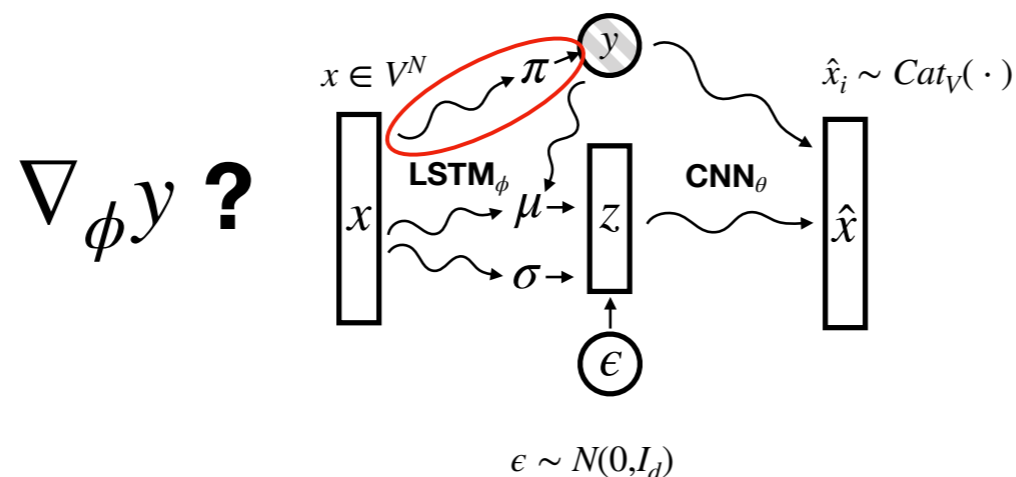
The challenge here is gradients with respect to ϕ through y in the unsupervised case:

- * We cannot naively use the reparameterization trick on discrete variables
- * If y is sufficiently small we can marginalize it out
- * but if marginalizing isn't reasonable, how can we get gradients wrt samples?
- * but Yang et al use a recent development called the Gumbel-Softmax distribution, which reparameterizes a categorical sample using the gumbel-argmax trick, then relaxes the argmax to a softmax, allowing for gradients w.r.t ϕ through $q(y)$ — pretty neat!

Classification as Inference in Semi-supervised VAEs

[Yang et al. 17] (again)

$$y \sim \text{Cat}_{\mathcal{Y}}(\theta) \vee \text{Cat}_{\mathcal{Y}}(\pi_{\phi}(x))$$



53

We can visualize this in a VAE architecture by looking back at Yang et al 17 — they also evaluate their dilated CNN decoder on semi-supervised classification

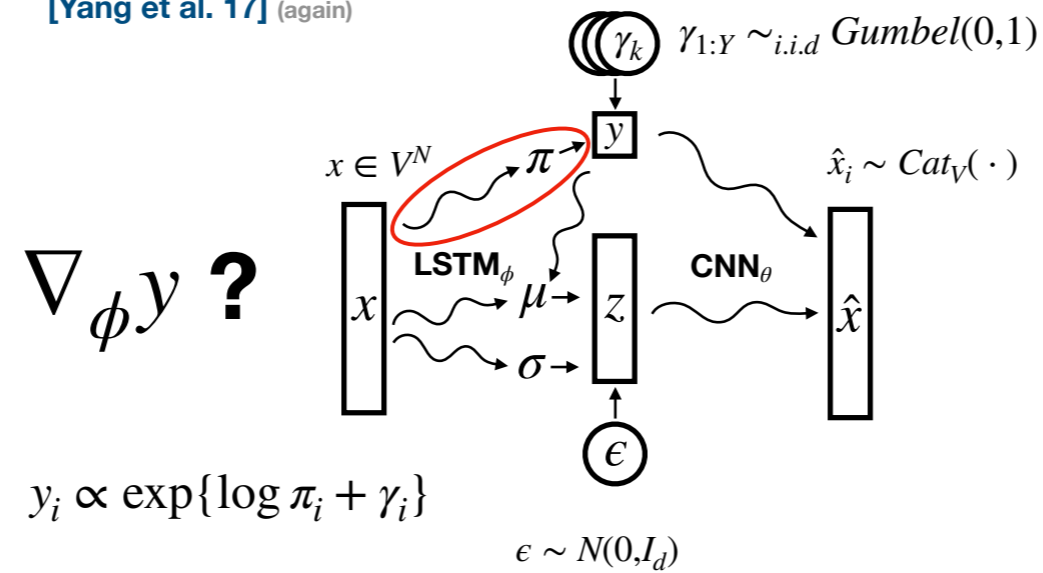
Here now our $q(z)$ depends on y , which may be fixed or sampled.

The challenge here is gradients with respect to ϕ through y in the unsupervised case:

- * We cannot naively use the reparameterization trick on discrete variables
- * If y is sufficiently small we can marginalize it out
- * but if marginalizing isn't reasonable, how can we get gradients wrt samples?
- * but Yang et al use a recent development called the Gumbel-Softmax distribution, which reparameterizes a categorical sample using the gumbel-argmax trick, then relaxes the argmax to a softmax, allowing for gradients w.r.t ϕ through $q(y)$ — pretty neat!

Classification as Inference in Semi-supervised VAEs

[Yang et al. 17] (again)

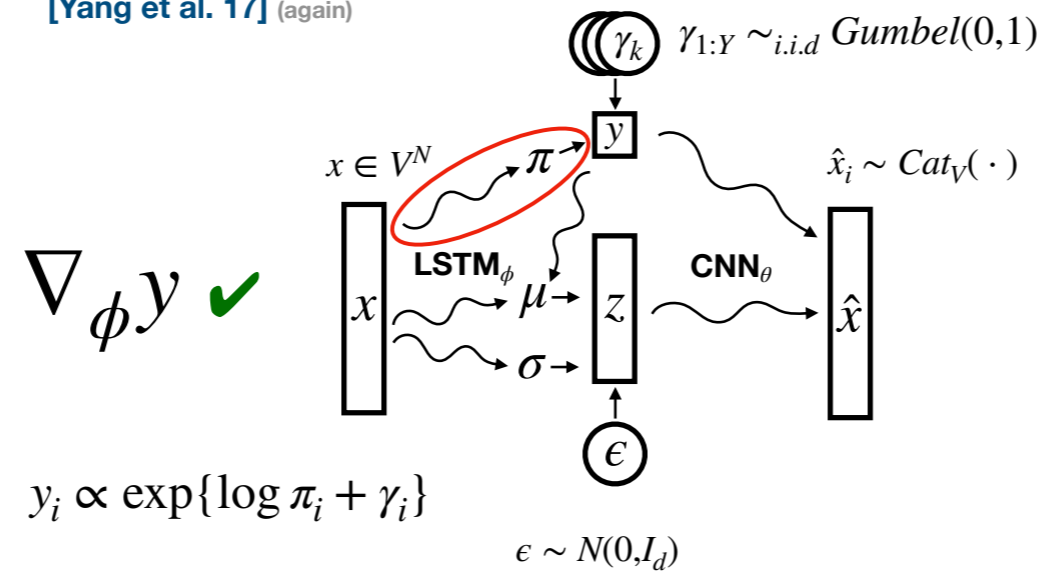


54

- * Yang et al use a recent development called the Gumbel-Softmax distribution, which reparameterizes a categorical sample using the gumbel-argmax trick, then relaxes the argmax to a softmax, allowing for gradients w.r.t phi through q(y) — pretty neat!

Classification as Inference in Semi-supervised VAEs

[Yang et al. 17] (again)

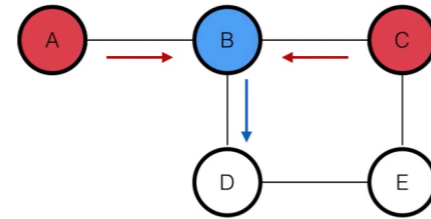


54

- * Yang et al use a recent development called the Gumbel-Softmax distribution, which reparameterizes a categorical sample using the gumbel-argmax trick, then relaxes the argmax to a softmax, allowing for gradients w.r.t phi through q(y) — pretty neat!

Loopy Belief Propagation

Sum-Product Belief Propagation



$$b_t(x_t) \propto \prod_{s \in \text{neighbors}(t)} m_{t \rightarrow s}(x_s)$$

$$m_{s \rightarrow t}(x_t) := \sum_{x_s} \left(\phi_{st}(x_s, x_t) \prod_{u \in \text{neighbors}(s) \setminus t} m_{u \rightarrow s}(x_s) \right)$$

$$m_{B \rightarrow D}(x_D) = \sum_{x_B} \phi(x_B, x_C) \times m_{A \rightarrow B}(x_B) \times m_{C \rightarrow B}(x_B)$$

Credit: Bert-Huang <https://www.youtube.com/watch?v=meBWAb0EWQk>

Globally Normalized Transition Based Neural Networks

Given an input x , most often a sentence, we define:

- A set of states $\mathcal{S}(x)$.
- A special start state $s^\dagger \in \mathcal{S}(x)$.
- A set of allowed decisions $\mathcal{A}(s, x)$ for all $s \in \mathcal{S}(x)$.
- A transition function $t(s, d, x)$ returning a new state s' for any decision $d \in \mathcal{A}(s, x)$.

$$\rho(s, d; \theta) = \phi(s; \theta^{(l)}) \cdot \theta^{(d)}$$

Local Normalization

$$p(d_j | d_{1:j-1}; \theta) = \frac{\exp \rho(d_{1:j-1}, d_j; \theta)}{Z_L(d_{1:j-1}; \theta)}, \quad (1)$$

where

$$Z_L(d_{1:j-1}; \theta) = \sum_{d' \in \mathcal{A}(d_{1:j-1})} \exp \rho(d_{1:j-1}, d'; \theta).$$

Globally Normalized, Early Updates

$$L_{\text{global-beam}}(d_{1:j}^*; \theta) = - \sum_{i=1}^j \rho(d_{1:i-1}^*, d_i^*; \theta) + \ln \sum_{d'_{1:j} \in \mathcal{B}_j} \exp \sum_{i=1}^j \rho(d'_{1:i-1}, d'_i; \theta).$$

Globally Normalized Transition Based Neural Networks

Method	En	En-Union			Ca	Ch	Cz	CoNLL '09				Avg
	WSJ	News	Web	QTB				En	Ge	Ja	Sp	
Linear CRF	97.17	97.60	94.58	96.04	98.81	94.45	98.90	97.50	97.14	97.90	98.79	97.17
Ling et al. (2015)	97.78	97.44	94.03	96.18	98.77	94.38	99.00	97.60	97.84	97.06	98.71	97.16
Our Local (B=1)	97.44	97.66	94.46	96.59	98.91	94.56	98.96	97.36	97.35	98.02	98.88	97.29
Our Local (B=8)	97.45	97.69	94.46	96.64	98.88	94.56	98.96	97.40	97.35	98.02	98.89	97.30
Our Global (B=8)	97.44	97.77	94.80	96.86	99.03	94.72	99.02	97.65	97.52	98.37	98.97	97.47
Parsey McParseface	-	97.52	94.24	96.45	-	-	-	-	-	-	-	-

Table 1: Final POS tagging test set results on English WSJ and Treebank Union as well as CoNLL'09. We also show the performance of our pre-trained open source model. "Parsey McParseface."

Method	WSJ		Union-News		Union-Web		Union-QTB	
	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
Martins et al. (2013)*	92.89	90.55	93.10	91.13	88.23	85.04	94.21	91.54
Zhang and McDonald (2014)*	93.22	91.02	93.32	91.48	88.65	85.59	93.37	90.69
Weiss et al. (2015)	93.99	92.05	93.91	92.25	89.29	86.44	94.17	92.06
Alberti et al. (2015)	94.23	92.36	94.10	92.55	89.55	86.85	94.74	93.04
Our Local (B=1)	92.95	91.02	93.11	91.46	88.42	85.58	92.49	90.38
Our Local (B=32)	93.59	91.70	93.65	92.03	88.96	86.17	93.22	91.17
Our Global (B=32)	94.61	92.79	94.44	92.93	90.17	87.54	95.40	93.64
Parsey McParseface (B=8)	-	-	94.15	92.51	89.08	86.29	94.77	93.17

Table 2: Final English dependency parsing test set results. We note that training our system using only the WSJ corpus (i.e. no pre-trained embeddings or other external resources) yields 94.08% UAS and 92.15% LAS for our global model with beam 32.

Globally Normalized Transition Based Neural Networks

Method	Generated corpus		Human eval	
	A	F1	read	info
Filippova et al. (2015)	35.36	82.83	4.66	4.03
Automatic	-	-	4.31	3.77
Our Local (B=1)	30.51	78.72	4.58	4.03
Our Local (B=8)	31.19	75.69	-	-
Our Global (B=8)	35.16	81.41	4.67	4.07

Label Bias Problem

In proving $\mathcal{P}_G \not\subseteq \mathcal{P}_L$ we will use a simple problem where every example seen in training or test data is one of the following two tagged sentences:

$$\begin{aligned}x_1 x_2 x_3 &= a b c, & d_1 d_2 d_3 &= A B C \\x_1 x_2 x_3 &= a b e, & d_1 d_2 d_3 &= A D E\end{aligned}\quad (7)$$

Note that the input $x_2 = b$ is ambiguous: it can take tags B or D. This ambiguity is resolved when the next input symbol, c or e , is observed.

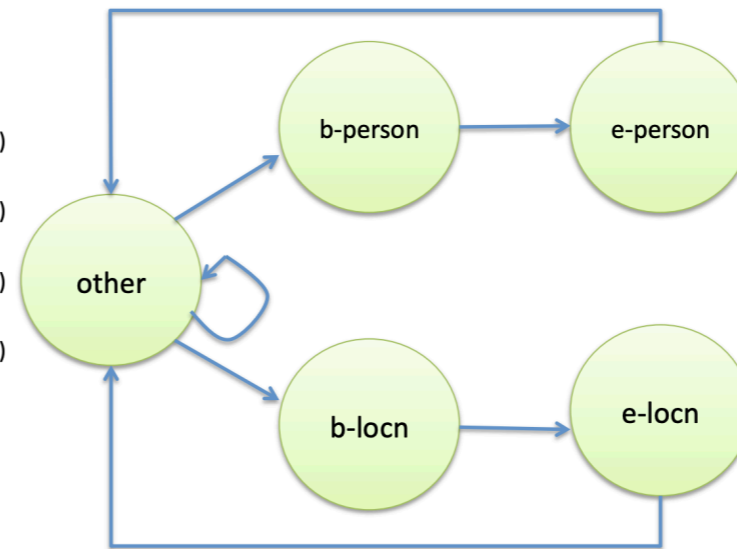
$$p_L(A B C|a b c) + p_L(A D E|a b e) \leq 1$$

$$p_G(A B C|a b c) + p_G(A D E|a b e) > 1$$

Label Bias Problem

corpus:
Harvey Ford
(person 9 times, location 1 time)
Harvey Park
(location 9 times, person 1 time)
Myrtle Ford
(person 9 times, location 1 time)
Myrtle Park
(location 9 times, person 1 time)

*second token a good indicator
of person vs. location*



Credit: <https://cs.nyu.edu/courses/spring17/CSCI-GA.2590-001/LabelBias.pdf>

Label Bias Problem

Conditional probabilities:

$$p(\text{b-person} \mid \text{other}, w = \text{Harvey}) = 0.5$$

$$p(\text{b-locn} \mid \text{other}, w = \text{Harvey}) = 0.5$$

$$p(\text{b-person} \mid \text{other}, w = \text{Myrtle}) = 0.5$$

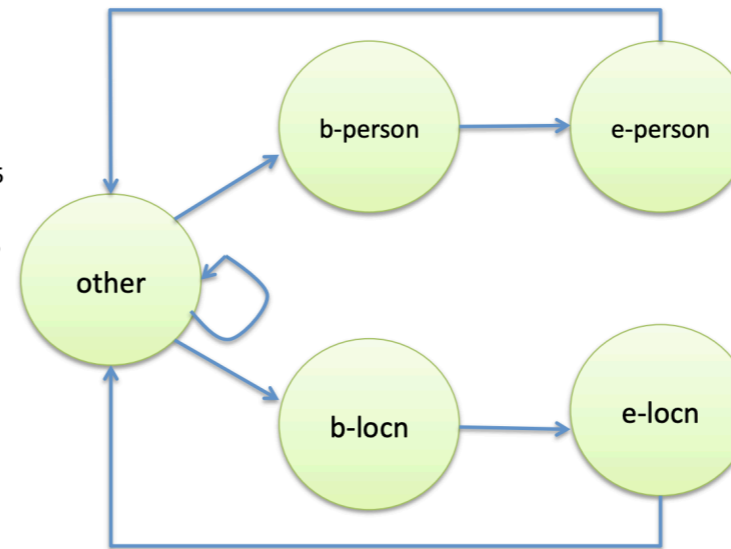
$$p(\text{b-locn} \mid \text{other}, w = \text{Myrtle}) = 0.5$$

$$p(\text{e-person} \mid \text{b-person}, w = \text{Ford}) = 1$$

$$p(\text{e-person} \mid \text{b-person}, w = \text{Park}) = 1$$

$$p(\text{e-locn} \mid \text{b-locn}, w = \text{Ford}) = 1$$

$$p(\text{e-locn} \mid \text{b-locn}, w = \text{Park}) = 1$$



Credit: <https://cs.nyu.edu/courses/spring17/CSCI-GA.2590-001/LabelBias.pdf>

Path, Score Gradients and Control Variates

$$PD : \quad \nabla_{\theta} \mathbb{E}_{p(z; \theta)} [f(z)] = \mathbb{E}_{p(\epsilon)} [\nabla_{\theta} f(g(\epsilon, \theta))]$$

$$SF : \quad \nabla_{\theta} \mathbb{E}_{p(z; \theta)} [f(z)] = \mathbb{E}_{p(z; \theta)} [f(z) \nabla_{\theta} \log p(z; \theta)]$$

Variance Reduction: Control Variates

$$\nabla_{\theta} \mathbb{E}_{p(z; \theta)} [f(z)] = \mathbb{E}_{p(z; \theta)} [(f(z) - \lambda) \nabla_{\theta} \log p(z; \theta)]$$

Language as Latent Variable

Model	Training Data		Recall			Precision			F-1		
	Labelled	Unlabelled	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
FSC	500K	-	30.817	10.861	28.263	22.357	7.998	20.520	23.415	8.156	21.468
ASC+FSC ₁	500K	500K	29.117	10.643	26.811	28.558	10.575	26.344	26.987	9.741	24.874
ASC+FSC ₂	500K	3.8M	28.236	10.359	26.218	30.112	11.131	27.896	27.453	9.902	25.452
FSC	1M	-	30.889	11.645	28.257	27.169	10.266	24.916	26.984	10.028	24.711
ASC+FSC ₁	1M	1M	30.490	11.443	28.097	28.109	10.799	25.943	27.258	10.189	25.148
ASC+FSC ₂	1M	3.8M	29.034	10.780	26.801	31.037	11.521	28.658	28.336	10.313	26.145
FSC	3.8M	-	30.112	12.436	27.889	34.135	13.813	31.704	30.225	12.258	28.035
ASC+FSC ₁	3.8M	3.8M	29.946	12.558	27.805	35.538	14.699	32.972	30.568	12.553	28.366

Table 1: Extractive Summarisation Performance. (1) The extractive summaries of these models are decoded by the pointer network (i.e the shared component of the ASC and FSC models). (2) R-1, R-2 and R-L represent the Rouge-1, Rouge-2 and Rouge-L score respectively.

Model	Training Data		Recall			Precision			F-1		
	Labelled	Unlabelled	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
FSC	500K	-	27.147	10.039	25.197	33.781	13.019	31.288	29.074	10.842	26.955
ASC+FSC ₁	500K	500K	27.067	10.717	25.239	33.893	13.678	31.585	29.027	11.461	27.072
ASC+FSC ₂	500K	3.8M	27.662	11.102	25.703	35.756	14.537	33.212	30.140	12.051	27.99
FSC	1M	-	28.521	11.308	26.478	33.132	13.422	30.741	29.580	11.807	27.439
ASC+FSC ₁	1M	1M	28.333	11.814	26.367	35.860	15.243	33.306	30.569	12.743	28.431
ASC+FSC ₂	1M	3.8M	29.017	12.007	27.067	36.128	14.988	33.626	31.089	12.785	28.967
FSC	3.8M	-	31.148	13.553	28.954	36.917	16.127	34.405	32.327	14.000	30.087
ASC+FSC ₁	3.8M	3.8M	32.385	15.155	30.246	39.224	18.382	36.662	34.156	15.935	31.915

Table 2: Abstractive Summarisation Performance. The abstractive summaries of these models are decoded by the combined pointer network (i.e. the shared pointer network together with the softmax output layer over the full vocabulary).

Semi-Supervised Structured Prediction with Neural CRF Autoencoder

$$\begin{aligned}
 P_{\Theta, \Lambda}(\hat{\mathbf{x}}, \mathbf{y} | \mathbf{x}) &= P_{\Theta}(\hat{\mathbf{x}} | \mathbf{y}) P_{\Lambda}(\mathbf{y} | \mathbf{x}) \\
 &= \left[\prod_t P(\hat{x}_t | y_t) \right] \frac{e^{\Phi(\mathbf{x}, \mathbf{y})}}{Z} \\
 &= \frac{e^{\sum_t s_t(\mathbf{x}, \mathbf{y})}}{Z}, \\
 P_{\Theta, \Lambda}(\hat{\mathbf{x}} | \mathbf{x}) &= \sum_{\mathbf{y}} P(\hat{\mathbf{x}}, \mathbf{y} | \mathbf{x}) \\
 &= \frac{U}{Z},
 \end{aligned}$$

where $U = \sum_{\mathbf{y}} e^{\sum_t s_t(\mathbf{x}, \mathbf{y})}$.

$$s_t(\mathbf{x}, \mathbf{y}) = \log P(x_t | y_t) + \phi(\mathbf{x}, y_t) + \psi(y_{t-1}, y_t).$$

$$\begin{aligned}
 loss_l &= -\log P_{\Theta, \Lambda}(\hat{\mathbf{x}}, \mathbf{y} | \mathbf{x}) \\
 &= -\left(\sum_t s_t(\mathbf{x}, \mathbf{y}) - \log Z \right)
 \end{aligned}$$

Algorithm 2 Mixed Expectation-Maximization

- 1: Initialize expected count table T_e using labeled data $\{\mathbf{x}, \mathbf{y}\}_i^l$ and use it as $\Theta^{(0)}$ in the decoder.
 - 2: Initialize $\Lambda^{(0)}$ in the encoder randomly.
 - 3: **for** t in *epochs* **do**
 - 4: Train the encoder on labeled data $\{\mathbf{x}, \mathbf{y}\}^l$ and unlabeled data $\{\mathbf{x}\}^u$ to update $\Lambda^{(t-1)}$ to $\Lambda^{(t)}$.
 - 5: Re-initialize expected count table T_e with **0**s.
 - 6: Use labeled data $\{\mathbf{x}, \mathbf{y}\}^l$ to calculate real counts and update T_e .
 - 7: Use unlabeled data $\{\mathbf{x}\}^u$ to compute the expected counts with parameters $\Lambda^{(t)}$ and $\Theta^{(t-1)}$ and update T_e .
 - 8: Obtain $\Theta^{(t)}$ globally and analytically based on T_e .
 - 9: **end for**
-

$$\begin{aligned}
 loss_u &= -\log P_{\Theta, \Lambda}(\hat{\mathbf{x}} | \mathbf{x}) \\
 &= -(\log U - \log Z).
 \end{aligned}$$

Semi-Supervised Structured Prediction with Neural CRF Autoencoder

	English	French	German	Italian	Russian	Spanish	Indonesian	Croatian
Tokens	254830	391107	293088	272913	99389	423346	121923	139023
Training	12543	14554	14118	12837	4029	14187	4477	5792
Development	2002	1596	799	489	502	1552	559	200
Testing	2077	298	977	489	499	274	297	297

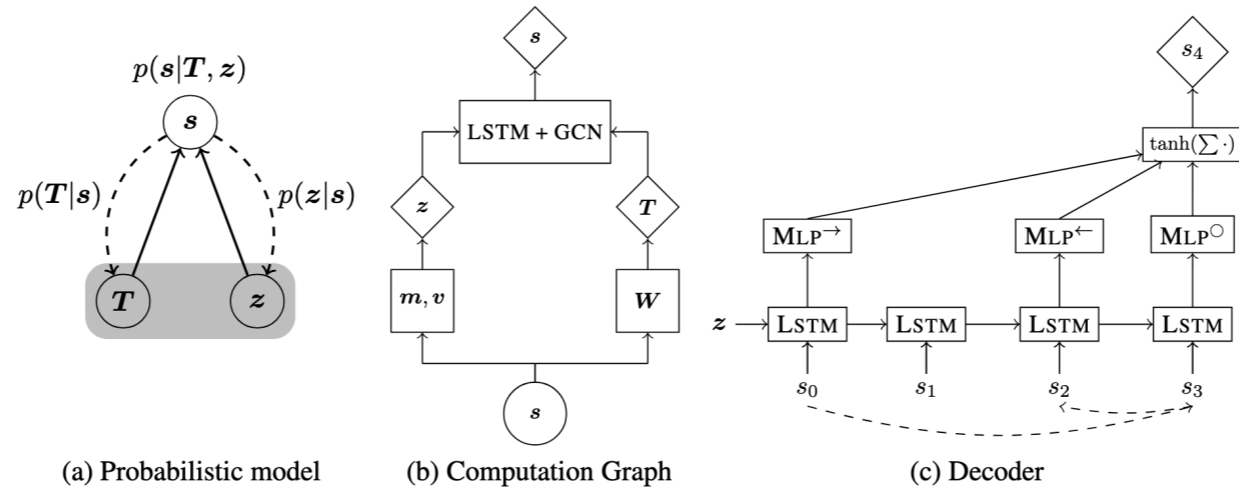
Models	English	French	German	Italian	Russian	Spanish	Indonesian	Croatian
HMM	86.28%	91.23%	85.59%	92.03%	79.82%	91.31%	89.40%	86.98%
CRF	89.96%	93.40%	86.83%	94.07%	83.38%	91.47%	88.63%	86.90%
LSTM	90.50%	94.16%	88.40%	94.96%	84.87%	93.17%	89.42%	88.95%
NCRF	91.52%	95.07%	90.27%	96.20%	93.37%	93.34%	92.32%	93.85%
NCRF-AE	92.50%	95.28%	90.50%	96.64%	93.60%	93.86%	93.96%	94.32%

Table 2: Supervised learning accuracy of POS tagging on 8 UD languages using different models

Models	English	French	German	Italian	Russian	Spanish	Indonesian	Croatian
NCRF _(OL)	88.01%	93.38%	90.43%	91.75%	86.63%	91.22%	88.35%	86.11%
NCRF-AE _(OL)	88.41%	93.69%	90.75%	92.17%	87.82%	91.70%	89.06%	87.92%
HMM-EM	79.92%	88.15%	77.01%	84.57%	72.96%	86.77%	83.61%	77.20%
NCRF-AE _(HEM)	86.79%	92.83%	89.78%	90.68%	86.39%	91.30%	88.86%	86.55%
NCRF-AE	89.43%	93.89%	90.99%	92.85%	88.93%	92.17%	89.41%	89.14%

Table 3: Semi-supervised learning accuracy of POS tagging on 8 UD languages. HEM means hard-EM, used as a self-training approach, and OL means only 20% of the labeled data is used and no unlabeled data is used.

Differentiable Perturb-and-Parse



Differentiable Perturb-and-Parse

$$\begin{aligned}
 \mathbf{W} &= \text{EMBPARAMS}(\mathbf{s}) \\
 \mathbf{P} &\sim \mathcal{G}(0, 1) \\
 \mathbf{T} &= \text{EISNER}(\mathbf{W} + \mathbf{P})
 \end{aligned}
 \quad \mathbb{E}_{q_\phi(\mathbf{T}|\mathbf{s})} [\log p_\theta(\mathbf{s}|\mathbf{T})] \simeq \log p_\theta(\mathbf{s}|\text{EISNER}(\mathbf{W} + \mathbf{P}))$$

Algorithm 1 This function search the best split point for constructing an element given its span. \mathbf{b} is a one-hot vector such that $b_{i-k} = 1$ iff k is the best split position.

```

1: function DEDUCE-URIGHT( $i, j, \mathbf{W}$ )
2:    $\mathbf{s} \leftarrow$  null-initialized vec. of size  $j - i$ 
3:   for  $i \leq k < j$  do
4:      $s_{i-k} \leftarrow [i \sqcup k]$ 
        $+ [k + 1 \triangle j]$ 
        $+ W_{j,i}$ 
5:    $\mathbf{b} \leftarrow$  ONE-HOT-ARGMAX( $\mathbf{s}$ )
6:   BACKPTR[ $i \sqcup j$ ]  $\leftarrow \mathbf{b}$ 
7:   WEIGHT[ $i \sqcup j$ ]  $\leftarrow \mathbf{b}^\top \mathbf{s}$ 

```

Algorithm 2 If item $[i \sqcup j]$ has contributed the optimal objective, this function sets $T_{i,j}$ to 1. Then, it propagates the contribution information to its antecedents.

```

1: function BACKTRACK-URIGHT( $i, j, \mathbf{T}$ )
2:    $T_{i,j} \leftarrow$  CONTRIB[ $i \sqcup j$ ]
3:    $\mathbf{b} \leftarrow$  BACKPTR[ $i \sqcup j$ ]
4:   for  $i \leq k < j$  do
5:     CONTRIB[ $i \sqcup k$ ]  $\leftarrow^+ b_{i-k} T_{i,j}$ 
6:     CONTRIB[ $k + 1 \triangle j$ ]  $\leftarrow^+ b_{i-k} T_{i,j}$ 

```

Differentiable Perturb-and-Parse

(a) Parsing results

	English	French	Swedish
Supervised	88.79 / 84.74	84.09 / 77.58	86.59 / 78.95
VAE w. z	89.39 / 85.44	84.43 / 77.89	86.92 / 80.01
VAE w/o z	89.50 / 85.48	84.69 / 78.49	86.97 / 79.80
Kipperwasser & Goldberg	89.88 / 86.49	84.30 / 77.83	86.93 / 80.12

(b) Dependency length analysis

Distance	Supervised Re / Pr	Semi-sup. Re / Pr
(to root)	93.46 / 89.30	93.84 / 92.41
1	95.61 / 94.07	95.33 / 94.57
2	93.01 / 90.88	92.50 / 92.09
3 ... 6	85.95 / 88.13	87.31 / 87.93
> 7	72.47 / 83.26	78.72 / 83.11

(c) Dependency label analysis

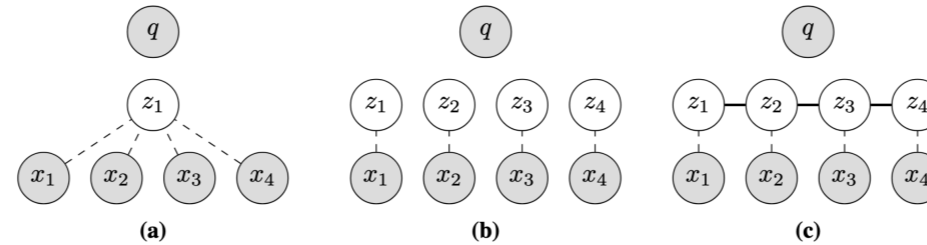
Label	Supervised Re / Pr	Semi-sup. Re / Pr
mwe	75.58 / 81.25	90.70 / 84.78
advmod	87.27 / 85.95	87.32 / 87.51
appos	77.49 / 80.27	81.39 / 81.03

A Latent View of Attention

[Kim et al. 17]

$$p(z_1 = i) = \frac{\exp\{v_q^\top W h_{x_i}\}}{\sum_{j=1}^N \exp\{v_q^\top W h_{x_j}\}}$$

$$p(z_i = 1) = \frac{\exp\{v_q^\top W h_{x_i} + \log \alpha_{i-1,1} + \log \beta_{i+1,1}\}}{\sum_{j \in \{0,1\}} \exp\{\mathbb{1}[j] v_q^\top W h_{x_i} + \log \alpha_{i-1,j} + \log \beta_{i+1,j}\}}$$



$$p(z_i = 1) = \sigma(v_q^\top W h_{x_i} + b)$$

Le and Titov extend the approximate CRF entity linking model of Ganea and Hofmann by including a “relational” attention mechanism between pairs of entities.

* Instead of giving all pairs equal weight, this mechanism effectively weighs certain pairs more highly to favor the influence of pairs which are likely related in the text to the final disambiguation score

Kim et al extend the typical categorical attention mechanism to use marginal probabilities of structured distributions as the attention scores.

* This allows for neighboring attentions to be correlated, or, when using dependency syntax CRF marginals, for word representations to be influenced by their most likely syntactic dependency parents in the sentence

(11) Latent Intention Dialogue Models

[Wen et al. 17]

