Unsupervised Natural Language Parsing

Kewei Tu — ShanghaiTech University
Yong Jiang — Alibaba DAMO Academy
Wenjuan Han — National University of Singapore
Yanpeng Zhao — University of Edinburgh

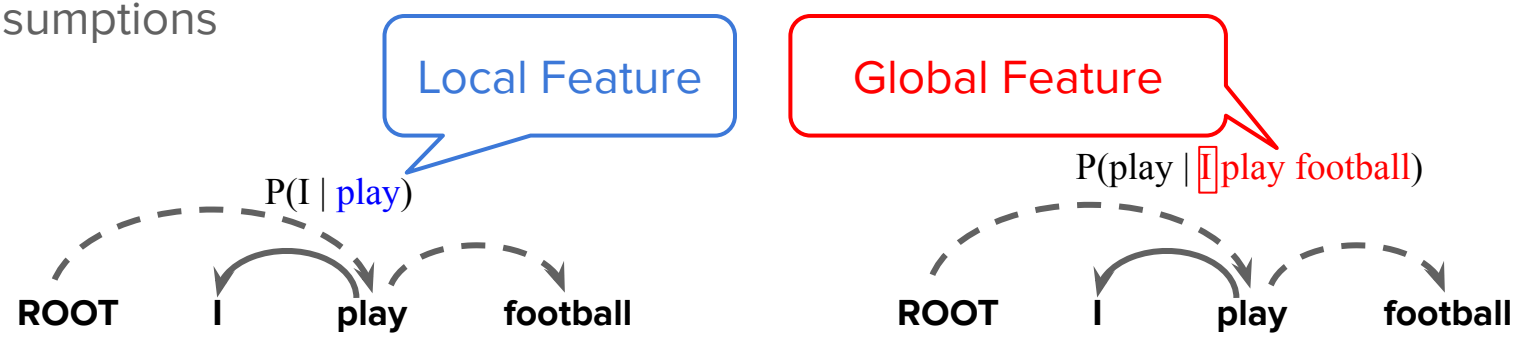
Tutorial Overview

1. Introduction (Kewei)
2. Generative Approaches (Kewei, Yong)
3. Discriminative Approaches (Wenjuan)
4. Special Topics (Yanpeng)
5. Summary (Kewei)

3. Discriminative Approaches

Why Discriminative?

- Generative approaches' limitation: Local Features due to independence assumptions



- Discriminative approaches: Leveraging the information from the whole sentence (Global feature)

Why Discriminative?

Definition: Model the conditional probability P(z|x) or score s(z|x) of the output z (e.g., parse tree) conditioned on the whole sentence.

- Local features ➔ Global features (i.e., contextual features from the whole sentence)
- Limited expressive power ➔ More expressive power

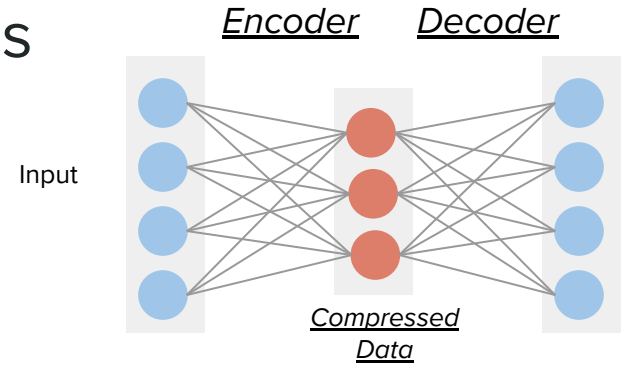
Outline

- Autoencoder-Based Approaches
- Variational Autoencoder-Based Approaches
- Other Approaches
  - Discriminative Clustering-Based Approaches
  - Self-Training-Based Approaches
- Hybrid Approaches

Outline

- Autoencoder-Based Approaches
- Variational Autoencoder-Based Approaches
- Other Approaches
  - Discriminative Clustering-Based Approaches
  - Self-Training-Based Approaches
- Hybrid Approaches

Autoencoder-Based Approaches



Objective Function:

$$J(\Theta) = \sum_{i=1}^{N} \log \mathrm{P}(\hat{\mathrm{x}}^{(i)} | \mathrm{x}^{(i)}; \Theta) + \textit{regularization}$$

Θ: Model parameters
N:#training samples
$\mathrm{x}^{(i)}$: Sentence
$\mathrm{z}^{(i)}$: Parsing tree
$\hat{\mathrm{x}}^{(i)}$: Copy of $\mathrm{x}^{(i)}$

Parse tree could be:

- Hidden variable in the encoder (i.e., Cai et al., 2017, Drozdov et al., 2019)
- Hidden variable in the decoder (i.e., Han et al., 2019a)

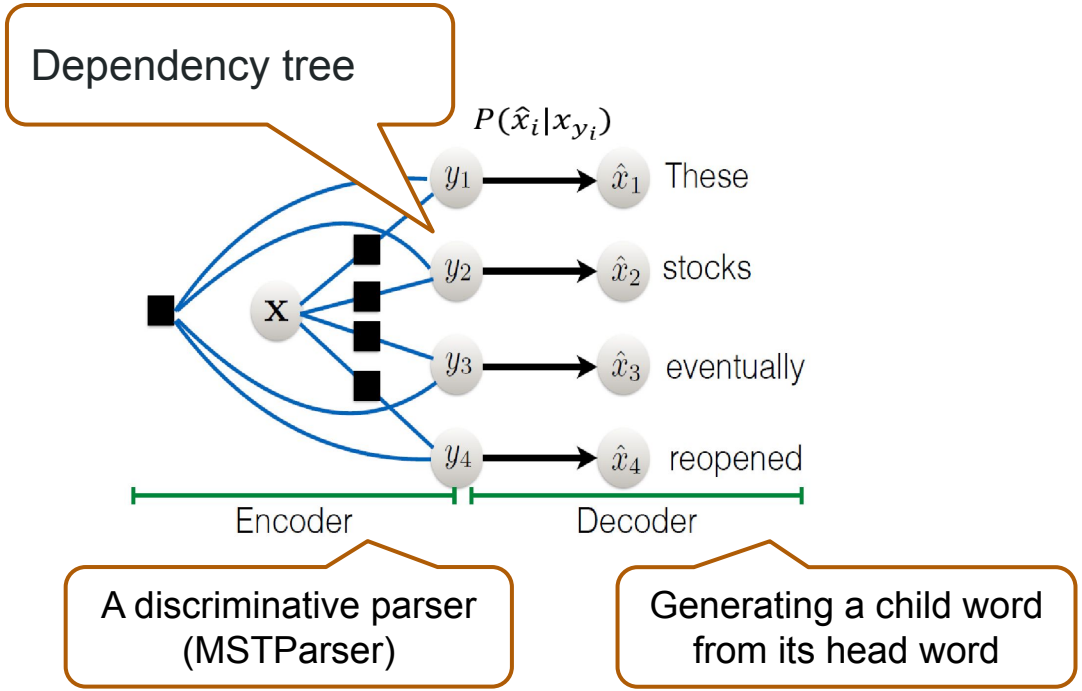
CRF Autoencoder (Cai et al., 2017): Modeling

Encoder: CRF Parser

$$P(\mathrm{z}|\mathrm{x}) = \frac{\exp(\phi(\mathrm{x}, \mathrm{z}))}{Z(\mathrm{x})}$$

Decoder: Multinomial distribution

$$P(\hat{\mathrm{x}}|\mathrm{z}) = \prod_{i=1}^{n} \theta_{\hat{x}_i|t_i}$$



CRF Autoencoder (Cai et al., 2017): Learning

Objective function:

Set of all possible trees.

$$J_1(\mathrm{w}, \theta) = -\sum_{n=1}^{N} \log \left(\sum_{\mathrm{z} \in \mathcal{S}(\mathrm{x}^{(n)})} P(\hat{\mathrm{x}}^{(n)}, \mathrm{z} | \mathrm{x}^{(n)}) \right) + \lambda \Omega(\mathrm{w})$$

Viterbi -- Best tree.

$$J_2(\mathrm{w}, \theta) = -\sum_{n=1}^{N} \log \left(\max_{\mathrm{z} \in \mathcal{S}(\mathrm{x}^{(n)})} P(\hat{\mathrm{x}}^{(n)}, \mathrm{z} | \mathrm{x}^{(n)}) \right) + \lambda \Omega(\mathrm{w})$$

Learning: Gradient Descent

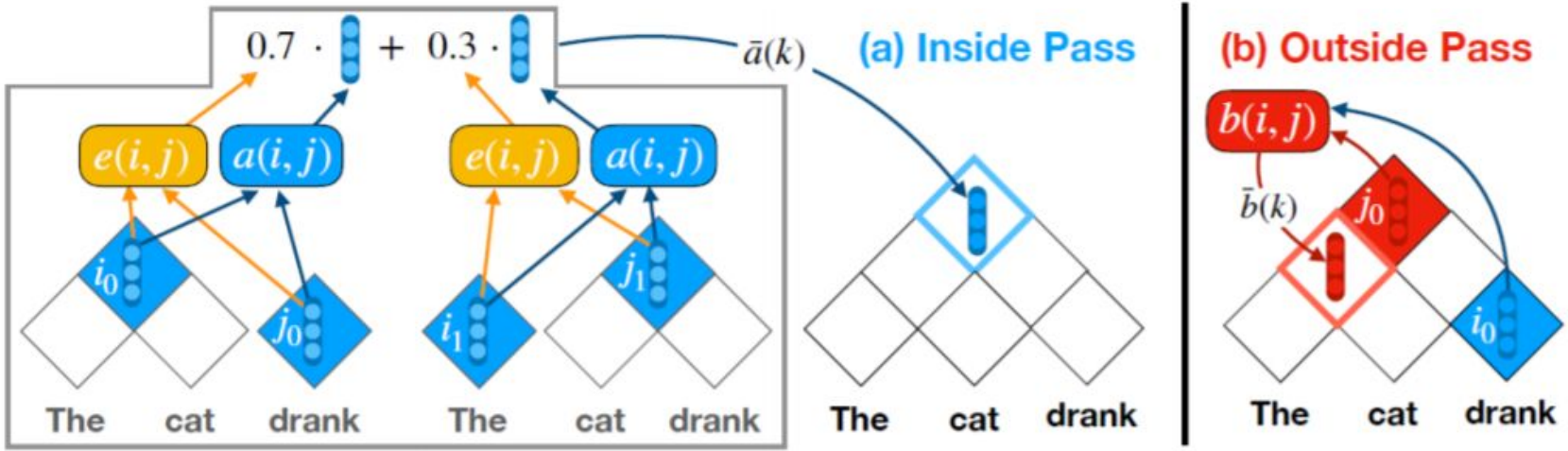
DIORA (Drozdov et al., 2019): Modeling

Encoder: Incorporates the inside-outside algorithm into a latent tree chart parser

Inside representation ⟶ Calculated by bottom-up inside step

Outside representation ⟶ Calculated by top-down outside step

Decoder: Reconstruct leaf input word by outside representation of the leaf cell.



DIORA (Drozdov et al., 2019): Learning

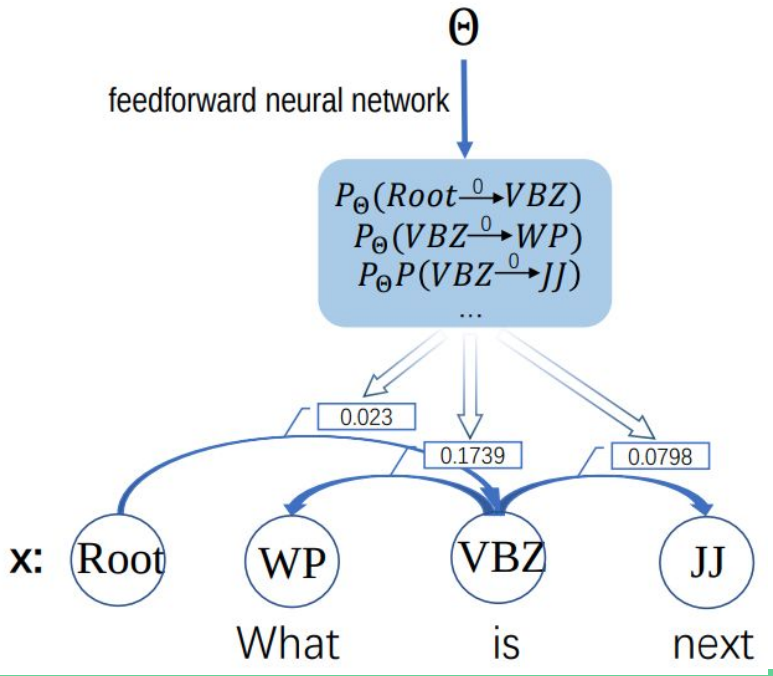
Objective: Reconstruction Probability

Outside representations of the leaf cells should reconstruct the corresponding leaf input word

D-NDMV (Han et al., 2019a) (the deterministic variant): Modeling

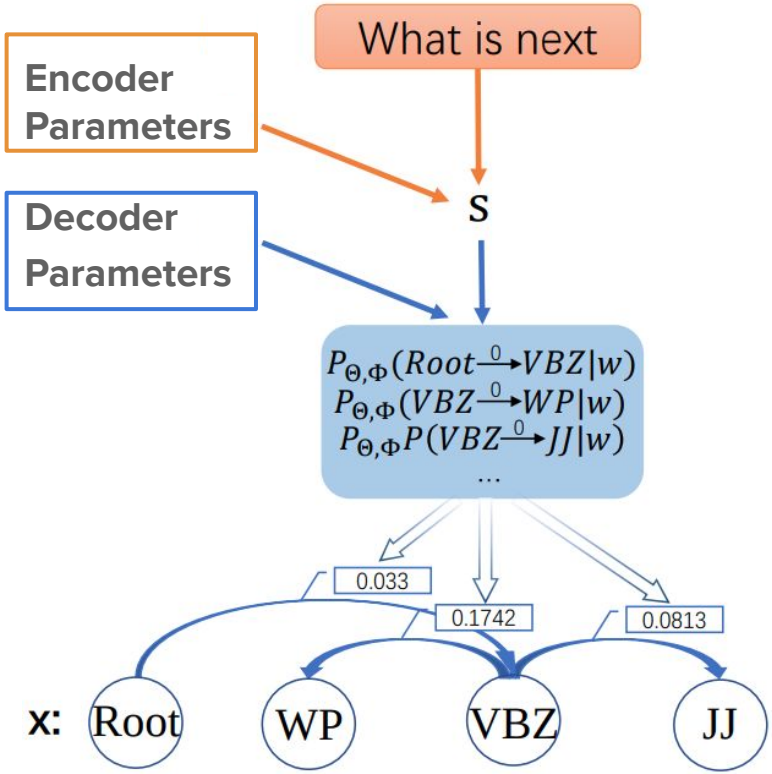
- (N)DMV learns the joint distribution P(x, z) of the given sentence and its parse.
- It lacks the global features of the entire sentence.

(N)DMV: refer to Part 2.



D-NDMV (Han et al., 2019a) (the deterministic variant): Modeling

- D-NDMV learns the joint distribution P(x, z|s) of a sentence and its parse tree conditioned on a continuous latent representation s.
- s encodes global contextual information of the generated sentence.
- Same idea on unsupervised constituency parsing (Kim et al., 2019a).



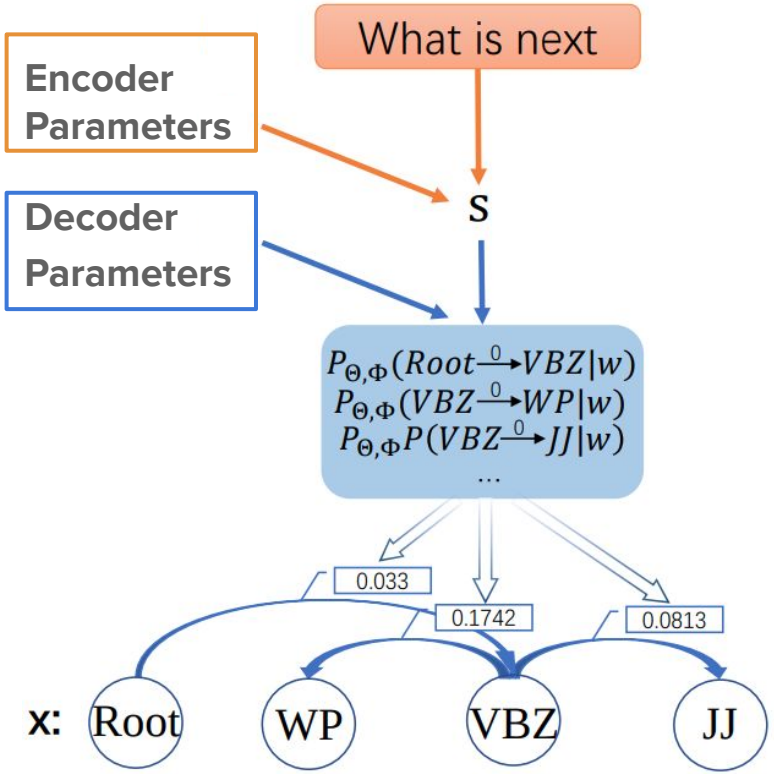
D-NDMV (Han et al., 2019a) (the deterministic variant): Modeling

Encoder: $\mathbf{s} = LSTM(\mathbf{x})$

Feature extractor transforming the observed data to a hidden representation s.

Decoder: $P(\mathbf{x}, \mathbf{z}|\mathbf{s}) = \prod_{\mathbf{r} \in \mathcal{R}(\mathbf{x},\mathbf{z})} P(\mathbf{r}|\mathbf{s})$

Generative latent variable model conditioned on the hidden representation s generating the latent variables as well as the observed data. (i.e., NDMV with s as an additional input)



D-NDMV (Han et al., 2019a) (the deterministic variant): Learning

Expectation–maximization (EM) algorithm: an iterative method between E-step and M-step

In E-step: Set auxiliary distribution $q(\mathbf{z}) = P_\Theta(\mathbf{z}|\hat{\mathbf{x}}^{(n)}, \mathbf{s}^{(n)})$

In M-step: Back-propagate the following objective into model parameters

$$J(\hat{\mathbf{x}}, \mathbf{x}; \Theta) = \frac{1}{N} \sum_{n=1}^{N} \sum_{z \in S(\mathbf{x}^{(n)})} q(\mathbf{z}) \log P_\Theta(\hat{\mathbf{x}}^{(n)}, \mathbf{z}|\mathbf{s}^{(n)}) + \text{constant}$$

Decoder parameters

\hat{x}

x

EM algoritimize the nd of objective ($q(\mathbf{z})$ is an auxiliary distribution):

$$Q(q, \Theta, \Phi) = \log P_{\Theta,\Phi}(\mathbf{x}|\mathbf{w}) - KL(q(\mathbf{z})\|P_{\Theta,\Phi}(\mathbf{z}|\mathbf{x}, \mathbf{w}))$$

Encoder parameters

- E-step: compute the e ected counts $E_{q(\mathbf{z})}c(r, \mathbf{x}, \mathbf{z})$ based on the optimal q which is set as

$$q(\mathbf{z}) = P_{\Theta,\Phi}(\mathbf{z}|\mathbf{x}, \mathbf{w})$$

- M-step: back-propagate the following objective into the parameters Θ, Φ.

$$Q(\Theta, \Phi) = \sum_r \log p(r|\mathbf{w}) E_{q(\mathbf{z})}c(r, \mathbf{x}, \mathbf{z}) - \text{Constant}$$

where r ranges over all the grammar rules

Outline

- Autoencoder-Based Approaches
- Variational Autoencoder-Based Approaches
- Other Approaches
  - Discriminative Clustering-Based Approaches
  - Self-Training-Based Approaches
- Hybrid Approaches

Variational Autoencoder-Based Approaches

Autoencoder:

- Encoder, Decoder

Variational Autoencoder:

- Encoder, Decoder, **Prior** Probability

Objective Function:

Evidence Lower Bound (ELBO): Lower bound of the marginalized probability

Marginalized Probability:

$$J(\Theta) = \sum_{i=1}^{N} \log \mathrm{P}(\mathrm{x}^{(i)}; \Theta)$$

Variational Autoencoder-Based Approaches

Autoencoder:

- Encoder, Decoder

Variational Autoencoder:

- Encoder, Decoder, **Prior** Probability

Objective Function:

Evidence Lower Bound (ELBO): Lower bound of the marginalized probability

ELBO: Conditional likelihood of the training data and an regularisation term given by the KL divergence

$$
\begin{aligned}
& \ln p(x) \\
&= \ln \int_z p(x,z) \\
&= \ln \int_z p(x,z) \frac{q(z|x)}{q(z|x)} \\
&\geq \mathbb{E}_{q(z|x)}[\ln \frac{p(x,z)}{q(z|x)}] \\
&= \mathbb{E}_{q(z|x)}[\ln \frac{p(x|z)p(z)}{q(z|x)}] \\
&= \mathbb{E}_{q(z|x)}[\ln p(x|z)] + \mathbb{E}_{q(z|x)}[\ln \frac{p(z)}{q(z|x)}] \\
&= \mathbb{E}_{q(z|x)}[\ln p(x|z)] + \int_z q(z|x) \ln \frac{p(z)}{q(z|x)} \\
&= \mathbb{E}_{q(z|x)}[\ln p(x|z)] - D_{KL}[q(z|x)||p(z)] \\
&= likelihood - KL
\end{aligned}
$$

D-NDMV (Han et al., 2019a) (the variational variant): Modeling

- Probabilistically models the intermediate continuous vector conditioned on the input sentence using a **Gaussian distribution**.
- Specifies a Gaussian **prior** over the intermediate continuous vector.

$$P(\mathrm{x},\mathrm{z}) = \int P(\mathrm{s}) \prod_{\mathrm{r}\in\mathcal{R}(\mathrm{x},\mathrm{z})} P(\mathrm{r}|\mathrm{s})d\mathrm{s}$$

D-NDMV (Han et al., 2019a) (the variational variant): Learning

- Almost the same as the deterministic variant except for a variational posterior distribution q and an additional KL term in the objective:

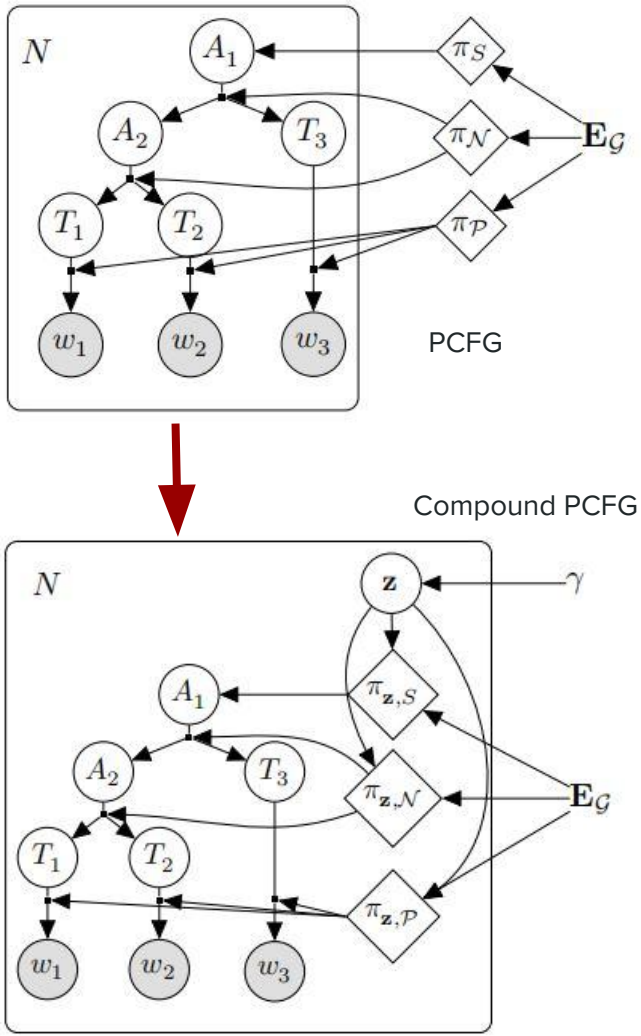
$$-KL(q_\Theta(\mathbf{s}^{(n)}|\mathbf{x}^{(n)})|P(\mathbf{s}^{(n)}))$$

- Share the same formulation of the encoder with vanilia VAE

Compound PCFG (Kim et al., 2019a)

Similar formulation.

- Use terminal/nonterminal embedding as input of neural networks to calculate Neural Probabilistic Context-Free Grammars rule's probability ➜ (Neural PCFG)
- Use LSTM+Gaussian to sample the representation of the sentence and then input it to the neural network, thereby affecting the calculation of rule probability



RNNG Based Approaches (Li et al., 2019, Kim et al., 2019b): Modeling

Background:

Recurrent Neural Network Grammars (RNNG) is a transition-based constituent (Dyer et al., 2016)/dependency (Li et al. 2019) parser.

Two variants of RNNG:

- Discriminative variant: Produces parse from sentence

- Generative variant: Produces both parse and sentence simultaneously

RNNG Based Approaches (Li et al., 2019, Kim et al., 2019b): Modeling

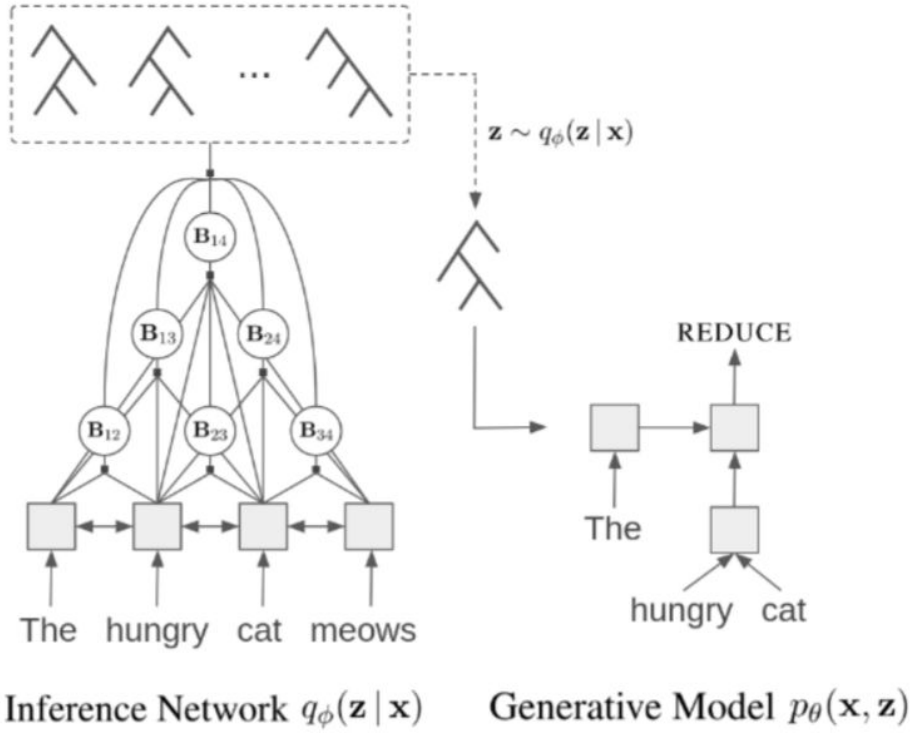
In unsupervised setting:

Encoder:

- Discriminative RNNG (Li et al., 2019) or CRF-parser (Kim et al., 2019b)

Decoder:

- Generative RNNG



Corro and Titov, 2018

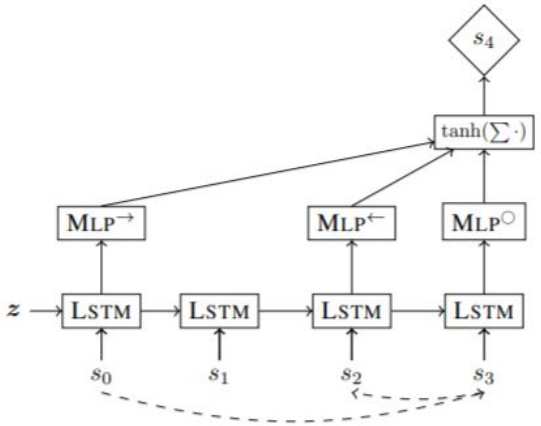
Modeling:

Encoder:

- CRF parser

Decoder:

- Graph convolutional neural network: Specify structure by the dependency tree to generate a sentence



Use Gumbel random perturbation as an efficient approximate sampling algorithm.

Objective: ELBO

PS: This work is performed in a semi-supervised setting. But it can be used as an unsupervised approach.

GAP (Zhang and Goldwasser, 2020)

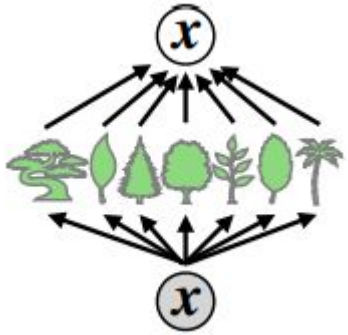
Modeling:

Encoder:

- CRF-parser

Decoder:

- Multinomial distribution: head->modifier



Compute all the possible dependency trees in an arc-decomposed manner, then regard each directed arc as an indicator variable from a Bernoulli distribution

Objective: ELBO

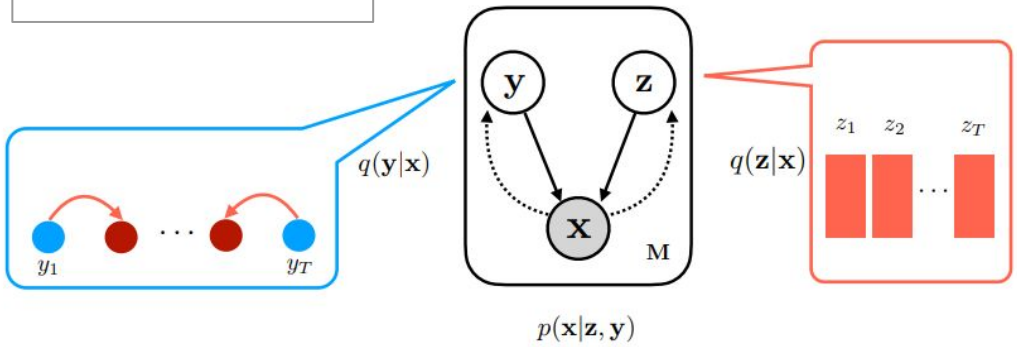
PS: This work is performed in a semi-supervised setting. But it can be used as an unsupervised approach.

Wang and Tu, 2020: Modeling

Modeling:

Continuous Latent Variable

Discrete Latent Variable



Encoder:

- Biaffine-parser

Decoder:

- Multinomial distribution: head->modifier

Divide the latent variables into: the discrete one for dependency tree and the continuous one for the content of sentences to avoid the difficulty in sampling

Objective: ELBO

PS: This work is performed in a semi-supervised setting. But it can be used as an unsupervised approach.

Outline

- Autoencoder-Based Approaches
- Variational Autoencoder-Based Approaches
- Other Approaches
  - Searching-Based Approaches
  - Discriminative Clustering-Based Approaches
  - Self-Training-Based Approaches
- Hybrid Approaches

Searching-Based Approaches (Daum´e III, 2009)

- SEARN in supervised setting considers each substructure prediction as a classification problem. Each classification is based on any part of the input and any previous decisions

- Adapts SEARN to Unsupervised SEARN by first predicting parse trees and then predicting sentences based on parse trees

Outline

- Autoencoder-Based Approaches
- Variational Autoencoder-Based Approaches
- Other Approaches
  - Discriminative Clustering-Based Approaches
  - Self-Training-Based Approaches
- Hybrid Approaches

Discriminative Clustering-Based Approaches (Grave and Elhadad, 2015): Modeling

First-order graph-based discriminative parser

- Parse tree y is represented as a binary vector with length n × n where each element is 1 if the corresponding arc is in the tree and 0 otherwise
- The weight of each edge is calculated by x•w, where w is the parameters

Search for the parse y and learn the parser with parameters w simultaneously.

Discriminative Clustering-Based Approaches (Grave and Elhadad, 2015): Learning

Learning objective makes the searched parses be close to the predicted parses by the parser. In other words, the parses should be easily predictable by the parser

Objective function:

$$\frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{2n_i}||\mathrm{y}_i - \mathrm{X}_i\mathrm{w}||_2^2 - \mu\mathrm{v}^T\mathrm{y}_i\right) + \frac{\lambda}{2}||\mathrm{w}||_2^2$$

- ➔ w and y are iteratively updated.
- ➔ The Frank-Wolfe algorithm is employed to update y and SGD is employed to update w.

v: Whether each dependency arc in y_i satisfies a set of pre-specified linguistic rules

λ, μ: Hyper-parameters

Outline

- Autoencoder-Based Approaches
- Variational Autoencoder-Based Approaches
- Other Approaches
  - Discriminative Clustering-Based Approaches
  - Self-Training-Based Approaches
- Hybrid Approaches

Self-Training-Based Approaches (Le and Zuidema, 2015)

Modeling: Do not set constraints on model architectures

Learning: Iterated reranking

- Produce the dependency tree by a unsupervised parser
- Iteratively improve these trees using supervised parsers that are trained on these trees

Outline

- Autoencoder-Based Approaches
- Variational Autoencoder-Based Approaches
- Other Approaches
  - Discriminative Clustering-Based Approaches
  - Self-Training-Based Approaches
- Hybrid Approaches

Jointly train (Jiang et al., 2017): Modeling

- Generative Approaches
  - Suitable for unsupervised learning
  - Easy to incorporate inductive bias
- Discriminative Approaches
  - Utilize rich features from the input sentence

Can we combine the two?

Jointly train (Jiang et al., 2017)

Do not set constraints on model architectures

Jointly train two models with a combined **objective**

$$J(\mathsf{M_F}, \mathsf{M_G}) = \sum_{\alpha=1}^{N} \min_{\mathbf{y}_\alpha \in \mathcal{Y}_\alpha} \left(F(\mathbf{x}_\alpha, \mathbf{y}_\alpha; \mathsf{M_F}) + G(\mathbf{x}_\alpha, \mathbf{y}_\alpha; \mathsf{M_G}) \right)$$

Objective of a generative model (e.g., LC-DMV)

Objective of a discriminative model (e.g., Convex-MST)

Jointly train (Jiang et al., 2017)

Learning by hard-EM

- M-step: given parses, do supervised learning
- E-step: given the two models, find the parses that optimizes the combine objective function
  - Solved by dual decomposition (Dantzig and Wolfe, 1960)

Input: Sentence $\mathbf{x}$, fixed parameters $\mathbf{w}$ and Θ
Initialize vector $\mathbf{u}$ of size $n \times n$ to $\mathbf{0}$
repeat
  $\hat{\mathbf{y}} = \arg\min_{\mathbf{y} \in \mathcal{Y}} F(\mathbf{x}, \mathbf{y}; \Theta) + \mathbf{u}^T \mathbf{y}$
  $\hat{\mathbf{z}} = \arg\min_{\mathbf{z} \in \mathcal{Y}} G(\mathbf{x}, \mathbf{z}; \mathbf{w}) - \mathbf{u}^T \mathbf{z}$
  if $\hat{\mathbf{y}} = \hat{\mathbf{z}}$ **then**
    return $\hat{\mathbf{y}}$
  else
    $\mathbf{u} = \mathbf{u} + \tau (\hat{\mathbf{y}} - \hat{\mathbf{z}})$
  end if
until Convergence