Semi-Supervised Dependency Parsing with Arc-Factored Variational Autoencoding

Ge Wang, Kewei Tu

ShanghaiTech University, Shanghai, China
{wangge,tukw}@shanghaitech.edu.cn

November 30, 2020
Outline

1. Motivation
2. Arc-Factored Variational Autoencoding
3. Semi-Supervised Learning
4. Experiments and Analysis
5. Conclusion
Outline

1. Motivation
2. Arc-Factored Variational Autoencoding
3. Semi-Supervised Learning
4. Experiments and Analysis
5. Conclusion
In dependency parsing, we try to find the dependency tree for each input sentence.

Training sentences labelled with golden parse trees are used to train parsing models.
Labelling training sentences with dependency trees is laborious and time-consuming.

The scarcity of labelled training samples is the main bottleneck in dependency parsing.
Semi-supervised methods are developed to tackle this problem. Besides labelled data, semi-supervised methods also utilize unlabelled data in training.
Variational Autoencoder (VAE) is an effective framework in utilizing unlabelled data.

The tree structure brings challenge for applying VAE in semi-supervised dependency parsing.

The content of training sentences and dependency structures encoded by the latent variables.
Outline

1 Motivation

2 Arc-Factored Variational Autoencoding

3 Semi-Supervised Learning

4 Experiments and Analysis

5 Conclusion
Following previous work (Corro and Titov, 2019), we 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.

Adaptions were made for the arc-factored setting (trees are split into arcs): latent variables are attributed to each token in sentences.
For VAE, the benefit of the arc-factored setting is obvious: no need to sample an entire tree for the structural latent variables, exact inference can be done instead; the learning procedure can be implemented in parallel for each token in sentences, etc.
Concretely, we formulate a input sentence as $x = x_{1:T}$, corresponding latent vectors $z = z_{1:T}$ and a latent tree $y = y_{1:T}$, where $y_i$ is the index of the head for $x_i$.

The ELBO for VAE is reformulated as

$$
\sum_{t} \mathbb{E}_{Q(z_{yt} | x)} \mathbb{E}_{Q(y_{t} | x)} \log P(x_{t} | z_{yt}, y_{t}) - \text{KL}(Q(y_{t} | x)Q(z_{yt} | x) \| P(y_{t})P(z_{yt}))
$$
Model Formulation

- During training procedure we relax the tree restriction in both encoding and decoding.
- For the encoding part, we use a biaffine encoder which gives the probability of dependency head selection for each token.

\[
p(y_i = 1|\mathbf{x}) \quad p(y_i = j|\mathbf{x}) \quad p(y_i = T|\mathbf{x})
\]

\[
p(y_j = 1|\mathbf{x}) \quad p(y_j = i|\mathbf{x}) \quad p(y_j = T|\mathbf{x})
\]
In the encoding part, we also encode each token along with their context to continuous latent vectors $z$. 

\[
\begin{array}{cccc}
z_1 & z_i & z_j & z_T \\
\uparrow & \uparrow & \uparrow & \uparrow \\
x_1 & x_i & x_j & x_T \\
\end{array}
\]
For the decoding part, we directly reconstruct each token given the latent vector for the head position since the contextual information is contained in it.
Model Architecture

\[ q(y_i | x) \]

\[ q(z_j | x) \]

\[ p(x_i | z_{y_i}, y_i = j) \]

\[ z_0 \]

\[ z_j \]

\[ z_T \]
Outline

1 Motivation

2 Arc-Factored Variational Autoencoding

3 Semi-Supervised Learning

4 Experiments and Analysis

5 Conclusion
Objective

- In semi-supervised learning methods, training data $\mathcal{D}$ is split into labelled data $\mathbb{L}$ and unlabelled data $\mathbb{U}$.
- The overall loss function $\mathcal{L}(\mathcal{D})$ thus consists of labelled loss $\mathcal{L}_l(\mathbb{L})$ and unlabelled loss $\mathcal{L}_u(\mathbb{U})$:

$$\mathcal{L}(\mathcal{D}) = \alpha \mathcal{L}_l(\mathbb{L}) + (1 - \alpha) \mathcal{L}_u(\mathbb{U})$$

We rewrite the loss function as the sum of three terms where $\mathcal{E}_{\phi,\theta}(x)$ is the ELBO formulation:

$$\mathcal{L}_{\theta,\phi}(\mathcal{D}) = -\alpha \sum_{(x,y^*) \in \mathbb{L}} q_{\phi}(y^* | x)$$

$$- \alpha \sum_{x \in \mathbb{L}} \mathcal{E}_{\phi,\theta}(x) - (1 - \alpha) \sum_{x \in \mathbb{U}} \mathcal{E}_{\phi,\theta}(x)$$
Semi-Supervised Learning

Learning

- Directly training autoencoders may cause the learned structures to diverge from the correct ones.
- Some tricks should be used in the learning process such as starting with the parsing loss on the labelled data and then optimizing the complete loss function.

Epoch $i$

\[ \text{Optimize the Parsing Loss} \rightarrow \text{Optimize the Complete Loss} \]

Epoch $i+1$

\[ \text{Optimize the Parsing Loss} \rightarrow \ldots \]
Learning

- Note that the tree constraint over $y$ is enforced only in the validation and test phase.
- The learned encoder is used to do tree predictions with maximum spanning tree (MST) algorithms.
Outline

1. Motivation
2. Arc-Factored Variational Autoencoding
3. Semi-Supervised Learning
4. Experiments and Analysis
5. Conclusion
We evaluate our method on 7 languages with the setting of 10% labelled data and 90% unlabelled data.

<table>
<thead>
<tr>
<th></th>
<th>English</th>
<th>French</th>
<th>German</th>
<th>Italian</th>
<th>Spanish</th>
<th>Hindi</th>
<th>Swedish</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours-Sup</td>
<td>92.00</td>
<td>84.58</td>
<td>80.41</td>
<td>87.11</td>
<td>84.63</td>
<td>92.22</td>
<td>80.29</td>
</tr>
<tr>
<td>Self-Training</td>
<td>91.82</td>
<td>85.27</td>
<td>81.33</td>
<td>87.62</td>
<td>85.08</td>
<td>91.74</td>
<td>78.33</td>
</tr>
<tr>
<td>NCRFAE</td>
<td>91.94</td>
<td>84.83</td>
<td>80.70</td>
<td>87.33</td>
<td>84.31</td>
<td>92.49</td>
<td>80.33</td>
</tr>
<tr>
<td>Ours-Semi</td>
<td><strong>92.55</strong></td>
<td><strong>85.57</strong></td>
<td><strong>81.52</strong></td>
<td><strong>88.58</strong></td>
<td><strong>85.46</strong></td>
<td><strong>92.56</strong></td>
<td><strong>80.85</strong></td>
</tr>
</tbody>
</table>

**Table:** The UAS results of our semi-supervised model compared with three baseline models on the datasets of 7 languages.
Experiments and Analysis

Comparision with More Sophisticated Methods

We also compare our method with the more sophisticated one proposed in Corro and Titov, 2019 denoted as C&T.

<table>
<thead>
<tr>
<th></th>
<th>English</th>
<th>French</th>
</tr>
</thead>
<tbody>
<tr>
<td>C&amp;T-Sup</td>
<td>88.79</td>
<td>84.09</td>
</tr>
<tr>
<td>Ours-Sup (weakened)</td>
<td>88.58</td>
<td>84.05</td>
</tr>
<tr>
<td>C&amp;T</td>
<td>89.50</td>
<td>84.69</td>
</tr>
<tr>
<td>Ours-Semi(weakened)</td>
<td><strong>89.67</strong></td>
<td><strong>84.94</strong></td>
</tr>
</tbody>
</table>

**Table:** The UAS results of our semi-supervised model compared with C&T. our encoder is deliberately weakened to make it comparable with that of C&T.
Varying the Proportion of Labelled Data

To analyze the impact brought by the amount of labelled data, we vary the proportion of labelled data and observe the changes in the UAS score.

<table>
<thead>
<tr>
<th>Ratio</th>
<th>Ours-Sup</th>
<th>Ours-Semi</th>
<th>∆ UAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01:0.99</td>
<td>85.72</td>
<td>86.21</td>
<td>+0.49</td>
</tr>
<tr>
<td>0.1:0.9</td>
<td>92.00</td>
<td>92.55</td>
<td>+0.55</td>
</tr>
<tr>
<td>0.3:0.7</td>
<td>93.94</td>
<td>94.15</td>
<td>+0.21</td>
</tr>
<tr>
<td>0.5:0.5</td>
<td>94.38</td>
<td>94.41</td>
<td>+0.03</td>
</tr>
</tbody>
</table>

**Table:** The UAS results of our supervised model and semi-supervised model when the proportion of labelled data varies.
Experiments and Analysis

Time Efficiency

As our model is arc-factored without the tree constraint, the time efficiency in both encoding and decoding is one of our prominent advantages.

<table>
<thead>
<tr>
<th></th>
<th>Encoder</th>
<th>Decoding</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sec/Sen</td>
<td>Ours</td>
</tr>
<tr>
<td>Ours</td>
<td>0.142</td>
<td></td>
</tr>
<tr>
<td>NCRFAE</td>
<td>2.365</td>
<td></td>
</tr>
</tbody>
</table>

![Diagram showing time efficiency comparison between Ours and NCRFAE in encoding and decoding processes.]
Outline

1. Motivation
2. Arc-Factored Variational Autoencoding
3. Semi-Supervised Learning
4. Experiments and Analysis
5. Conclusion
Conclusion

- We presented a semi-supervised dependency parsing model based on the VAE framework.
- We relax the tree constraint during training to tackle the challenges in VAE brought by the tree structure.
- The experiments show that the simplicity of the model does not hinder its performance and it achieves better performance with faster speed than baseline work.
Semi-Supervised Dependency Parsing with Arc-Factored Variational Autoencoding

Ge Wang, Kewei Tu

ShanghaiTech University, Shanghai, China
{wangge, tukw}@shanghaitech.edu.cn

November 30, 2020