Code

Paper

Second-Order Unsupervised Neural Dependency Parsing

Songlin Yang[1], Yong Jiang[2], Wenjuan Han[3], Kewei Tu[1]
[1]School of Information Science and Technology, ShanghaiTech University
[2]Alibaba DAMO Academy, Alibaba Group
[3]Department of Computer Science, National University of Singapore

Motivation

Most of the unsupervised dependency parsers are based on first-order probabilistic generative models that only consider local parent-child information, which lacks expressiveness.

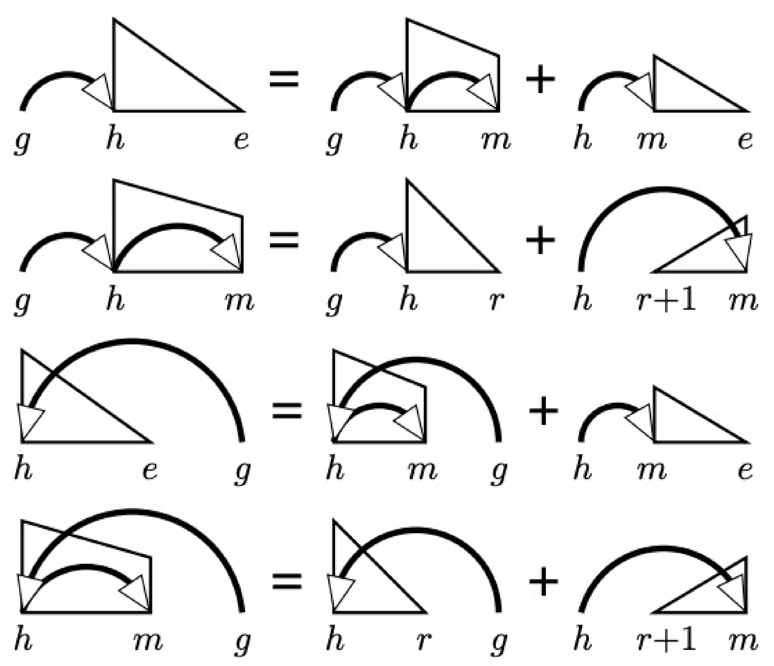
Proposed Solution

We proposed a second-order extension of unsupervised neural dependency models that incorporate grandparent-child or sibling information. To reduce the number of parameters, we use the decomposed trilinear function to score. To reduce the number of grammar rules, we use the agreement-based learning framework to jointly train a second-order unlexicalized model and a first-order lexicalized model.

Empirical Results

Experiments on WSJ dataset show that our unlexicalized second-order sibling NDMV model outperforms the previous state of the art unsupervised parser. Our joint first-order unlexicalized and second-order sibling NDMV model improves the result further. Experiments on seven languages from UD dataset also show the effectiveness of our method.

Second-Order Parsing



The picture shows the dynamic-programming structures and derivations of second-order (grandparent-child variant) parsing model (Koo and Collin, 2010). We design dynamic programming algorithms adapted from their ideas to calculate the marginal likelihood and obtain the Viterbi parse tree.

Learning via Gradient Method

$$\begin{aligned}\nabla_\theta(\log p_\theta(x)) &= \sum_{z\in\mathcal{T}(x)} p_\theta(z\mid x)\nabla_\theta \log p_\theta(x,z)\\ &= \sum_{z\in\mathcal{T}(x)} p_\theta(z\mid x)\sum_{r\in\mathcal{R}} c(r,x,z)\nabla_\theta \log p_\theta(r)\\ &= \sum_{r\in\mathcal{R}} e(r,x)\nabla_\theta \log p_\theta(r)\end{aligned}$$

where $c(r,x,z)$ is the number of times rule r is used in dependency parse tree z of sentence x, e(r,x) is the expected count of rule r in sentence x, $\mathcal{T}(x)$ is the whole set of parse tree. In the EM algorithm, we need to calculate the expected counts of grammar rules for the entire training data set, while in mini-batch gradient ascent, we calculate the expected counts of grammar rules in the mini-batch, which is similar to online EM algorithm.

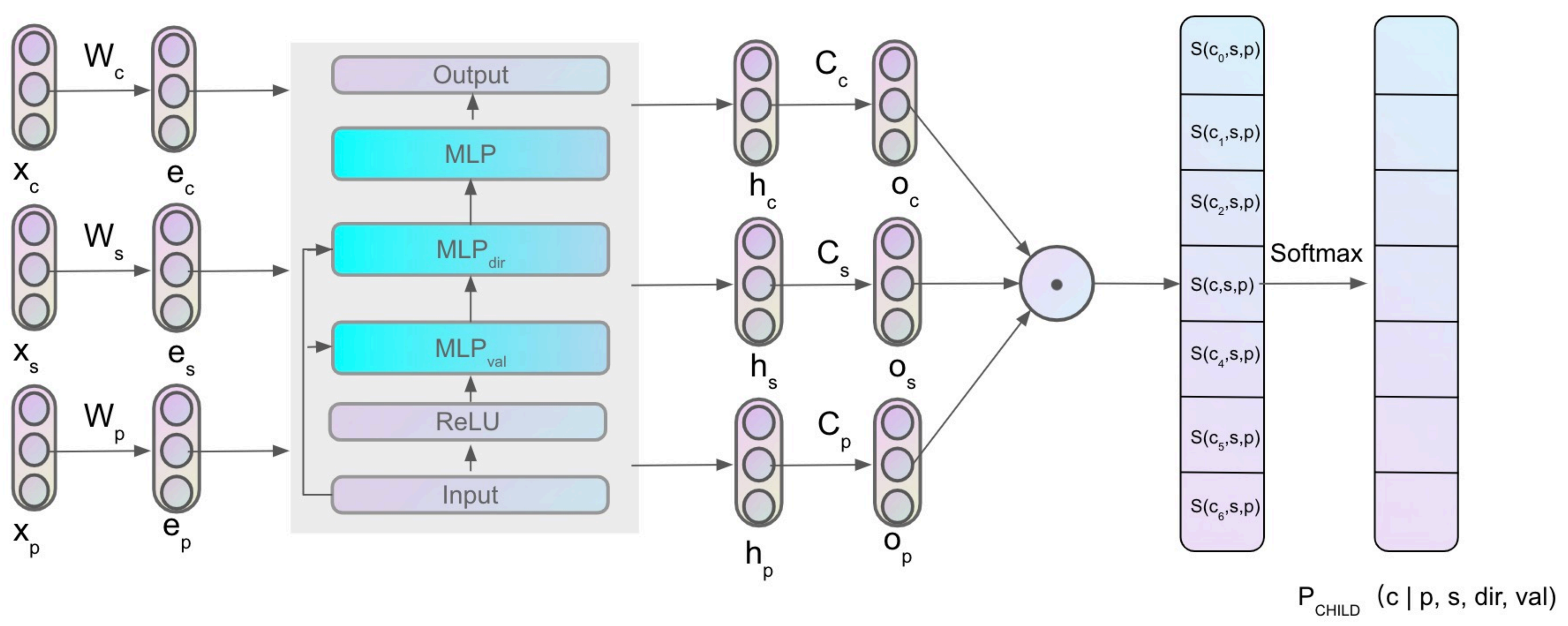
Agreement-Based Learning

In our second-order DMV model, the number of grammar rules is $4|V|^3+4|V|^2+|V|$, which is cubic in the vocabulary size $|V|$. When our model is lexicalized, the vocabulary may contain thousands of words or more, making the model size less manageable. Instead of learning a second-order lexicalized model, we propose to jointly learn a second-order unlexicalized model (whose vocabulary consists of POS tags instead of words) and a first-order lexicalized model based on the agreement-based learning framework (Liang et al., 2007). The jointly learned models have a manageable number of grammar rules while still benefiting from both second-order parsing and lexicalization. The joint objective function is shown below:

$$\mathcal{O}_{\text{agree}}(\theta) \stackrel{\text{def}}{=} \log \sum_{z\in\mathcal{T}(x)} (p_{\theta_0}(x,z)\cdot p_{\theta_1}(x,z))$$

where θ_0 is the parameters of the lexicalized first-order NDMV and θ_1 is the parameters of the unlexicalized second-order NDMV.

Neural Architecture and Decomposed Trilinear Scoring Function



We use a shared POS embedding layer and apply different linear transformations to get role-specific representations for child, parent, grandparent, or sibling, then go through MLP layers to encode the direction and valence information. We use a decomposed trilinear function to compute the unnormalized rule probability from the three vectors h_c, h_p, h_s.

Experiments

Methods	WSJ10	WSJ
DMV	58.3	39.4
UR-A E-DMV	71.4	57.0
CRFAE	71.7	55.7
Neural DMV	72.5	57.6
HDP-DEP	73.8	-
VV D-NDMV	75.5	60.4
DV D-NDMV	75.6	61.4
L-NDMV	75.1	59.5
grand-NDMV	71.4	57.3
sibling-NDMV	77.5	64.5
L-grand-NDMV	63.0	52.6
L-sibling-NDMV	78.3	66.4
Joint grand-NDMV + L-NDMV	76.0	64.3
Joint sibling-NDMV + L-NDMV	**79.9**	**67.5**

Table 1. Result on Wall Street Journal (WSJ) dataset. We train on sentences of length ≤ 10 and report the result for both WSJ10 and the whole WSJ.

	NDMV	LD	DV	VV	+sibling	+grand
Basque	**47.8**	45.4	39.9	42.4	30.5	33.0
Dutch	35.6	34.1	42.4	43.7	**45.0**	43.7
French	38.1	48.6	57.2	58.5	**64.9**	61.4
German	50.4	50.5	54.5	52.9	**61.5**	46.7
Italian	63.6	71.1	60.2	61.3	**71.8**	69.7
Polish	62.8	63.7	66.7	73.0	**75.0**	62.4
Portuguese	49.0	**67.2**	64.7	65.7	63.7	60.3
Spanish	58.0	61.9	64.3	64.4	**66.8**	65.0
Average	50.7	55.3	56.2	57.7	**59.9**	55.2

Table 2. Results on seven languages from Universal Dependency (UD) Treebanks. We train on sentences of length ≤ 15 and test on sentences of length ≤ 40.

	UAS (WSJ10 / WSJ)		
	L-NDMV	sibling-NDMV	joint parsing
separate training	76.6 / 62.7	77.5 / 64.8	78.4 / 65.8
joint training	79.2 / 65.4	78.7 / 65.6	79.9 / 67.5

Table 3. The effect of joint training and joint parsing. We found joint parsing is helpful and joint training improves the performance of both models.