LING5702: Lecture Notes 25
Neural grammar induction experiments

Contents

25.1 Neural inducer (Jin et al., 2021)

We get better results with a neural inducer, which directly optimizes probability of sentences σ:

$$\mathbf{W}^{(t)} = \mathbf{W}^{(t-1)} - \frac{\partial}{\partial \mathbf{W}^{(t-1)}} \sum_{\sigma \in \mathcal{D}} - \ln \mathsf{P}(\sigma)$$

Sentence probability comes from rule probabilities, as before:

$$\mathsf{P}(\sigma) = \sum_{\tau \text{ for } \sigma} \prod_{\eta \in \tau \text{ s.t. } c_\eta \rightarrow c_{\eta 1}\ c_{\eta 2}} \mathsf{P}(c_\eta \rightarrow c_{\eta 1}\ c_{\eta 2} \mid c_\eta) \cdot \prod_{\eta \in \tau \text{ s.t. } c_\eta \rightarrow w_\eta} \mathsf{P}(c_\eta \rightarrow w_\eta \mid c_\eta)$$

Rule probabilities rely on a terminal/nonterminal decision:

$$\mathsf{P}(\text{Stop}{=}s \mid c_\eta) = \underset{s \in \{0,1\}}{\text{SoftMax}}(\mathbf{W}_{\text{stop}}\ \overbrace{\mathbf{E}\,\delta_{\mathbf{c}_\eta}}^{\text{category embedding}})$$

The non-terminal and terminal probabilities are also estimated by neural networks:

1. If **non-terminal**, we use a neural decision given the expanded category:

$$\mathsf{P}(c_\eta \rightarrow c_{\eta 1}\ c_{\eta 2} \mid c_\eta) = \mathsf{P}(\text{Stop}{=}0 \mid c_\eta) \cdot \underset{c_{\eta 1}, c_{\eta 2} \in C \times C}{\text{SoftMax}}(\mathbf{W}_{\text{nont}}\ \overbrace{\mathbf{E}\,\delta_{c_\eta}}^{\text{category embedding}})$$

2. If **terminal**, we use a different neural decision given the expanded category:

$$\mathsf{P}(c_\eta \rightarrow w_\eta \mid c_\eta) = \mathsf{P}(\text{Stop}{=}1 \mid c_\eta) \cdot \underset{w_\eta \in W}{\text{SoftMax}}(\mathbf{W}_{\text{term}}\ \overbrace{\mathbf{E}\,\delta_{c_\eta}}^{\text{category embedding}})$$

25.2 Character model

Alternatively, we try a **recurrent neural character model** (an 'LSTM'):

$$\mathsf{P}(c_\eta \to w_\eta \mid c_\eta) = \mathsf{P}(\text{Stop=1} \mid c_\eta) \cdot \overbrace{\prod_{\ell_i \in \{\ell_1, \ldots, \ell_n\}} \mathsf{P}(\ell_i \mid c_\eta, \ell_1, \ldots, \ell_{i-1})}^{\text{prob. of each letter comes from LSTM}}$$

$$\mathsf{P}(\ell_i \mid c_\eta, \ell_1, \ldots, \ell_{i-1}) = \underset{\ell_i \in \{\text{a,b,}\ldots\}}{\text{SoftMax}}(\mathbf{W}_{\text{char}}\, \mathbf{h}_{i,B,c_\eta})$$

$$\mathbf{h}_{i,b,c_\eta}, \mathbf{c}_{i,b,c_\eta} = \text{LSTM}(\mathbf{h}_{i,b-1,c_\eta}, \mathbf{h}_{i-1,b,c_\eta}, \mathbf{c}_{i-1,b,c_\eta})$$

$$\mathbf{h}_{0,b,c_\eta}, \mathbf{c}_{0,b,c_\eta} = \text{ReLU}(\mathbf{W}_{b,\text{term}}\, \underbrace{\mathbf{E}\, \delta_{c_\eta}}_{\text{category embedding}}), \mathbf{0}$$

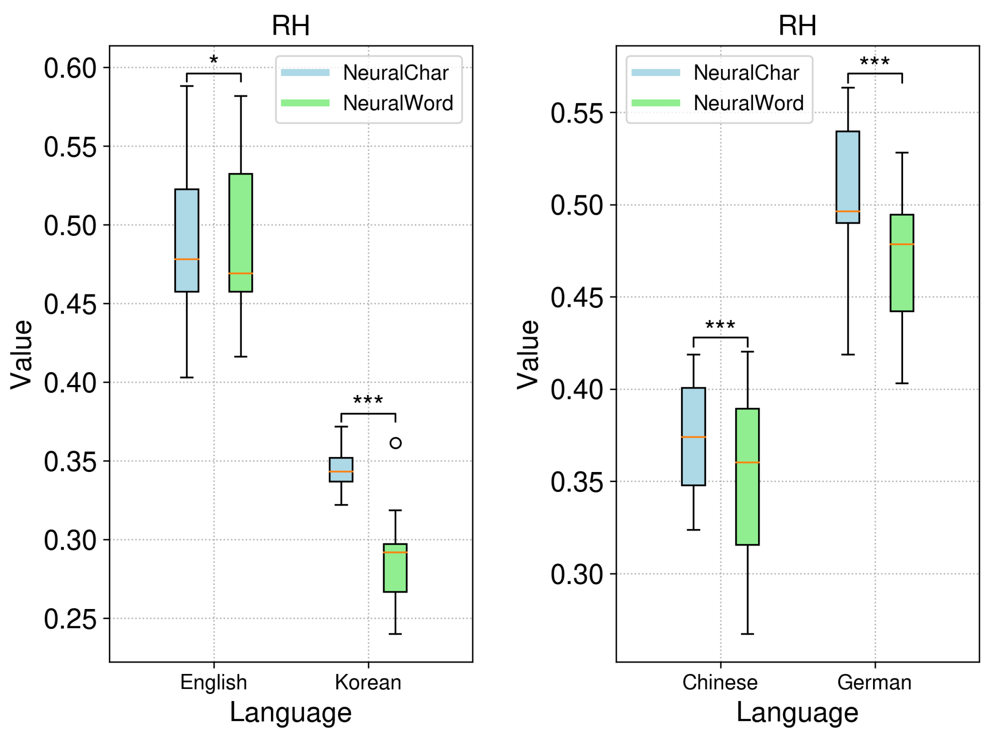
LSTMs (Long Short-Term Memories) have hidden units $\mathbf{h}_{i,b,c_\eta}$ and durable memory cells $\mathbf{c}_{i,b,c_\eta}$.

This lets the model learn patterns of character sequences for each category (e.g. verbs end in *-ing*).

25.3 Results on child-directed speech transcripts: character model is better

Human-annotated corpora

Supervised machine-annotated



Data: MacWhinney (2000).

25.4 Results on newswire data: character model is generally better

Models / RH	Ar	Zh	En	Fr	De	He	Ja	Ko	Pl	Vi	Avg
	Individual languages										
DIMI (Jin et al., 2018)	16.5	12.4	23.4	16.8	10.3	14.9	23.5	7.1	6.3	8.1	13.9
Compound (Kim et al., 2019)	21.1	21.2	**36.8**	37.7	**41.4**	23.5	15.2	5.6	**35.1**	15.8	25.3
Compound-v (Kim et al., 2019)	16.9	22.6	35.0	39.9	39.4	29.1	13.1	7.0	33.0	**24.0**	26.0
L-PCFG (Zhu et al., 2020)	24.4	19.4	15.0	18.2	28.3	17.0	30.1	10.2	17.4	10.2	19.0
NeurWord (Jin et al., 2021)	23.0	20.8	29.7	29.8	33.8	21.6	29.8	11.7	22.0	15.1	23.7
Flow (Jin et al., 2019)	25.4	18.7	21.6	25.3	29.7	25.4	24.4	15.0	31.0	—	24.1
NeurChar (Jin et al., 2021)	**29.1**	**23.9**	33.4	**40.7**	39.3	**29.5**	**40.2**	**16.3**	21.0	12.8	**28.5**

Models / F1	Ar	Zh	En	Fr	De	He	Ja	Ko	Pl	Vi	Avg
	Individual languages										
DIMI (Jin et al., 2018)	35.3	36.6	50.6	39.6	36.4	45.4	36.2	26.5	43.2	**42.7**	39.3
Compound (Kim et al., 2019)	32.4	34.2	**51.7**	48.2	**49.7**	40.5	22.9	19.1	**50.1**	34.3	38.3
Compound-v (Kim et al., 2019)	27.6	37.4	50.9	49.6	47.9	**49.2**	21.6	20.7	47.2	38.3	39.1
L-PCFG (Zhu et al., 2020)	**45.0**	**46.2**	36.2	34.4	46.8	38.4	45.2	30.0	32.1	27.3	38.2
NeurWord (Jin et al., 2021)	36.9	41.3	44.4	41.5	44.4	40.0	42.4	23.3	35.2	37.5	38.7
Flow (Jin et al., 2019)	35.3	38.1	38.6	40.3	38.0	45.0	33.8	34.4	47.1	—	39.0
NeurChar (Jin et al., 2021)	42.0	44.9	49.9	**51.5**	47.7	48.6	**55.9**	**34.6**	33.1	28.7	**43.7**

References

Jin, L., Doshi-Velez, F., Miller, T., Schuler, W., & Schwartz, L. (2018). Depth-bounding is effective: Improvements and evaluation of unsupervised PCFG induction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 2721–2731).

Jin, L., Doshi-Velez, F., Miller, T., Schwartz, L., & Schuler, W. (2019). Unsupervised learning of PCFGs with normalizing flow. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 2442–2452).

Jin, L., Oh, B.-D., & Schuler, W. (2021). Character-based PCFG induction for modeling the syntactic acquisition of morphologically rich languages. In *Findings of the Association for Computational Linguistics: EMNLP 2021* (pp. 4367–4378). Punta Cana, Dominican Republic: Association for Computational Linguistics.

Kim, Y., Dyer, C., & Rush, A. (2019). Compound probabilistic context-free grammars for grammar induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 2369–2385).

MacWhinney, B. (2000). *The CHILDES project: Tools for analyzing talk*. Mahwah, NJ: Lawrence Elrbaum Associates, third edition.

Zhu, H., Bisk, Y., & Neubig, G. (2020). The return of lexical dependencies: Neural lexicalized PCFGs. *Transactions of the Association for Computational Linguistics*, 8, 647–661.