Isolating the Effects of Modeling Recursive Structures: A Case Study in Pronunciation Prediction of Chinese Characters

Minh Nguyen
National University of Singapore
elenguy@nus.edu.sg

Gia H. Ngo
Cornell University
ghn8@cornell.edu

Nancy F. Chen
Institute for Infocomm Research Singapore
nancychen@alum.mit.edu

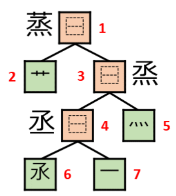
Abstract

Finding that explicitly modeling structures leads to better generalization, we consider the task of predicting Cantonese pronunciations of logographs (Chinese characters) using logographs' recursive structures. This task is a suitable case study for two reasons. First, logographs' pronunciations depend on structures (i.e. the hierarchies of sub-units in logographs) Second, the quality of logographic structures is consistent since the structures are constructed automatically using a set of rules. Thus, this task is less affected by confounds such as varying quality between annotators. Empirical results show that modeling structures explicitly using treeLSTM outperforms LSTM baseline, reducing prediction error by 6.0% relative.

1 Introduction

Modeling structures has led to improvement in machine translation (Yamada and Knight, 2001), natural language inference (Bowman et al., 2016), and parsing (Dyer et al., 2016; Zhang et al., 2016). However, in some cases, modeling structures showed little improvement (Li et al., 2015; Lan and Xu, 2018). The lack of improvement could be due to either (1) models cannot exploit structures effectively or (2) structures provide no additional relevant information.

We consider the task of predicting logographs' Cantonese pronunciation from logographic structures, since the structures provide relevant information for determining logographs' pronunciation (Hsiao and Shillcock, 2006). Figure 1 shows an example of logographic structure. For this task, the quality of structures is consistent since the structures are constructed automatically using a set of rules. Hence, improvement can be attributed directly to whether models can exploit logographic structures effectively.



Character	Onset	Nucleus	Coda
蒸 - to steam (verb)	z	i	ng
烝 - steam (noun)	z	i	ng
丞 - to assist	s	i	ng
氶 - name of a river	c	i	ng
一 - one	j	a	t

Figure 1: *An example of logographic structure. The binary tree on the left represents the logograph 蒸. The leaf nodes (position 2, 5, 6, 7) are sub-units forming the logograph (analogous to letters forming English words). The inner nodes (position 1, 3, 4) are composition operators (such as vertical stacking) applied to children nodes. The sub-trees rooted at positions 3, 4, 5, 6, 7 also form logographs (烝, 丞, 氶, 灬, 一). The table shows the logographs' meanings and their pronunciation in Cantonese.*

In Cantonese, logographs are characters, which can be constructed using the same sub-units. To construct character embeddings for predicting pronunciation, one can apply CNN on images of logographs (Dai and Cai, 2017; Liu et al., 2017; Toyama et al., 2017; Su and Lee, 2017) or average embeddings of characters and sub-units (Shi et al., 2015). We treated a logograph as a binary tree and construct the embedding from the tree by composing its nodes bottom-up.

2 Model

We compared LSTM (Graves, 2013) and bidirectional LSTM (biLSTM), which are structure-agnostic, against treeLSTM (Tai et al., 2015), which is better suited for modeling tree structures. The task-specific layer uses the last hidden layer, $\boldsymbol{h}$, to predict the pronunciation (onset, nucleus and coda).

$$
\begin{aligned}
cd &= \mathrm{softmax}(W^{cd}\boldsymbol{h}) \\
nu &= \mathrm{softmax}(W^{nu}[\boldsymbol{h}, cd]) \\
on &= \mathrm{softmax}(W^{on}[\boldsymbol{h}, cd, nu])
\end{aligned}
$$

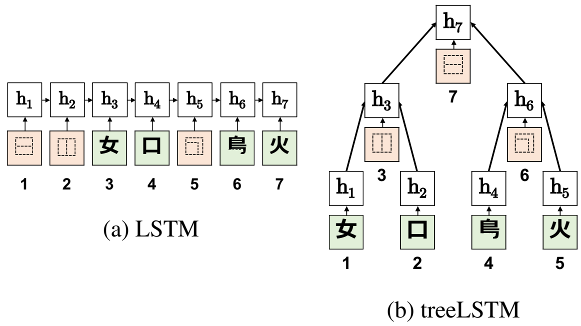


Figure 2: *LSTM and treeLSTM models. In both models,* $\mathbf{h_7}$ *was input of the task-specific layer to predict the logograph's pronunciation.*

Decomposition of logographs into sub-units is necessary to locate the sub-units hinting at pronunciation. Logographs are decomposed recursively into binary trees (Figure 1) using rules defined by the Kyoto University's CHISE[1] project (Morioka, 2008).

3 Experiments

Input	SER	TER	On.	Nu.	Cd.
LSTM 1-layer	58.5	33.1	42.8	37.5	19.0
LSTM 2-layer	57.5	33.3	42.8	38.3	18.9
BiLSTM 1-layer	63.5	37.0	47.0	42.6	21.3
BiLSTM 2-layer	60.6	34.5	45.2	39.3	19.0
treeLSTM	**56.9**	**31.3**	**40.9**	**35.7**	**17.3**

Table 1: *Phonemes prediction error rate (%)*

We extracted entries from the UniHan database, where each entry has a logograph and its Cantonese pronunciation. The data was randomly split into training (16000), validation (2400) and test set (2400). We evaluated the models' accuracy in predicting Cantonese pronunciation (onset, nucleus, and coda) using string error rate (SER) and token error rate (TER). A wrongly predicted onset, nucleus or coda was counted as one token error. An output with one or more token error(s) was counted as one string error. In general, biLSTM performed worse than LSTM so we just compared LSTM against treeLSTM. The treeLSTM yields 6.0% lower relative TER and 1.0% lower relative SER than the 2-layer LSTM respectively (Table 1).

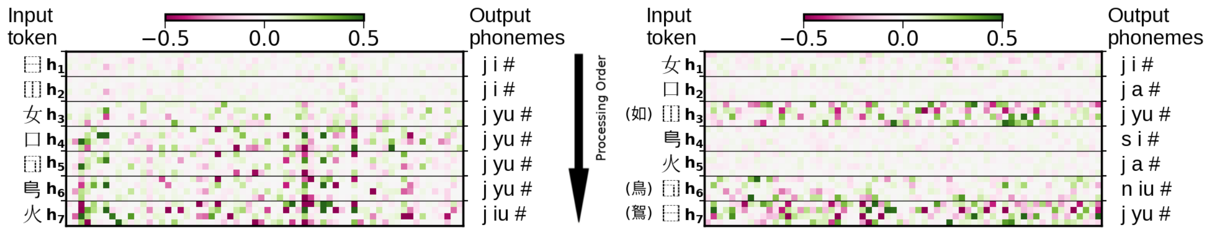


Figure 3: *Predictions for* 鴽 *(meaning: quail, pronunciation:* ***j yu #****). Central panels show the hidden states* $\boldsymbol{h}_i$*. Refer to the order of processing in Figure 2. The sequences on the right of the panels are the predictions using the hidden states* $\boldsymbol{h}_i$*. The final predictions are at the bottom (****j iu #*** *and* ***j yu #****). LSTM made a mistake while treeLSTM did not.*

We also analyzed how the models predict (Figure 3). treeLSTM predicted correctly by focusing on the relevant sub-unit (如), while LSTM obtained the correct prediction but forgot it at the last step.

4 Conclusion

We showed that treeLSTM is better than LSTM at building representation from recursive structures, reducing the relative error rates of pronunciation prediction by 6.0%. Using treeLSTM to build better character representation may benefit other tasks such as language modeling and sentiment analysis.

[1]The Kyoto University's CHaracter Information Service Environment (CHISE) project: http://www.chise.org/
IRG, a committee advising the Unicode Consortium about logographs, uses these rules to check for duplicate characters.

References

Samuel R Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D Manning, and Christopher Potts. 2016. A Fast Unified Model for Parsing and Sentence Understanding. In *Proceedings of ACL*.

Falcon Z Dai and Zheng Cai. 2017. Glyph-aware Embedding of Chinese Characters. In *Proceedings of EMNLP*.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A Smith. 2016. Recurrent Neural Network Grammars. In *Proceedings of NAACL-HLT*.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Janet Hui-wen Hsiao and Richard Shillcock. 2006. Analysis of a Chinese phonetic compound database: Implications for orthographic processing. *Journal of psycholinguistic research*, 35(5):405–426.

Wuwei Lan and Wei Xu. 2018. Neural Network Models for Paraphrase Identification, Semantic Textual Similarity, Natural Language Inference, and Question Answering. In *Proceedings of COLING*.

Jiwei Li, Thang Luong, Dan Jurafsky, and Eduard Hovy. 2015. When Are Tree Structures Necessary for Deep Learning of Representations? In *Proceedings of EMNLP*.

Frederick Liu, Han Lu, Chieh Lo, and Graham Neubig. 2017. Learning Character-level Compositionality with Visual Features. In *Proceedings of ACL*.

Tomohiko Morioka. 2008. CHISE: Character processing based on character ontology. In *International Conference on Large-Scale Knowledge Resources*, pages 148–162. Springer.

Xinlei Shi, Junjie Zhai, Xudong Yang, Zehua Xie, and Chao Liu. 2015. Radical embedding: Delving deeper to Chinese radicals. In *Proceedings of ACL*, volume 2, pages 594–598.

Tzu-Ray Su and Hung-Yi Lee. 2017. Learning Chinese Word Representations From Glyphs Of Characters. In *Proceedings of EMNLP*.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of ACL*.

Yota Toyama, Makoto Miwa, and Yutaka Sasaki. 2017. Utilizing Visual Forms of Japanese Characters for Neural Review Classification. In *Proceedings of IJCNLP*, volume 2, pages 378–382.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of ACL*.

Xingxing Zhang, Liang Lu, and Mirella Lapata. 2016. Top-down tree long short-term memory networks. In *Proceedings of NAACL-HLT*.