Supervised Attention for Sequence-to-Sequence Constituency Parsing

Hidetaka Kamigaito†, Katsuhiko Hayashi,
Tsutomu Hirao and **Masaaki Nagata**
NTT Communication Science Laboratories, NTT Corporation
2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0237, Japan
†kamigaito.hidetaka@lab.ntt.co.jp

Hiroya Takamura and **Manabu Okumura**
Tokyo Institute of Technology, Yokohama, Kanagawa 226-8503, Japan

Abstract

The sequence-to-sequence (Seq2Seq) model has been successfully applied to machine translation (MT). Recently, MT performances were improved by incorporating supervised attention into the model. In this paper, we introduce supervised attention to constituency parsing that can be regarded as another translation task. Evaluation results on the PTB corpus showed that the bracketing F-measure was improved by supervised attention.

1 Introduction

The sequence-to-sequence (Seq2Seq) model has been successfully used in natural language generation tasks such as machine translation (MT) (Bahdanau et al., 2014) and text summarization (Rush et al., 2015). In the Seq2Seq model, attention, which encodes an input sentence by generating an alignment between output and input words, plays an important role. Without the attention mechanism, the performance of the Seq2Seq model degrades significantly (Bahdanau et al., 2014). To improve the alignment quality, Mi et al. (2016), Liu et al. (2016), and Chen et al. (2016) proposed a method that learns attention with the given alignments in a supervised manner, which is called supervised attention. By utilizing supervised attention, the translation quality of MT is improved.

The Seq2Seq model can also be applied to other NLP tasks. We can regard parsing as a translation task from a sentence to an S-expression, and Vinyals et al. (2015) proposed a constituent parsing method based on the Seq2Seq model. Their method achieved the state-of-the-art performance.

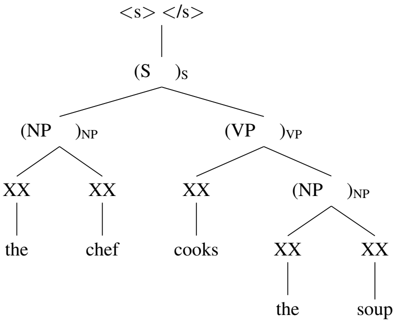


Figure 1: S-expression format for Vinyals et al. (2015)'s Seq2seq constituency parser. The Seq2seq model employs "<s> (S (NP XX XX $)_{NP}$ (VP XX (NP XX XX $)_{NP}$ $)_{VP}$ $)_{S}$ </s>" as output tokens. <s> and </s> are start and end of sentence symbols, respectively.

In their method, based on the alignment between a nonterminal and input words, the attention mechanism has also an important role. However, since the attention is learned in an unsupervised manner, the alignment quality might not be optimal. If we can raise the quality of the alignments, the parsing performance will be improved. Unlike MT, however, the definition of a gold standard alignment is not clear for the parsing tasks.

In this paper, we define several linguistically-motivated annotations between surface words and nonterminals as "gold standard alignments" to enhance the attention mechanism of the constituency parser (Vinyals et al., 2015) by supervised attention. The PTB corpus results showed that our method outperformed Vinyals et al. (2015) by over 1 point in the bracketing F-measure.

Proceedings of the The 8th International Joint Conference on Natural Language Processing, pages 7–12,
Taipei, Taiwan, November 27 – December 1, 2017 ©2017 AFNLP

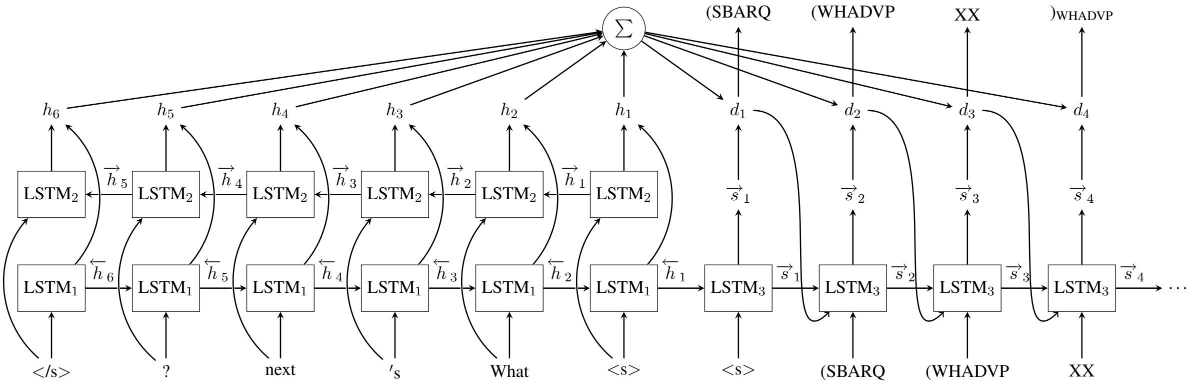


Figure 2: Network structure of our sequence-to-sequence model.

2 Sequence-to-Sequence based Constituency Parser on Supervised Attention Framework

The Seq2Seq constituency parser (Vinyals et al., 2015) predicts nonterminal labels $\mathbf{y} = (y_1, ..., y_m)$, for input words $\mathbf{x} = (x_1, ..., x_n)$, where m and n are respectively the lengths of the word and the label sequences. As shown in Fig. 1, we use normalized labels (Vinyals et al., 2015) in our Seq2Seq model, which consists of encoder and decoder parts. Its overall structure is shown in Fig. 2.

The encoder part employs a 3-layer stacked bidirectional Long Short-Term Memory (LSTM) to encode input sentence $\mathbf{x}$ into a sequence of hidden states $\mathbf{h} = (h_1, ..., h_n)$. Each h_i is a concatenation of forward hidden layer $\overrightarrow{h}_i$ and backward hidden layer $\overleftarrow{h}_i$. $\overleftarrow{h}_1$ is inherited by the decoder as an initial state.

The decoder part employs a 3-layer stacked forward LSTM to encode previously predicted label y_{t-1} into hidden state s_t.

For each time t, with a 2-layer feed-forward neural network r, encoder and decoder hidden layers $\mathbf{h}$ and $\overrightarrow{s}_t$ are used to calculate the attention weight:

$$\alpha_t^i = \frac{exp(r(h_i, \overrightarrow{s}_t))}{\sum_{i'=1}^{n} exp(r(h_{i'}, \overrightarrow{s}_t))}.$$

Using attention weight α_t^i and 1-layer feed-forward neural network u, label probabilities are calculated as follows:

$$P(y_t \mid y_{t-1}, ..., y_1) = \frac{exp(u(d_t)_{v=y_t})}{\sum_{v=1}^{V} exp(u(d_t)_v)},$$
$$d_t = [\sum_{i=1}^{n} \alpha_i^t \cdot h_i, \overrightarrow{s}_t],$$

where V is the label size. Note that d_t and the embedding of label y_t are concatenated and fed to the decoder at time $t+1$.

In a supervised attention framework, attentions are learned from the given alignments. We denote a link on an alignment between y_t and x_i as $a_t^i = 1$ ($a_t^i = 0$ denotes that y_t and x_i are not linked.). Following a previous work (Liu et al., 2016), we adopt a soft constraint to the objective function:

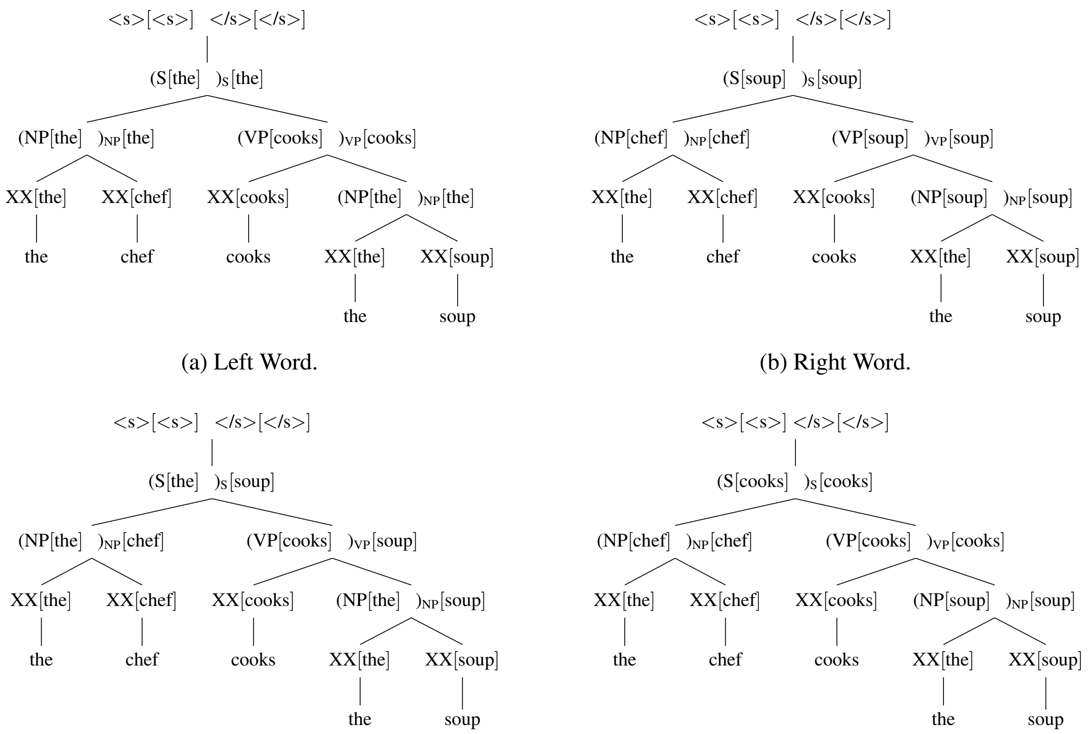
$$-\sum_{t=1}^{n} logP(y_t \mid y_{t-1}, ..., y_0, \mathbf{x})$$
$$-\lambda \times \sum_{i=1}^{n} \sum_{t=1}^{m} a_t^i \times log\alpha_t^i,$$

to jointly learn the attention and output distributions. All our alignments are represented by one-to-many links between input words $\mathbf{x}$ and nonterminals $\mathbf{y}$.

3 Design of our Alignments

In the traditional parsing framework (Hall et al., 2014; Durrett and Klein, 2015), lexical features have been proven to be useful in improving parsing performance. Inspired by previous work, we enhance the attention mechanism utilizing the linguistically-motivated annotations between surface words and nonterminals by supervised attention.

In this paper, we define four types of alignments for supervised attention. The first three methods use the monolexical properties of heads without incurring any inferential costs of lexicalized annotations. Although the last needs manually constructed annotation schemes, it can capture bilexical relationships along dependency arcs. The followings are the details:


<s>[<s>] </s>[</s>]
(S[the])S[the]
(NP[the])NP[the] (VP[cooks])VP[cooks]
XX[the] XX[chef] XX[cooks] (NP[the])NP[the]
the chef cooks XX[the] XX[soup]
the soup
<s>[<s>] </s>[</s>]
(S[soup])S[soup]
(NP[chef])NP[chef] (VP[soup])VP[soup]
XX[the] XX[chef] XX[cooks] (NP[soup])NP[soup]
the chef cooks XX[the] XX[soup]
the soup
<s>[<s>] </s>[</s>]
(S[the])S[soup]
(NP[the])NP[chef] (VP[cooks])VP[soup]
XX[the] XX[chef] XX[cooks] (NP[the])NP[soup]
the chef cooks XX[the] XX[soup]
the soup
<s>[<s>] </s>[</s>]
(S[cooks])S[cooks]
(NP[chef])NP[chef] (VP[cooks])VP[cooks]
XX[the] XX[chef] XX[cooks] (NP[soup])NP[soup]
the chef cooks XX[the] XX[soup]
the soup

(a) Left Word.

(b) Right Word.

(c) Span Word.

(d) Lexical Head.

Figure 3: Example of our alignments. The word in [ ] is linked to each output token.

- **Left word**: In English, the syntactic head of a verb phrase is typically at the beginning of the span. Based on this notion, this method uses the identity of the starting word of a nonterminal span. Figure 3a shows an alignment example where an output token is linked to its leftmost word of the span.

- **Right word**: On the contrary, the syntactic head of a simple English noun phrase is often at the end of the span. The alignment example in Fig. 3b is produced by this method, where an output token is linked to the rightmost word of the span.

- **Span word**: Here, we unify the above two methods. All output tokens are linked to their leftmost word, except the ending bracket tokens, which are linked to their rightmost word. Figure 3c shows an alignment example produced by this method.

- **Lexical head**: Lexicalization (Eisner, 1996; Collins, 1997), which annotates grammar nonterminals with their head words, is useful for resolving the syntactic ambiguities involved by such linguistic phenomena as coordination and PP attachment. As shown in Fig. 3d, this method produces alignments by linking an output token and its head word[1].

4 Experimental Evaluation

4.1 Evaluation Settings

We experimentally evaluated our methods on the English Penn Treebank corpus (PTB), and split the data into three parts: The Wall Street Journal (WSJ) sections 02-21 for training, section 22 for development and section 23 for testing.

In our models, the dimensions of the input word embeddings, the fed label embeddings, the hidden layers, and an attention vector were respectively set to 150, 30, 200, and 200. The LSTM depth was set to 3. Label set L_{con} had a size of 61. The input vocabulary size of PTB was set to 42393. Supervised attention rate λ was set to 1.0. To use entire words as a vocabulary, we integrated word dropout (Iyyer et al., 2015) into our models with smoothing rate 0.8375 (Cross and Huang, 2016). We used dropout layers (Srivastava et al., 2014) to

[1] For head annotations, we used ptbconv 3.0 tool (Yamada and Matsumoto, 2003), which is available from http://www.jaist.ac.jp/h-yamada/.

Setting	**WSJ Section 22**				**WSJ Section 23**			
	P	**R**	**F$_1$**	**AER**	**P**	**R**	**F$_1$**	**AER**
Seq2Seq	88.1	88.0	88.1	–	88.3	87.6	88.0	–
Seq2Seq+random	67.1	66.3	66.7	96.3	66.5	65.5	66.0	96.3
Seq2Seq+first	70.3	69.7	70.0	0.0	69.6	68.7	69.2	0.0
Seq2Seq+last	66.7	66.1	66.4	0.0	66.1	64.8	65.4	0.0
Seq2Seq+head	89.2	88.9	89.1	6.9	89.2	88.1	88.6	6.9
Seq2Seq+left	**89.6**	**89.4**	**89.5**	1.8	89.4	**88.7**	89.0	1.7
Seq2Seq+right	89.2	88.9	89.0	4.7	**89.5**	88.6	**89.1**	4.7
Seq2Seq+span	89.3	89.1	89.2	1.6	89.2	88.4	88.8	1.6
Vinyals et al. (2015) w att†	–	–	88.7	–	–	–	88.3	–
Vinyals et al. (2015) w/o att†	–	–	< 70	–	–	–	< 70	–
Seq2Seq+beam	89.0	88.7	88.8	–	89.1	88.3	88.7	–
Seq2Seq+beam+random	71.0	69.9	70.4	96.3	69.4	68.1	68.7	96.3
Seq2Seq+beam+first	73.9	73.0	73.5	0.0	73.2	71.8	72.5	0.0
Seq2Seq+beam+last	70.5	69.6	70.0	0.0	69.7	68.1	68.9	0.0
Seq2Seq+beam+head	89.6	89.2	89.4	6.9	89.6	88.4	89.0	6.9
Seq2Seq+beam+left	**89.9**	**89.6**	**89.8**	1.8	89.8	**89.0**	89.4	1.7
Seq2Seq+beam+right	89.6	89.2	89.4	4.7	89.7	88.9	89.3	4.7
Seq2Seq+beam+span	89.6	89.4	89.5	1.6	**90.0**	**89.0**	**89.5**	1.6
Seq2Seq+ens(base)	90.5	90.1	90.3	–	90.6	89.6	90.1	–
Seq2Seq+ens(feat)	**91.3**	**90.7**	**91.0**	–	**91.5**	**90.5**	**91.0**	–
Vinyals et al. (2015) w att+ens†	–	–	90.7	–	–	–	90.5	–
Seq2Seq+beam+ens(base)	91.4	90.9	91.1	–	91.5	90.5	91.0	–
Seq2Seq+beam+ens(feat)	**91.9**	**91.4**	**91.7**	–	**92.1**	**91.0**	**91.5**	–

Table 1: Results of parsing evaluation: **Seq2Seq** indicates the Seq2Seq model on a single model with greedy decoding. +beam shows the beam decoding results. +lex, +left, +right and +span respectively show the results on our proposed *lexical head*, *left word*, *right word*, and *span word* alignments. +random, +first, and +last respectively show the results on the alignment of baselines *random*, *first word*, and *last word*. +ens(base) shows the ensemble results of five Seq2Seq models without the given alignments. +ens(feat) shows the ensemble results of a Seq2Seq model without a given alignment and Seq2Seq models with *lexical head*, *left word*, *right word* and *span word* alignments. † denotes the scores reported in the paper.

each LSTM input layer (Vinyals et al., 2015) with a dropout rate of 0.3.

The stochastic gradient descent (SGD) was used to train models on 100 epochs. SGD's learning rate was set to 1.0 in the first 50 epochs. After the first 50 epochs, the learning rate was halved after every 5th epoch. All gradients were averaged in each mini-batch. The maximum mini-batch size was set to 16. The mini-batch order was shuffled at the end of every epoch. The clipping threshold of the gradient was set to 1.0.

We used greedy and beam searches for the decoding. The beam size was set to ten. The decoding was performed on both a single model and five-model ensembles. We used the products of the output probabilities for the ensemble.

All models were written in C++ on Dynet (Neubig et al., 2017).

We compared Seq2Seq models with and without our alignments. To investigate the influence of the supervised attention method itself, we also compared our alignments to the following alignments:

- **Random**: Based on uniform distribution, each output token was randomly linked to at most one input token.
- **First word**: All output tokens were linked to the start of the sentence tokens in the input sentence.
- **Last word**: All output tokens were linked to the end of the sentence tokens in the input sentence.

We evaluated the compared methods using bracketing Precision, Recall and F-measure. We used evalb[2] as a parsing evaluation. We also eval-

[2] http://nlp.cs.nyu.edu/evalb/

uated the learned attention using alignment error rate (AER) (Och and Ney, 2003) on their alignments. Following a previous work (Luong et al., 2015), attention evaluation was conducted on gold output.

4.2 Results

Table 1 shows the results. All our *lexical head*, *left word*, *right word* and *span word* alignments improved bracket F-measure of baseline on every setting. From the +random, +first, and +last results, only supervised attention itself did not improve the parsing performances. Furthermore, each AER indicates that the alignments were correctly learned. These results support our expectation that our alignments improve the parsing performance with Seq2Seq models.

5 Discussion

All of the baseline alignments *random*, *first word* and *last word*, largely degraded the parsing performances. *random* prevented the learning of attention distributions, and *first word* and *last word* fixed the attention distributions. These resemble disable the attention mechanism. Vinyals et al. (2015) reported that the bracket F-measure of Seq2Seq without an attention mechanism is less than 70. Our evaluation results, which are consistent with their score, and it supports our expectation that the attention mechanism is critical for Seq2Seq constituency parsing.

Comparing the results of our proposed alignments in Table 1, even though the bracket F-measure of the *lexical head* is lower than that of the *left word*, *right word* and *span word*, the *lexical head* is the most intuitive alignment. Except for *random*, the AER of *lexical head* is the highest in all the alignments. This means that *lexical head* is difficult to learn on attention distribution. The prediction difficulty may degrade the parsing performances. Our analysis indicates that an alignment which can be easily predicted is suitable for the supervised attention of Seq2Seq constituency parsing.

6 Conclusion

We proposed methods that use traditional parsing features as alignments for the sequence-to-sequence based constituency parser in the supervised attention framework. In our evaluation, the proposed methods improved the bracketing scores on the English Penn Treebank against the baseline methods. These results emphasize, the effectiveness of our alignments in parsing performances.

Acknowledgement

We thank the anonymous reviewers for their careful reading and useful comments.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Wenhu Chen, Evgeny Matusov, Shahram Khadivi, and Jan-Thorsten Peter. 2016. Guided alignment training for topic-aware neural machine translation. *CoRR*, abs/1607.01628.

Michael Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 8th European chapter of the Association for Computational Linguistics*, pages 16–23. Association for Computational Linguistics.

James Cross and Liang Huang. 2016. Span-based constituency parsing with a structure-label system and provably optimal dynamic oracles. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1–11, Austin, Texas. Association for Computational Linguistics.

Greg Durrett and Dan Klein. 2015. Neural crf parsing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 302–312, Beijing, China. Association for Computational Linguistics.

Jason M Eisner. 1996. Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of the 16th conference on Computational linguistics-Volume 1*, pages 340–345. Association for Computational Linguistics.

David Hall, Greg Durrett, and Dan Klein. 2014. Less grammar, more features. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 228–237, Baltimore, Maryland. Association for Computational Linguistics.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1681–1691, Beijing, China. Association for Computational Linguistics.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Neural machine translation with supervised attention. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3093–3102, Osaka, Japan. The COLING 2016 Organizing Committee.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Haitao Mi, Zhiguo Wang, and Abe Ittycheriah. 2016. Supervised attentions for neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2283–2288, Austin, Texas. Association for Computational Linguistics.

Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, et al. 2017. Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980*.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Oriol Vinyals, Ł ukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2773–2781. Curran Associates, Inc.

Hiroyasu Yamada and Yuji Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of IWPT*, volume 3, pages 195–206.