Deterministic Attention for Sequence-to-Sequence Constituent Parsing

Chunpeng Ma,[1*] **Lemao Liu,**[2†] **Akihiro Tamura,**[2] **Tiejun Zhao,**[1] **Eiichiro Sumita**[2]
[1]Machine Intelligence and Translation Laboratory, Harbin Institute of Technology, Harbin, China
[2]ASTREC, National Institute of Information and Communications Technology (NICT), Kyoto, Japan
{cpma, tjzhao}@hit.edu.cn, {lmliu, akihiro.tamura, eiichiro.sumita}@nict.go.jp

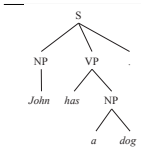
Abstract

The sequence-to-sequence model is proven to be extremely successful in constituent parsing. It relies on one key technique, the probabilistic attention mechanism, to automatically select the context for prediction. Despite its successes, the probabilistic attention model does not always select the most important context. For example, the headword and boundary words of a subtree have been shown to be critical when predicting the constituent label of the subtree, but this contextual information becomes increasingly difficult to learn as the length of the sequence increases. In this study, we proposed a deterministic attention mechanism that deterministically selects the important context and is not affected by the sequence length. We implemented two different instances of this framework. When combined with a novel bottom-up linearization method, our parser demonstrated better performance than that achieved by the sequence-to-sequence parser with probabilistic attention mechanism.

1 Introduction

The sequence-to-sequence model, based on recurrent neural networks (RNNs) (Cho et al. 2014; Sutskever, Vinyals, and Le 2014), provides a unified solution for a number of sequence-to-sequence transformation tasks, including machine translation (Bahdanau, Cho, and Bengio 2014), image captioning (Xu et al. 2015), and grapheme-to-phoneme conversion (Liu et al. 2016a). In particular, Vinyals et al. (2015) applied the sequence-to-sequence model to constituent parsing after linearizing the parse tree as a sequence in a top-down manner (see the first row of Table 1) and viewing it as a specific instance of the sequence-to-sequence transformation task.

The sequence-to-sequence model has notable advantages over task-specific RNN parsing models (Watanabe and Sumita 2015; Dyer et al. 2016), whose architectures are specifically designed for the parsing task. On one hand, the sequence-to-sequence architecture is so general that any improvements made in other sequence-to-sequence tasks can be simultaneously transferred to constituent parsing. On the other hand, it is easy to implement, and even off-the-shelf toolkits can be used with minor modifications, to create a competitive constituent parsing system.



TD	(S (NP XX)$_{NP}$ (VP XX (NP XX XX)$_{NP}$)$_{VP}$ XX)$_{S}$
BU	sh label-NP sh nolabel sh nolabel sh nolabel comb label-NP comb label-VP comb nolabel sh nolabel comb label-S

Table 1: A constituent tree and its linearization. TD = top-down, BU = bottom-up.

The key technique in the sequence-to-sequence constituent parsing is the probabilistic attention mechanism (Bahdanau, Cho, and Bengio 2014; Luong, Pham, and Manning 2015), which can automatically select a context relevant to each time step to yield an accurate prediction. While this has been successful, we argue that this general probabilistic attention mechanism ignores the inherent characteristics of constituent parsing: the boundary and head information that has proven to be critical when identifying phrase structure and predicting its label. There is no guarantee that the probabilistic attention mechanism can select these critical words and use them as the context, which is discussed by Liu et al. (2016b) and Mi, Wang, and Ittycheriah (2016) in the task of machine translation. Consequently, this limits the accuracy of sequence-to-sequence constituent parsing.

Figure 1 illustrates an example of this. When predicting the token "$)_{NP}$" of the noun phrase structure "the fiscal year just ended", the attention mechanism mainly focuses on the verb "ended" and pays almost no attention to the boundary word (symbol) "," and the headword "year". However, these words are very informative in determining the phrase structure: "," indicates that a phrase precedes it, and "year" shows that this phrase should be labeled "NP". The failure to take account of these clues leads to a parsing error. This "pay-attention-to-one-word" phenomenon is quite common. A statistical analysis of the parsing results of Vinyals et al. (2015) on the development set of the Wall Street Journal (WSJ) part of the Penn Treebank (Marcus, Marcinkiewicz, and Santorini 1993) showed that for 91.46% of the sentences, more than 90% of the parser's attention was given

[*]This work was done during the internship of the author at NICT.

[†]Corresponding author.

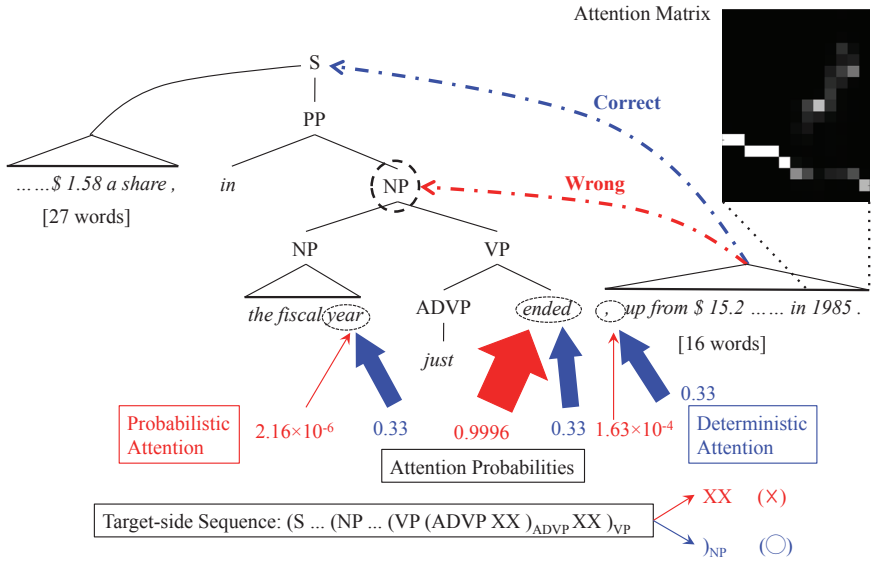


Figure 1: A sentence that the parser of Vinyals et al. (2015) is unable to parse correctly. The parser should predict an "$)_{NP}$", suggesting that the construction of the circled NP structure has been finished and the entire string should be attached to the root "S" node, whereas the parser actually predicts an "XX", causing these words be attached to the circled "NP" node. Because the probabilistic attention model pays too much attention to the verb "ended", the termination of the "NP" cannot be predicted. In contrast, the deterministic attention model takes account of the comma in the queue and the word "year" in the generated tree, which together suggests the endpoint of the "NP". The intensity attributed to a cell in the attention matrix represents the probability that the corresponding word and transition action should be aligned. This disordered attention matrix shows the weakness of the probabilistic attention mechanism when handling long sequences.

to one specific source-side word in more than 80% of the parsing time steps. This unbalance probability indicates that boundary words are not given sufficient attention. Furthermore, as the input sequence becomes longer, the probabilistic attention mechanism finds it increasingly difficult to select the most informative words. This is illustrated by the attention matrix shown in Figure 1, where the selected words are disordered, and the last word is never selected.

To address this issue, we proposed a *deterministic* attention mechanism that is able to select informative words such as boundary words to establish the context, whatever the length of the input sequence. We implemented two different instances of the proposed deterministic attention framework.

However, applying our deterministic attention mechanism directly to the top-down linearized sequence remains challenging. As shown in the target-side sequence of Figure 1, when predicting an "(NP", even if it is established that the left boundary of an NP structure lies between "in" and "the", the right boundary remains unknown since the full NP structure is unavailable during decoding, which makes the parser difficult to determine what words should be paid attention to. To make our deterministic attention mechanism practical, therefore, we utilized a bottom-up linearization method. This is shown in the second row of Table 1 and discussed in more details in Section 3.1. This makes the boundary information available in the course of incremental decoding.

This paper makes the following contributions:

- It proposes a deterministic-attention-based sequence-to-sequence model for constituent parsing, which is implemented on top of a new linearization method. The proposed model guarantees that the selected contexts are informative, while retaining the advantages of the sequence-to-sequence model.
- When tested on the standard WSJ treebank, the deterministic attention model produced significant improvements over probabilistic attention models, delivering a 90.6 F-score on the test set, using ensembling but without requiring pre-training, tri-training, or POS-tagging.
- As a by-product, it is shown that the sequence-to-sequence model can learn the POS-tag information from an analysis of word embeddings. This helps explain why sequence-to-sequence constituent parsing can achieve competitive parsing accuracy without using POS-tag information. This was reported by Vinyals et al. (2015), but without explanation.

Our deterministic attention model is quite general in the sense that it has a range of implementations and can be applied to any bottom-up linearization method, although in this study we only present its two implementations and applied it on top of one specific bottom-up linearization method.

2 Background

The sequence-to-sequence architecture was proposed by Cho et al. (2014) and almost simultaneously by Sutskever, Vinyals, and Le (2014). Given a source sequence

$(x_0, \ldots, x_T)$, to find a target sequence $(y_0, \ldots, y_{T'})$ that maximizes the conditional probability $p(y_0, \ldots, y_{T'} \mid x_0, \ldots, x_T)$, the sequence-to-sequence model uses one RNN to encode the source sequence into a fixed-length context vector c and a second RNN to decode this vector and generate the target sequence. Formally, the probability of the target sequence can be calculated as follows:

$$p(y_0, \ldots, y_{T'} \mid x_0, \ldots, x_T) = \prod_{t=0}^{T'} p(y_t \mid c, y_0, \ldots, y_{t-1}), \tag{1}$$

where each conditional probability can be calculated using a target-side RNN:

$$p(y_t \mid c, y_0, \ldots, y_{t-1}) = g(y_{t-1}, s_t, c). \tag{2}$$

Here, g is a nonlinear function and s_t is the hidden state of the target-side RNN, which can be calculated as $s_t = f(s_{t-1}, y_{t-1}, c)$, where f is also a nonlinear function. The context vector c is also calculated using a source-side RNN:

$$c = q(h_0, \ldots, h_T), \tag{3}$$

where q is a nonlinear function and the hidden state of the source-side RNN is $h_t = f(x_t, h_{t-1})$.

As shown by Cho et al. (2014), the performance deteriorates rapidly as the length of the source sequence increases, which makes it inappropriate to encode source sequences with different lengths into vectors of the same size. To address this, Bahdanau, Cho, and Bengio (2014) introduced an attention mechanism. Instead of encoding the source sequence into a fixed vector c, the attention model uses different c_i-s when calculating the target-side output y_i at time step i. Formally, the context vector c_i is calculated as follows:

$$c_i = \sum_{j=0}^{T} \alpha_{ij} h_j, \tag{4}$$

where α_{ij} is given by:

$$\alpha_{ij} = \frac{\exp(a(s_{i-1}, h_j))}{\sum_{k=0}^{T} \exp(a(s_{i-1}, h_k))}, \tag{5}$$

The function $a(s_{i-1}, h_j)$ can be regarded as the soft alignment between the target-side RNN hidden state s_{i-1} and the source-side RNN hidden state h_j.

Note that in Equation 5, α_{ij} satisfies the Kolmogorov axioms, indicating that it is a valid probability. Thus c_i in Equation 4 can be regarded as the expectation of h_i with respect to the distribution α_{ij}, i.e. $\mathbb{E}_{\alpha_{ij}}[h_j]$. We therefore named this a "probabilistic attention" mechanism.

Vinyals et al. (2015) used this architecture to solve the problem of constituent parsing by linearizing the constituent tree in a simple way, following a depth-first traversal order. This is shown in Table 1. This linearization method converts the constituent tree into a sequence, allowing constituent parsing to be done using the sequence-to-sequence model. They demonstrated that the attention mechanism was able to significantly improve constituent parsing. Using only the WSJ corpus for training, on the test set of the WSJ Treebank, they demonstrated an F-score lower than 70 when the attention mechanism was not used. This increased to 88.3 when the attention mechanism was added.

3 Methodology

As Vinyals et al. (2015) had already demonstrated the effectiveness of the attention mechanism, it is natural to use our deterministic attention method on their framework. However, as noted above, determining what words should be paid attention to is challenging when using top-down linearization. For example, when predicting a label such as "(NP", since the entire structure of this "NP" is unavailable, we cannot determine what the boundary words are, or what the head word is, or what other informative words should be taken into consideration. A bottom-up linearization method avoids this problem, because the tree node is labeled only after all the child nodes have been generated, clarifying the alignment. We discuss this in Section 3.1.

In Section 3.2, we present a general description of the deterministic attention mechanism. In Section 3.3, we present two specific implementation schemes for applying the mechanism to the constituent parsing task.

However, the top-down linearization method also has certain advantages. It can encode global information both in training and decoding using a bidirectional RNN. In contrast, bottom-up linearization can encode global information only in training. Nevertheless, as we were able to show experimentally, the combination of deterministic attention and bottom-up linearization was able to achieve results that were comparable with or superior to those of top-down linearization. This suggested that our new attention mechanism was able to compensate for the inability of the bottom-up linearization method to encode information while decoding.

3.1 Bottom-up Linearization

Currently, the most widely used bottom-up linearization methods (Sagae and Lavie 2005; Zhang and Clark 2009; Zhu et al. 2013) require binarized constituent trees. For a given constituent tree, the lengths of the transition action sequences are usually different due to the unary rules, which is challenging for RNN. Several novel deductive systems have been proposed to solve this problem (Mi and Huang 2015; Cross and Huang 2016a; 2016b). We adopted the deductive system proposed by Cross and Huang (2016b), which does not have these disadvantages. Table 2 illustrates the parsing process of the tree from Table 1, showing the stack every two steps. The label "XX" is a special symbol, representing a state in which the label of a node has not been determined. Note that this linearization method does the binarization implicitly by using the combination of "comb" and "nolabel", and the length of the transition action sequence is determined by the length of the source-side sequence.

Table 3 gives a formal description of this linearization method. The transition state is represented as a tuple $\langle z, \sigma, t \rangle$, where z is the current time step, σ is a stack saving the subtrees generated, t is the index of the last unprocessed word. A subtree in the stack is represented as ${}_iX_j$, where X is the root node of this subtree, and the words in the leaf nodes are $x_i, \ldots, x_{j-1}$ in the source sequence.

This linearization method allows constituent parsing to be done using the sequence-to-sequence architecture with attention mechanism.

Steps	Action		Stack
0 - 1	sh	label-NP	${}_0NP_1$
2 - 3	sh	nolabel	${}_0NP_1$, ${}_1\mathrm{XX}_2$
4 - 5	sh	nolabel	${}_0NP_1$, ${}_1\mathrm{XX}_2$, ${}_2\mathrm{XX}_3$
6 - 7	sh	nolabel	${}_0NP_1$, ${}_1\mathrm{XX}_2$, ${}_2\mathrm{XX}_3$, ${}_3\mathrm{XX}_4$
8 - 9	comb	label-NP	${}_0NP_1$, ${}_1\mathrm{XX}_2$, ${}_2NP_4$
10 - 11	comb	label-VP	${}_0NP_1$, ${}_1VP_4$
12 - 13	comb	nolabel	${}_0\mathrm{XX}_4$
14 - 15	sh	nolabel	${}_0\mathrm{XX}_4$, ${}_4\mathrm{XX}_5$
16 - 17	comb	label-S	${}_0S_5$

Table 2: Parsing of the tree in Table 1 using the bottom-up linearization method.

input:	$x_0, \ldots, x_T$
axiom:	$\langle 0, [], 0 \rangle$
goal:	$\langle 2(2T+1), [{}_0X_{T+1}], T+1 \rangle$
sh	$\dfrac{\langle z, \sigma\|_i X_j, t\rangle}{\langle z+1, \sigma\|_i X_j\|_j \mathrm{XX}_{j+1}, t+1\rangle}$ $j < T$, even z
$comb$	$\dfrac{\langle z, \sigma\|_i X_j\|_j X_k, t\rangle}{\langle z+1, \sigma\|_i \mathrm{XX}_k, t\rangle}$ even z
$label - X$	$\dfrac{\langle z, \sigma\|_i \mathrm{XX}_j, t\rangle}{\langle z+1, \sigma\|_i X_j, t\rangle}$ odd z
$nolabel$	$\dfrac{\langle z, \sigma\|_i \mathrm{XX}_j, t\rangle}{\langle z+1, \sigma\|_i \mathrm{XX}_j, t\rangle}$ $z < (4T+3)$, odd z

Table 3: Formal representation of the bottom-up linearization method. σ can be empty. For sh action, ${}_iX_j$ may also be empty, in which case the stack should be ${}_0\mathrm{XX}_1$ after the sh action is implemented.

3.2 General Description of Deterministic Attention

To address the problems of the probabilistic attention mechanism, we used a novel method to calculate the context vector:

$$c_i = \sum_{j \in \mathcal{D}_i} \mathbf{A}_m h_j. \quad (6)$$

Here, $\mathcal{D}_i$ is a list, saving the indices of the words that should be paid attention to at time step i while generating the target-side sequence. $\mathbf{A}_m$ is a deterministic alignment matrix with the shape $\dim(c) \times \dim(h)$, where $\dim(c)$ and $\dim(h)$ are the dimensions of any c_i-s and any h_j-s, respectively, and $1 \leq m \leq |\mathcal{D}_i|$ denoting the index of the parameter.

Compared with the probabilistic attention mechanism, our deterministic attention mechanism has the following characteristics:

- Instead of calculating the context vector based on all source-side words, it *deterministically* selects a list of indices of words, i.e. $\mathcal{D}_i$, where most of the obviously unrelated source-side words are filtered. This allows the model to focus on the most important words, both improving the decoding accuracy and shortening the decoding time.
- Unlike the α_{ij} (scalars) used in the probabilistic attention model, the parameters $\mathbf{A}_m$ (matrices) are not valid probabilities, allowing the parameters to be adjusted more flexibly. Also, the use of matrices rather than scalars as the parameters significantly increases the capacity of the model.

Note that different from the probabilistic attention mechanism which selects the context based on a probability distribution, the proposed framework deterministically specify $\mathcal{D}_i$ as the context, which resembles the idea of feature engineering in feature-rich parsing. In this sense, the proposed method bridges the gap between the new sequence-to-sequence constituent parsing framework (Vinyals et al. 2015) and the conventional feature-rich parsing framework (Zhu et al. 2013).

Our deterministic attention mechanism is quite general. By using different schemes to determine $\mathcal{D}_i$, different deterministic attention models can be derived. These can be combined to further improve parsing. Note that this description is quite general, whereas specific instances of constituent parsing are presented in Section 3.3.

3.3 Different Schemes of Deterministic Attention for Constituent Parsing

By making different selections of list $\mathcal{D}_i$, the deterministic attention mechanism can be implemented in different ways. In our experiments, two schemes were implemented.

The first deterministic attention scheme (denoted as "datt-bound") followed the intuition that the boundary words of subtrees located in the top-most and the second top-most positions in the stack should be useful for constituent parsing. The first and the last words of the sentence should also be useful, because they encode the complete sentence in two different directions by using the bidirectional RNN. Therefore, $\mathcal{D}_i = (0, r, s, t, T)$, where the stack is $[\sigma|_r X_s|_s X_t]$. For example, at the "4 - 5" time steps in Table 2, $\mathcal{D}_i = (0, 1, 2, 3, 5)$.

The second scheme (denoted as "datt-bound-head") was inspired by the observation made by Collins (1999) and many others that headwords are useful for constituent parsing. As well as using the boundary words, we also applied Collins' head-finding rules to identify the indices of the headwords of the subtrees in the stack, adding them to the list $\mathcal{D}_i$, i.e. $\mathcal{D}_i = (0, r, s, t, h_{rs}, h_{st}, T)$, where h_{st} and h_{rs} are the indices of the headwords of the top-two subtrees of the stack, respectively.

For decoding with both schemes, we not only use a stack-and-buffer as Cross and Huang (2016b) to calculate $\mathcal{D}_i$ for deterministic attention, but also maintain the recurrent hidden units for each partial derivations. In addition, for the second scheme, we heuristically forced the headword to be identical to the headword of the right-most child. The reason is that after a "comb" action has been implemented, the constituent label of the partial tree has not been determined and thus both h_{rs} and h_{st} are unknown.

4 Experiments

4.1 Settings

All experiments were conducted using the WSJ part of the Penn Treebank. Following previous studies such as that of Watanabe and Sumita (2015), we used Sections 2-21, 22 and 23 as the training set, development set and testing set, respectively. We implemented the deterministic attention mechanism based on an open-source sequence-to-sequence

Attention	Configuration	F(dev)	F(test)
Probabilistic	top-down(1)	87.18	87.21
	top-down(5)	89.85	89.74
	bottom-up(1)	86.91	86.52
	bottom-up(5)	89.37	89.15
Deterministic	datt-bound(1)	88.99	88.57
	datt-bound-head(1)	88.36	88.22
	datt-bound(5)	90.53	90.33
	datt-bound-head(5)	90.49	90.32
	datt-bound(5) +datt-bound-head(5)	90.83	90.60
Vinyals et al. (2015) (single)		88.7	88.3
Vinyals et al. (2015) (ensemble)		90.7	90.5

Table 4: Parsing results (F-score) on the development set and test set of WSJ corpus.

toolkit `nematus`[1]. Both the encoder and the decoder modules used the gated recurrent unit (GRU) (Cho et al. 2014) as the hidden unit. We used only one hidden layer with 256 units, set the word embedding dimension as 512, and used dropout for regularization, following the configuration of Vinyals et al. (2015). Pre-training was not implemented. Instead, the word embedding matrix and other network parameters were initialized randomly. For decoding, we used a beam search strategy with a fixed beam size of 10.

4.2 Parsing Results

The parsing results are summarized in Table 4. For probabilistic attention, the results of both linearization methods were reported, whereas for deterministic attention, only the result of the bottom-up linearization was reported. The number in the bracket indicates the number of models used. The models had the same configuration, but different initialization.

The first two rows show the differences attributed to the linearization method. As the top-down linearization method can utilize global information, the F-score was higher than when bottom-up linearization was used.

Using the bottom-up linearization method, with the help of deterministic attention mechanism, the F-score on the test set improves from 86.52 to 88.57, using only one "datt-bound" model. Combining the five models improved the F-score further, to 90.33. This is comparable to the ensemble result reported by Vinyals et al. (2015) (see the last row).

Confounding our expectations, the F-score of "datt-bound-head" was lower than that of "datt-bound", even though the former made use of more information. This suggested that, after implementing a "comb" action, it is inappropriate to force the index of the headword to be the same as the index of the headword of the right-most child. To verify this, we extracted the indices of the headwords of all the tree nodes generated by "comb" action in the training corpus, either using the head-finding rules of Collins, or by simply choosing the headword of the right-most child. The percentage of coincidence in these two cases is only 77.78%, confirming our conjecture.

[1]https://github.com/rsennrich/nematus

Euclidean distance		cosine distance	
Probabilistic	Deterministic	Probabilistic	Deterministic
rumbling	tailored	rumbling	bordering
repaid	controlled	controlled	built
spared	scattered	streaming	tailored
arriving	rumbling	regarded	rumbling
regarded	affiliated	repaid	controlled
affiliated	bordering	spared	completed
trudging	tracked	abates	magnetized
jailed	containing	affiliated	resulting
coupled	Made	jailed	wracked
magnetized	tendered	arriving	associated

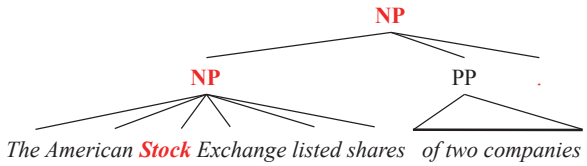
Table 5: Top-10 most similar words to the word "listed" in each attention model.

To enlarge the diversity between the models used for ensembling, we combined the different deterministic attention schemes (5 models for each), obtaining another 0.3 point improvement, getting the result better than the ensemble result of Vinyals et al. (2015).

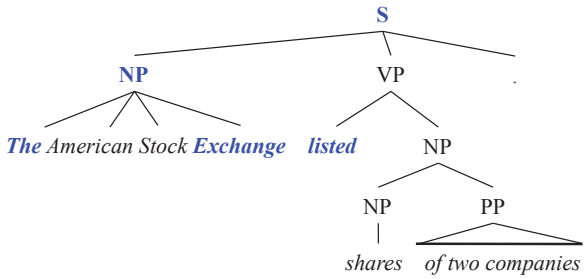
Note that the reported results for the case of probabilistic attention with top-down linearization are lower than the results reported by Vinyals et al. (2015), which is listed at the bottom two rows. The main reason may be that they used three-layer RNNs with long short-term memory (LSTM) units. However, with the help of deterministic attention mechanism, our parser outperforms their parser on both the single-model case (88.57 vs. 88.3, test set) and the ensemble case (90.60 vs. 90.5). Furthermore, we do not need pre-training, which is proven to have the potential to increase the F-score about 0.3 to 0.4 points (Vinyals et al. 2015).

4.3 Case Analysis: Probabilistic Attention vs. Deterministic Attention

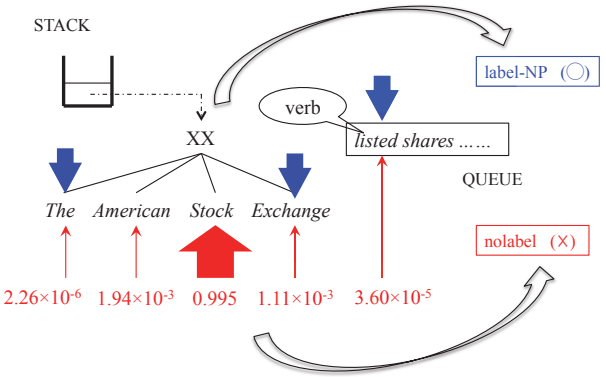
Figure 2(a) and 2(b) show the parsing results of a sentence from the test set of the WSJ treebank, using the probabilistic attention model and the deterministic attention model. Highlighted words/nodes accounted for the difference in parsing results. The parsing result of the deterministic attention model was identical to the golden tree. Figure 2(c) shows what happened when the first parsing error occurred. The probabilistic attention model gave too much attention to the word "Stock" and almost ignored the next word "listed" (a verb), so that the parser attached more words to the current subtree. In contrast, the deterministic attention model used the information provided by the verb "listed", making the correct prediction that the "NP" already comprised enough words. Note that, in both cases, the model had already learned the POS-tag of the word "listed", as shown in Table 5. The top-10 most similar words are all verbs. If the probabilistic attention model could pay more attention to the word "listed", it is quite possible to make a correct prediction. This analysis also explains why sequence-to-sequence constituent parsing can achieve competitive parsing accuracy without POS-tag information, which was firstly observed in Vinyals et al. (2015) but without explanation.



(a) Result using the probabilistic attention.



(b) Result using deterministic attention (identical to the golden tree).



(c) Detailed analysis for the first error

Figure 2: Comparison of parsing results and detailed analysis.

Figure 3 shows the attention matrix learned by the probabilistic attention model, with errors marked. The first error was that analyzed above. In the rectangle area, the alignment was quite disordered, preventing the parser from identifying the correct words to base a prediction on. It can also be seen that the last item (the period) could not be aligned with high probability. In fact, this encoded information on the complete source-side sequence, and was therefore useful for predicting the root node. As a result, the probabilistic attention model predicted that the root node was "NP", whereas the deterministic attention model made the correct prediction that it was "S", because the deterministic attention model used the information of the last word.

4.4 Comparison with State-of-the-art Parsers

Table 6 compares the performance of our parser with those of some state-of-the-art parsers. We can see that our parser is competitive to the feature-rich parsers, which require tedious feature engineering and external toolkits such as a POS tagger for feature extraction. In addition, although one

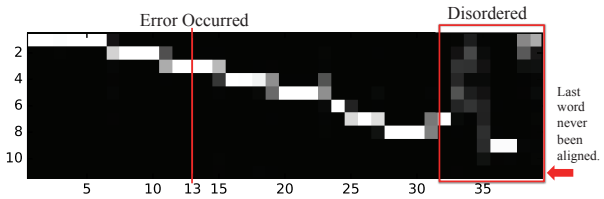


Figure 3: Probabilistic attention matrix. (1) Red line: in the first parsing error, the probabilistic attention model paid (almost) all its attention to the word "Stock" (marked in Figure 2(a)), whereas the deterministic attention model paid attention to three words (marked in Figure 2(b)). (2) Red rectangle: disordered alignment. (3) Red arrow: the last item (i.e., the period, encoding the entire sentence) was not aligned with high probability, causing the prediction error of the root node label (marked in Figure 2(a) and 2(b)).

Config.	Parser	F-score
Feature-rich parser	Petrov and Klein (2007)‡	90.1
	Zhu et al. (2013)‡	90.4
	Charniak and Johnson (2005)†‡	91.0
	Carreras, Collins, and Koo (2008)‡	91.1
	Zhu et al. (2013)‡*	91.3
	Huang (2008)†‡	91.7
	Petrov (2010)‡	91.9
Task-specific neural models	Watanabe and Sumita (2015)	90.7
	Wang, Mi, and Xue (2015)*	90.7
	Durrett and Klein (2015)*	91.1
	Cross and Huang (2016b)‡	91.4
	Dyer et al. (2016)	91.7
	Dyer et al. (2016)†	93.3
Task-free neural models	Vinyals et al. (2015)	90.5
	Vinyals et al. (2015)*	92.8
	This work	90.6

Table 6: Comparison with other state-of-the-art parsers. "General purpose" means that the framework can be used for other tasks without modification. *Semi-supervise. †Reranking. ‡Need POS tagger.

does achieve gains through task-specific neural parsers by specifically designing neural architectures for parsing such as Dyer et al. (2016)[2], our architecture relies on the general sequence-to-sequence framework and thereby it is easily implemented with minor modifications on off-the-shelf toolkits. In particular, our parser performs slightly better than its direct baseline, Vinyals et al. (2015), even if our network is more simple with only one single GRU layer and it does not use additional data for pre-training. The outstanding performance of Vinyals et al. (2015) with the help of semi-supervised learning implies the potential of our parser, which remains as the future work.

5 Conclusions and Future Work

In this study, we proposed a new attention mechanism that aligns the source-side words and target-side words in a deter-

[2]These excellent results were updated by Dyer et al. (2016) after our submission, see https://arxiv.org/pdf/1602.07776v4.pdf.

ministic way, and applied it to the task of constituent parsing. Two different schemes were used, and their performance was compared. We demonstrated experimentally that the deterministic attention model was able to outperform conventional probabilistic attention models. When the two attention schemes were combined, the results were comparable to those of state-of-the-art neural-network-based constituent parsers. We also analyzed differences between the deterministic attention mechanism and the probabilistic mechanism using a specific case, demonstrating the advantages of the deterministic attention model when undertaking constituent parsing.

In the future work, we will use pre-training and semi-supervised learning methods to further improve the performance of constituent parsing. We will also apply our approach to dependency parsing, character-based parsing, and other tasks that can be addressed using the sequence-to-sequence model.

6 Acknowledgments

We are grateful to the anonymous reviewers for their insightful comments and suggestions. This work was done during the internship of Chunpeng Ma at NICT. Tiejun Zhao is supported by the National Natural Science Foundation of China (NSFC) via grant 91520204 and State High-Tech Development Plan of China (863 program) via grant 2015AA015405.

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Carreras, X.; Collins, M.; and Koo, T. 2008. Tag, dynamic programming, and the perceptron for efficient, feature-rich parsing. In *Proceedings of CoNLL*, 9–16.

Charniak, E., and Johnson, M. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of ACL*, 173–180.

Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of EMNLP*, 1724–1734.

Collins, M. 1999. *Head-driven Statistical Models for Natural Language Parsing*. Ph.D. Dissertation, University of Pennsylvania.

Cross, J., and Huang, L. 2016a. Incremental parsing with minimal features using bi-directional lstm. In *Proceedings of ACL*, volume 2, 32–37.

Cross, J., and Huang, L. 2016b. Span-based constituency parsing with a structure-label system and dynamic oracles. In *Proceedings of EMNLP*, 1–11.

Durrett, G., and Klein, D. 2015. Neural crf parsing. In *Proceedings of ACL-IJCNLP*, volume 1, 302–312.

Dyer, C.; Kuncoro, A.; Ballesteros, M.; and Smith, N. A. 2016. Recurrent neural network grammars. In *Proceedings of HLT-NAACL*, 199–209.

Huang, L. 2008. Forest reranking: Discriminative parsing with non-local features. In *Proceedings of ACL-HLT*, 586–594.

Liu, L.; Finch, A.; Utiyama, M.; and Sumita, E. 2016a. Agreement on target-bidirectional lstms for sequence-to-sequence learning. In *Proceedings of AAAI*, 2630–2637.

Liu, L.; Utiyama, M.; Finch, A.; and Sumita, E. 2016b. Neural machine translation with supervised attention. In *Proceedings of COLING*.

Luong, T.; Pham, H.; and Manning, C. D. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of EMNLP*, 1412–1421.

Marcus, M. P.; Marcinkiewicz, M. A.; and Santorini, B. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics* 19(2):313–330.

Mi, H., and Huang, L. 2015. Shift-reduce constituency parsing with dynamic programming and pos tag lattice. In *Proceedings of NAACL-HLT*, 1030–1035.

Mi, H.; Wang, Z.; and Ittycheriah, A. 2016. Supervised attentions for neural machine translation. In *Proceedings of EMNLP*, 2283–2288.

Petrov, S., and Klein, D. 2007. Improved inference for unlexicalized parsing. In *Proceedings of HLT-NAACL*, 404–411.

Petrov, S. 2010. Products of random latent variable grammars. In *Proceedings of HLT-NAACL*, 19–27.

Sagae, K., and Lavie, A. 2005. A classifier-based parser with linear run-time complexity. In *Proceedings of IWPT*, 125–132.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Proceedings of NIPS*, 3104–3112.

Vinyals, O.; Kaiser, Ł.; Koo, T.; Petrov, S.; Sutskever, I.; and Hinton, G. 2015. Grammar as a foreign language. In *Proceedings of NIPS*, 2773–2781.

Wang, Z.; Mi, H.; and Xue, N. 2015. Feature optimization for constituent parsing via neural networks. In *Proceedings of ACL-IJCNLP*, volume 1, 1138–1147.

Watanabe, T., and Sumita, E. 2015. Transition-based neural constituent parsing. In *Proceedings of ACL*, volume 1, 1169–1179.

Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhutdinov, R.; Zemel, R. S.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of ICML*, 2048–2057.

Zhang, Y., and Clark, S. 2009. Transition-based parsing of the chinese treebank using a global discriminative model. In *Proceedings of the 11th International Conference on Parsing Technologies*, 162–171.

Zhu, M.; Zhang, Y.; Chen, W.; Zhang, M.; and Zhu, J. 2013. Fast and accurate shift-reduce constituent parsing. In *Proceedings of ACL-IJCNLP*, volume 1, 434–443.