CKY-based Convolutional Attention for Neural Machine Translation

Taiki Watanabe and **Akihiro Tamura** and **Takashi Ninomiya**
Ehime University
3 Bunkyo-cho, Matsuyama, Ehime, JAPAN
{t_watanabe@ai.cs, tamura@cs, ninomiya@cs}.ehime-u.ac.jp

Abstract

This paper proposes a new attention mechanism for neural machine translation (NMT) based on convolutional neural networks (CNNs), which is inspired by the CKY algorithm. The proposed attention represents every possible combination of source words (e.g., phrases and structures) through CNNs, which imitates the CKY table in the algorithm. NMT, incorporating the proposed attention, decodes a target sentence on the basis of the attention scores of the hidden states of CNNs. The proposed attention enables NMT to capture alignments from underlying structures of a source sentence without sentence parsing. The evaluations on the Asian Scientific Paper Excerpt Corpus (ASPEC) English-Japanese translation task show that the proposed attention gains 0.66 points in BLEU.

1 Introduction

Recently, neural machine translation (NMT) based on neural networks (NNs) is known to provide both high-precision and human-like translation through its simple architecture. In NMT, the encoder-decoder model, which is intensively studied, converts a source-language sentence into a fixed-length vector and then generates a target-language sentence from the vector by using recurrent NNs (RNNs) with gated recurrent units (GRUs) (Cho et al., 2014a) or long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997; Gers et al., 2000; Sutskever et al., 2014). An attention-based NMT (ANMT) is one of the state-of-the-art technologies for MT, which is an extension of the encoder-decoder model and provides highly accurate translation (Luong et al., 2015; Dzmitry et al., 2015). ANMT is a method of translation in which the decoder generates a target-language sentence, referring to the history of the encoder's hidden layer state.

The encoder-decoder model has also been extended to syntax-based NMT, which utilizes structures of source sentences, target sentences, or both. In particular, Eriguchi et al. (2016b) have shown that a source-side structure (i.e., constituent trees of source sentences) are useful for NMT on the English-Japanese translation. However, syntax-based NMT requires sentence parsing in advance.

This paper proposes a new attention mechanism for NMT based on convolutional neural networks (CNNs) to leverage the structures of source sentences in NMT without parsing. In the parsing field, the CKY algorithm (Kasami, 1965; Younger, 1967) parses a sentence in a bottom-up manner through the CKY table, which efficiently considers all possible combinations of words and represents the structure of the sentence through dynamic programming. Inspired by the algorithm, we incorporate CNNs that imitate the CKY table into the attention mechanism of ANMT. In particular, the proposed attention constructs CNNs in the same order as the calculation procedures in the CKY table, and then ANMT decodes a target sentence by referring to each state of the hidden layers of CNNs, which corresponds to each cell in the CKY table. The proposed attention enables the ANMT model to capture underlying structures of a source sentence that are useful for a prediction of each target word, without sentence parsing in advance.

The evaluations on the ASPEC English-Japanese translation task (Nakazawa et al., 2016) show that the proposed attention gains 0.66 points in BLEU. Furthermore, they show that our attention can capture structural alignments (e.g., align-

Proceedings of the The 8th International Joint Conference on Natural Language Processing, pages 1–6,
Taipei, Taiwan, November 27 – December 1, 2017

ment to a case structure), which is not a word-level alignment.

There are several previous studies on NMT using CNNs (Kalchbrenner and Blunsom, 2013; Cho et al., 2014b; Lamb and Xie, 2016; Kalchbrenner et al., 2016). Their models consist of serially connected multi-layer CNNs for encoders or decoders, similar to image recognition CNNs for 1D image processing. Therefore, their models do not have any direct mechanisms for dealing with the connections between phrases/words in long distance. Our model adopts CKY-based connections between multi-layer CNNs, which enables the NMT to calculate direct connections between phrases/words in encoders, and the attention mechanism enables the NMT to capture structural alignment between decoders and encoders[1].

2 Attention-based NMT (ANMT)

ANMT (Luong et al., 2015; Dzmitry et al., 2015) is an extension of the encoder-decoder model (Sutskever et al., 2014; Cho et al., 2014a). The model uses its RNN encoder to convert a source-language sentence into a fixed-length vector and then uses its RNN decoder to generate a target-language sentence from the vector.

We used a bi-directional two-layer LSTM network as the encoder. Given a source-language sentence $\mathbf{x} = x_1, x_2, \cdots, x_T$, the encoder represents the i-th word, x_i, as a d-dimensional vector, v_i, by a word embedding layer. The encoder then computes the hidden state of v_i, h_i, as follows:

$$\overrightarrow{h_i^{(1)}} = LSTM^{(1)}(v_i), \quad (1)$$

$$\overleftarrow{h_i^{(1)}} = LSTM^{(1)}(v_i), \quad (2)$$

$$\overrightarrow{h_i^{(2)}} = LSTM^{(2)}(\overrightarrow{h_i^{(1)}}) + \overrightarrow{h_i^{(1)}}, \quad (3)$$

$$\overleftarrow{h_i^{(2)}} = LSTM^{(2)}(\overleftarrow{h_i^{(1)}}) + \overleftarrow{h_i^{(1)}}, \quad (4)$$

$$h_i = \overrightarrow{h_i^{(2)}} + \overleftarrow{h_i^{(2)}}, \quad (5)$$

where $\rightarrow$ and $\leftarrow$ indicate the forward direction (i.e., from the beginning to the end of a sentence) and the reverse direction, respectively. $LSTM^{(1)}$ and $LSTM^{(2)}$ represent the first- and second-layer LSTM encoders, respectively. The dimensions of $\overleftarrow{h_i^{(1)}}$, $\overrightarrow{h_i^{(1)}}$, $\overleftarrow{h_i^{(2)}}$, $\overrightarrow{h_i^{(2)}}$, and h_i are d.

[1] In a preliminary experiment, we directly applied a CNN to the encoder of the encoder-decoder model. However, the method (BLEU: 25.91) does not outperform our proposed method (BLEU: 26.75).

In ANMT, the decoder generates a target-language sentence, referring to the hidden layer's states of the LSTM encoder, h_i. The attention mechanism explained below is called *global attention (dot)* (Luong et al., 2015). We used a two-layer LSTM network as the decoder. The initial states of the first- and second-layer LSTM decoders are initialized as the states of the first- and second-layer LSTM encoders in the reverse direction, respectively.

Each state of the hidden layers of LSTM decoders, $s_j^{(1)}$ and $s_j^{(2)}$, is calculated by

$$s_j^{(1)} = LSTM^{(1)}([w_{j-1}; \hat{s}_{j-1}]), \quad (6)$$

$$s_j^{(2)} = LSTM^{(2)}(s_j^{(1)}), \quad (7)$$

where w_{j-1} indicates word embedding of the output word y_{j-1}, ';' represents a concatenation of matrices, and $\hat{s}_{j-1}$ is an attentional vector used for generating the output word y_{j-1}, which is explained below[2].

The dimensions of w_{j-1} and $\hat{s}_{j-1}$ are d. The attention score $\alpha_j(i)$ is calculated as follows:

$$\alpha_j(i) = \frac{exp(h_i \cdot s_j^{(2)})}{\sum_{k=1}^{T} exp(h_k \cdot s_j^{(2)})}. \quad (8)$$

The context vector c_j for generating a target-language sentence is calculated by

$$c_j = \sum_{i=1}^{T} \alpha_j(i) h_i. \quad (9)$$

The attentional vector $\hat{s}_j$ is calculated by using the context vector as follows:

$$\hat{s}_j = tanh(W_c[s_j^{(2)}; c_j]), \quad (10)$$

and then using the state of this hidden layer, the probability of the output word y_j is given by

$$p(y_j | y_{<j}, \mathbf{x}) = softmax(W_s \hat{s}_j), \quad (11)$$

where W_c and W_s represent weight matrices[3].

3 NMT with CKY-based Convolutional Attention

Figure 1 shows the overall structure of the proposed attention. In the proposed attention, a gen-

[2] Providing an attentional vector as inputs to the LSTM in the next time step is called input feeding (Luong et al., 2015).

[3] In our experiments, target sentences are generated by the greedy algorithm on the basis of output probabilities.

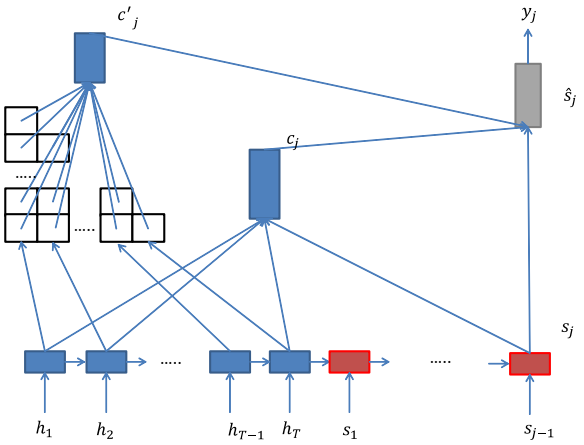

Figure 1: Overall View of CKY-based Attention

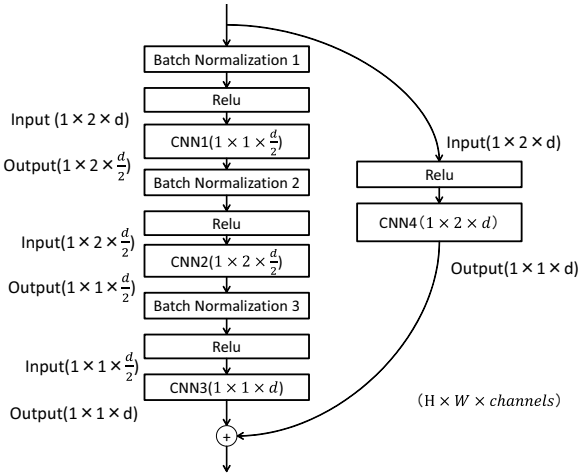

Figure 2: Deduction Unit in CKY-based Attention

erative rule in the CKY algorithm is imitated by the network structure shown in Figure 2. We call the network as the *Deduction Unit (DU)*. In a DU, four types of CNNs are connected by a residual connection[4]. In Figure 2, the size of filters and the number of output channels for each CNN are shown in a parenthesis. In particular, the filter sizes of CNN1, CNN2, CNN3, and CNN4, are 1×1, 1×2, 1×1, and 1×2, and their channel numbers are $\frac{d}{2}$, $\frac{d}{2}$, d, and d, respectively. Each DU receives d-dimensional vectors (states) of two cells in a CKY table and computes a d-dimensional vector for an upper-level cell, which corresponds to a generation rule in the CKY algorithm. By using DUs, the state of each cell in a CKY table is induced by folding the states of lower-level cells in the same order as the calculation procedures in the CKY algorithm. We call the network for this overall procedure as the *CKY-CNN*. We hereafter denote the state of the j-th cell in the i-th CKY-CNN layer as $h_{i,j}^{(cky)}$. Note that the states of the first-layer of the CKY-CNN (i.e., $\mathbf{h_1^{(cky)}} = (h_{1,1}^{(cky)}, ..., h_{1,T}^{(cky)})$) are set to the states of the LSTM encoder (i.e., $\mathbf{h} = (h_1, ..., h_T)$). In the CKY-CNN, the state of a cell is induced from multiple candidates of outputs from DUs, similar to the CKY algorithm. Specifically, the state of a cell is set to the output vector with the highest sum of values of all dimensions as follows:

$$h_{i,j}^{(cky)} = Max_{1 \leq k \leq i-1} DU(h_{k,j}^{(cky)}, h_{i-k,j+k}^{(cky)}) \quad (12)$$

[4]Through a preliminary experiment, we confirmed that a simple DU composed of one type of CNN did not work well. Therefore, we have improved the DU in reference to (He et al., 2016).

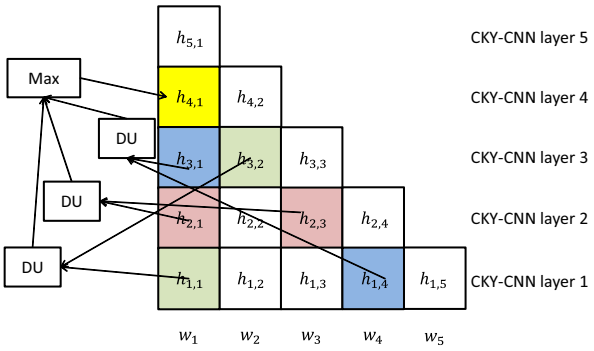

Figure 3: An Example of Max-pooling with CKY-CNN

Figure 3 shows an example of convolutions in the CKY-CNN, highlighting the process of generating the state of the yellow cell. In this process, three DUs generate vectors based on the states of the two blue cells, those of the two red cells, and those of the two green cells, respectively. The vector with the highest sum of vector elements is then set to the state of the yellow cell. Through the CKY-CNN, the states of the cells in a CKY table ($\mathbf{h}^{(cky)}$) are obtained.

NMT with the CKY-based convolutional attention decodes a target sentence by referring to the states of the hidden layers of the CKY-CNN in addition to the states of the hidden layer of the LSTM encoder. The alignment scores are calculated as follows:

$$\alpha'(i,j) = \frac{exp(h_i \cdot s_j^{(2)})}{\sum_{k=1}^{T} exp(h_k \cdot s_j^{(2)}) + \sum_{k=1}^{T} \sum_{l=1}^{T-k+1} exp(h_{k,l}^{(cky)} \cdot s_j^{(2)})}, \quad (13)$$

$$\alpha''(i_1, i_2, j) = \frac{exp(h^{(cky)}_{i_1,i_2} \cdot s^{(2)}_j)}{\sum_{k=1}^{T} exp(h_k \cdot s^{(2)}_j) + \sum_{k=1}^{T} \sum_{l=1}^{T-k+1} exp(h^{(cky)}_{k,l} \cdot s^{(2)}_j)}. \quad (14)$$

Note that $s^{(2)}_j$ is the hidden layer's state of the second-layer LSTM encoder (see Section 2). The context vector c'_j for CKY-CNN is calculated by

$$c'_j = \sum_{k=1}^{T} \alpha'(k, j) h_k + \sum_{k=1}^{T} \sum_{l=1}^{T-k+1} \alpha''(k, l, j) h^{(cky)}_{k,l}. \quad (15)$$

$\hat{s}_j$ is calculated on the basis of the context vector of the LSTM encoder (c_j), which is defined in Section 2, and that of the CKY-CNN (c'_j) as follows:

$$\hat{s}_j = tanh(\hat{W}[s^{(2)}_j; c_j; c'_j]), \quad (16)$$

where $\hat{W} \in R^{d \times 3d}$ is a weight matrix. By applying the softmax function to the $\hat{s}_j$ in the same way as in the conventional ANMT (see Section 2), the encoder predicts the j-th target word.

4 Experiments

4.1 Settings

We used Asian Scientific Paper Excerpt Corpus (ASPEC)'s English-Japanese corpus[5] in this experiment. We used the Moses decoder for word segmentation of the English corpus and Kytea (Neubig et al., 2011) for the Japanese corpus. For each corpus, all characters are lowercased. We used the first 100,000 sentences (< 50 words) for training, 1,790 sentences for parameter tuning, and 1,812 sentences for testing. The words that appeared less than twice in the training data were replaced with the special symbol UNK.

The number of dimensions of word vectors and hidden layers was 256. Adam (Kingsma and Ba, 2014) was used for learning each parameter, and the initial values of the parameters were set to $\alpha = 0.01$, $\beta_1 = 0.9$, and $\beta_2 = 0.99$. The learning rate was halved after 9 and 12 epochs. A gradient clipping technique was used with a clipping value of 3.0, following (Eriguchi et al., 2016a). We used dropout (Srivastava et al., 2014) and weight decay to prevent over-fitting. The dropout ratio for LSTMs was 0.2, that for the CNN was 0.3, and the weight decay coefficient was 10^{-6}.

Table 1: Evaluation Results

	BLEU (%)
Baseline Model	26.09
Proposed Model	26.75

4.2 Results

We compared the NMT with the CKY-based convolutional attention (see Section 3) with the NMT with the conventional attention (see Section 2) to confirm the effectiveness of the proposed CKY-based attention. The only difference between the baseline and the proposed model is their attention mechanisms. Table 1 shows the translation performance by BLEU (Papineni et al., 2002). For reference, we obtained a 18.69% BLEU score using the Moses phrase-based statistical machine translation system (Koehn et al., 2007) with the default settings.

Table 1 shows that the proposed model outperforms the baseline model, which indicates that the proposed attention is useful for NMT.

Figure 4 shows the attention scores of an instance in the test data. The deeper color of a cell represents a higher attention score. The vertical axis represents a source sentence. In Figure 4, the test sentence is "finally, this paper describes the recent trend and problems in this field .". The horizontal axis indicates the depth of the CKY-CNN. Note that an attention score of the first layer of the CKY-CNN corresponds to an attention score of the hidden layer of the LSTM. Figure 4 shows that for the words whose alignments are clearly defined such as content words (e.g., "最後 (finally)", "分野 (field)", "述べ (describe)"), high alignment scores are located in the first layer. On the other hand, for the words whose alignments are not clearly defined such as function words (e.g., "に", "け", "る"), high alignment scores are located at a deeper layer. The Japanese word "に" shows a case structure, and "け" and "る" are parts of the Japanese preposition "おける (in)". This indicates that while the conventional attention finds word-level alignments, the proposed attention captures structural alignments.

[5] http://orchid.kuee.kyoto-u.ac.jp/WAT/WAT2015/index.html

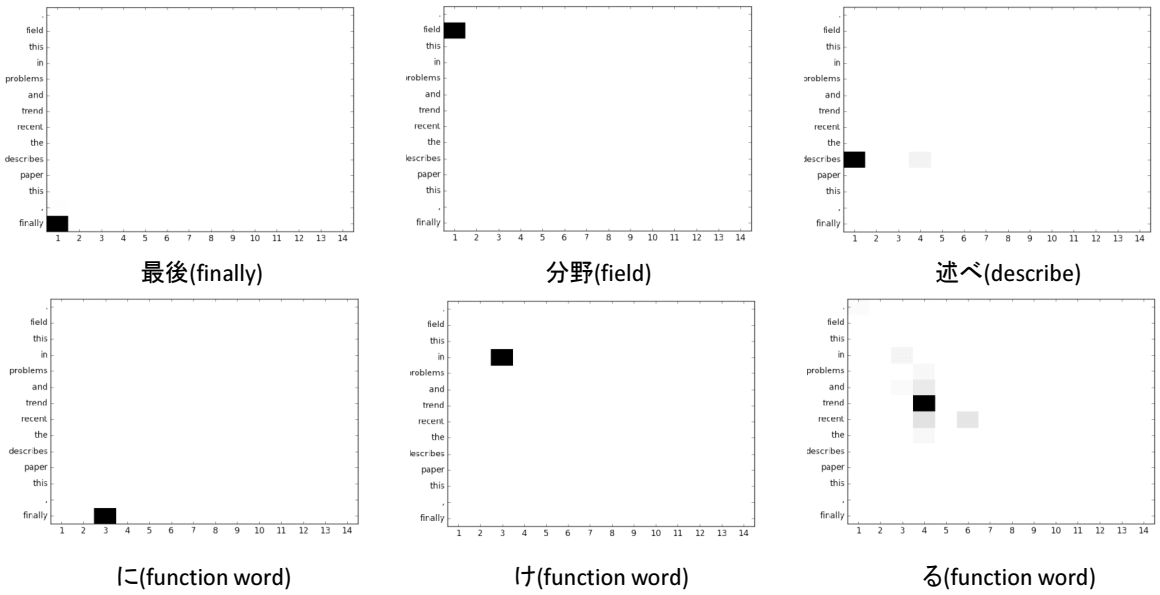

Figure 4: Examples of Attention Scores

5 Conclusions

This paper proposed an attention mechanism for NMT based on CNNs, which imitates the CKY algorithm. The evaluations on the ASPEC English-Japanese translation task showed that the proposed attention gained 0.66 points in BLEU and captured structural alignments, which could not be captured by a conventional attention mechanism. The proposed model consumes excessive amounts of memory because the proposed model keeps hidden states of all cells in a CKY table. In future, we would like to improve the proposed attention in terms of memory consumption, and then verify the effectiveness of the proposed attention for larger datasets.

Acknowledgements

This work was supported by JSPS Grants-in-Aid for Scientific Research Grant Number 25280084. We are grateful to Shinsuke Mori, Kazuma Hashimoto, and Akiko Eriguchi for their technical advice to this work.

References

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014a. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014b. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Bahdanau Dzmitry, Cho KyungHyun, and Bengio Yoshua. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations.*

Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. 2016a. Character-based decoding in tree-to-sequence attention-based neural machine translation. In *Proceedings of the 3rd workshop on Asian Translation*, pages 175–183.

Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. 2016b. Tree-to-sequence attentional neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 823–833.

Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. 2000. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on*

computer vision and pattern recognition, pages 770–778.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, page 413.

Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. 2016. Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099*.

Tadao Kasami. 1965. An efficient recognition and syntax algorithm for context-free languages. Technical Report AFCRL-65-758.

Diederik Kingsma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *5th International Conference on Learning Representations*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the Association for Computational Linguistics on interactive poster and demonstration sessions*, pages 177–180.

Andrew Lamb and Michael Xie. 2016. Convolutional encoders for neural machine translation. *arXiv preprint arXiv:1611.02344*.

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.

Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. 2016. Aspec: Asian scientific paper excerpt corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 2204–2208.

Graham Neubig, Yosuke Nakata, and Shinsuke Mori. 2011. Pointwise prediction for robust, adaptable japanese morphological analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 529–533.

Kishore Papineni, Salam Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

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

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Daniel H. Younger. 1967. Recognition and parsing of context-free languages in time n^3. *Information and Control*, 2(10):189–208.