Deep Inside-outside Recursive Autoencoder with All-span Objective

Ruyue Hong[†], **Jiong Cai**[†◇], **Kewei Tu**[†◇*]
[†]School of Information Science and Technology, ShanghaiTech University
[◇]Shanghai Engineering Research Center of Intelligent Vision and Imaging
Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences
University of Chinese Academy of Sciences
{hongry,caijiong,tukw}@shanghaitech.edu.cn

Abstract

Deep inside-outside recursive autoencoder (DIORA) is a neural-based model designed for unsupervised constituency parsing. During its forward computation, it provides phrase and contextual representations for all spans in the input sentence. By utilizing the contextual representation of each leaf-level span, the span of length 1, to reconstruct the word inside the span, the model is trained without labeled data. In this work, we extend the training objective of DIORA by making use of all spans instead of only leaf-level spans. We test our new training objective on datasets of two languages: English and Japanese, and empirically show that our method achieves improvement in parsing accuracy over the original DIORA.

1 Introduction

Constituency parsing produces constituent parse trees that can be useful for downstream tasks, such as text classification (Tai et al., 2015), machine translation (Eriguchi et al., 2017; Wang et al., 2018) and semantic role labelling (Gildea and Palmer, 2002; He et al., 2017). Although supervised neural methods have very high accuracy, they require annotated data which can be very limited for low-resource languages and domains. Unsupervised constituency parsing is a task aiming to learn a constituency parser from text without annotated parse trees. There is a recent trend of solving this task with neural approaches (Shen et al., 2018; Kim et al., 2019a; Shen et al., 2019; Kim et al., 2019b; Drozdov et al., 2019). DIORA (Drozdov et al., 2019) is one of these models. It mimics the inside-outside algorithm (Baker, 1979) to build an inside chart and an outside chart for each input sentence. Each cell in these charts represents a representation of the inside text or the outside context of a span of the sentence. DIORA is trained by using the contextual representation of every single word to predict the word itself, which can be seen as a reconstruction procedure of the input sentence.

The original training objective of DIORA only takes the n leaf-level spans into account for an n-word sentence. Since such a sentence has $(n^2 + n)/2$ spans in total, we suppose that utilizing the rest of the spans during training can help the model learn better. Thus we propose a new training objective for DIORA that takes into account the reconstruction of all spans. We design weighting strategies to balance the influence from spans and sentences of different lengths in the new objective function. In the experiments, we train and test DIORA with our new objective function on datasets of two languages, English and Japanese. For both languages, we show empirically that the new training objective improves the performance of unsupervised constituency parsing.

2 Approach

2.1 DIORA

DIORA, the deep inside-outside recursive autoencoder, incorporates the inside-outside algorithm into a latent tree chart parser. During the bottom-up inside pass, it computes the inside vector of each span in the input sentence, which captures the phrase information of the inner content in the span. During the top-down outside pass, the model computes the outside vector of each span, which models the contextual

* Kewei Tu is the corresponding author.

information of the span. It also computes an inside score and an outside score for each span, which intuitively evaluate the constituency of the span based on its content and context. For all the model details of DIORA, we refer the reader to (Drozdov et al., 2019). For a span i, denote the inside score by $\bar{e}(i)$, inside vector by $\bar{a}(i)$, outside score by $\bar{f}(i)$ and outside vector by $\bar{b}(i)$.

2.2 Original Objective Function

DIORA is trained by reconstructing the word in each leaf-level span with its outside contextual vector. Drozdov et al.(2019) proposed two objective functions, the max-margin loss and the cross-entropy loss. For sentence x, the max-margin loss is:

$$L_x = \sum_{i \in \Phi(x)} \sum_{i^* \in \mathcal{N}(i)} \max(0, 1 - \bar{b}(i) \cdot \bar{a}(i) + \bar{b}(i) \cdot \bar{a}(i^*))$$

and the cross-entropy loss is:

$$L_x = -\sum_{i \in \Phi(x)}^{l} \log \left(\frac{\exp(\bar{b}(i) \cdot \bar{a}(i))}{Z^*(i) + \exp(\bar{b}(i) \cdot \bar{a}(i))} \right)$$

$$Z^*(i) = \sum_{i^* \in \mathcal{N}(i)} \exp(\bar{b}(i) \cdot \bar{a}(i^*))$$

where $\Phi(x)$ denotes the set of leaf-spans in the sentence x, $\mathcal{N}(i)$ denotes the set of negative samples for span i. In our experiments, we use the cross-entropy loss by default.

2.3 All-span Objective Function

The original objective function of DIORA is constrained to utilize leaf-level information only. However, higher-level spans also embody meaningful information which could be utilized. Our goal is thus to integrate all-level spans into the objective function.

Instead of treating all spans evenly, we prefer the spans that are more likely to be constituents within the ground-truth parse tree, because it makes more sense to predict a constituent from its context in comparison with a distituent. The inside and outside scores in DIORA can be interpreted as a measure of constituency of spans. In particular, Drozdov et al. (2019) use the inside scores as input to a CYK parser to produce the final parse tree. Therefore, we assign weights to spans in our objective function based on their inside and outside scores.

$$L_x = \sum_{i \in \Pi(x)} w_i \cdot L_i$$

where $\Pi(x)$ denotes the set of all spans in the sentence x. We consider two types of weight computation, w_i^α and w_i^β, based on inside and outside scores respectively. We notice that in the model trained with the original objective, the inside and outside scores are influenced by span lengths and sentence lengths (visually depicted in Fig. 1). In brief, inside scores are positively correlated with span length while outside scores are negatively correlated with span length but positively correlated with sentence length. In order for our objective function to balance the weights of spans with different span lengths and sentence lengths, we define the two types of weights of span i as follows.

$$w_i^\alpha = \exp(\frac{\bar{e}(i)}{m_i}) \qquad w_i^\beta = \exp(\frac{\bar{f}(i)}{(n - m_i + 1) \cdot n})$$

where m_i is the length of span i, while n is the length of the sentence. We apply exponentiation because inside and outside scores may be negative.

To balance the influence of different sentences, we also assign a weight to each sentence in the objective function over the full training corpus, such that the influence of a sentence is proportional to its length:

$$L_x = n \cdot \sum_{i \in \Pi(x)} \frac{w_i}{\sum_{j \in \Pi(x)} w_j} \cdot L_i$$

For leaf-level spans, the negative examples are still sampled according to word frequency as in (Drozdov et al., 2019). For higher-level spans, if we randomly sample spans from the training corpus as negative examples, then during training we have to either cache the inside vectors of all the spans from all the sentences or recompute the inside vectors of the sampled spans at each gradient descent step, both of which are computationally very expensive. Therefore, we choose to sample negative examples from all the spans within the training batch that each training sentence belongs to.

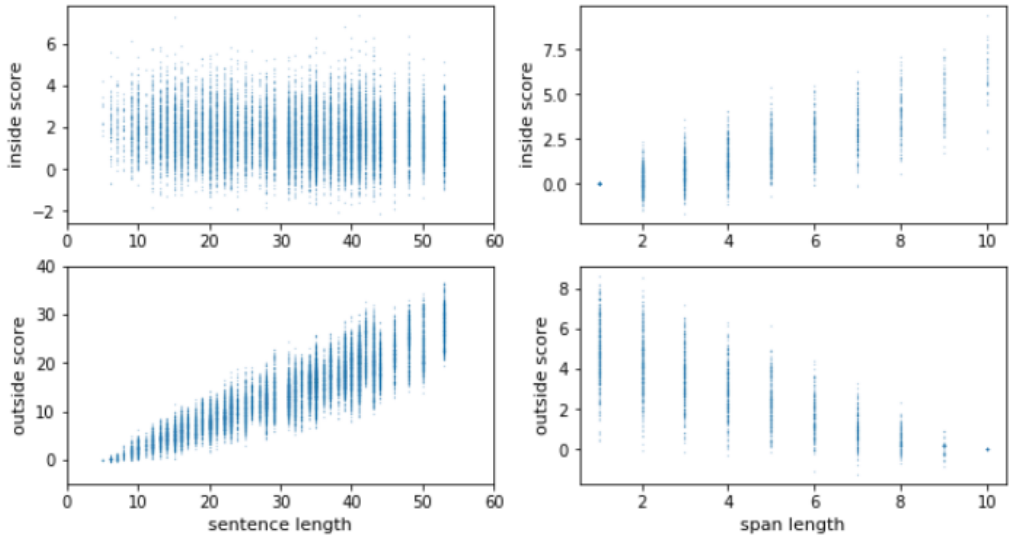

Figure 1: Left: Score distribution of spans of length 5 in sentences of different lengths. Right: Score distribution of spans of different lengths in sentences of length 10.

3 Experiments

3.1 Datasets and Setting

We test our new objective with datasets of two languages: English and Japanese. For English, we use the PTB corpus (Marcus et al., 1993). For Japanese, we use the KTB corpus (Butler et al., 2012). For PTB, we follow the standard split and use section 0-21 for training, 22 for validation, and 23 for testing. For KTB, we shuffle the corpus and take 80% for training, 10% for validation, and 10% for testing.

Following the settings of (Li et al., 2020), we preprocessed the corpora. For punctuation marks, for each language we run two experiments, one with punctuation and one without. For both languages, we use the sentences of length ≤ 40 for training and do not apply any length limit for the test data. Since DIORA needs external word embeddings, we use ELMO (Peters et al., 2018) for English and fasttext (Grave et al., 2018) for Japanese.

3.2 Hyper-parameter

We follow previous work and tune the hyper-parameters such as the learning rate, number of negative samples, hidden size, and the choice of span weight computation on the validation set of each dataset. The hyper-parameters used in our experiments are listed in Table 1.

3.3 Experimental Result

We run each experiment 5 times with different random seeds and report the mean, standard deviation and maximum of the parsing accuracies. The constituency parsing accuracy is evaluated with the unlabelled

	PTB		KTB	
	w. punc	no punc	w. punc	no punc
Hidden Size	200	200	400	400
Learning Rate	2e-3	2e-3	2e-3	2e-3
# Negative Samples	100	100	100	50
Batch Size	64	64	8	8
Span Weight Type	Inside	Inside	Outside	Outside

Table 1: Hyper-parameters for our experiments.

	With Punctuation				No Punctuation			
	F1-10$_\mu$	F1-10$_{\max}$	F1-all$_\mu$	F1-all$_{\max}$	F1-10$_\mu$	F1-10$_{\max}$	F1-all$_\mu$	F1-all$_{\max}$
DIORA	55.15 $\pm$ 0.86	56.61	44.63 $\pm$ 0.46	45.02	60.01 $\pm$ 0.40	60.64	**49.31** $\pm$ 0.45	**49.73**
DIROA-all	**59.18** $\pm$ 4.17	**63.64**	**47.03** $\pm$ 4.20	**50.63**	**61.23** $\pm$ 2.31	**63.33**	47.95 $\pm$ 1.25	49.47
Upper Bound	75.15	-	78.96	-	86.64	-	85.34	-

Table 2: Experimental results on English PTB.

	With Punctuation				No Punctuation			
	F1-10$_\mu$	F1-10$_{\max}$	F1-all$_\mu$	F1-all$_{\max}$	F1-10$_\mu$	F1-10$_{\max}$	F1-all$_\mu$	F1-all$_{\max}$
DIORA	39.33 $\pm$ 2.92	42.83	28.93 $\pm$ 4.29	32.33	44.02 $\pm$ 5.02	**49.18**	35.26 $\pm$ 3.10	38.02
DIROA-all	**43.30** $\pm$ 5.18	**47.73**	**33.00** $\pm$ 3.71	**36.93**	**47.09** $\pm$ 1.79	49.17	**36.37** $\pm$ 3.58	**41.56**
Upper Bound	61.41	-	62.53	-	67.25	-	67.32	-

Table 3: Experimental results on Japanese KTB.

F1 score computed by Evalb[1]. We report F1 scores on test sentences of length ≤ 10 and of all lengths. For the performance of the original DIORA, we rerun the experiments with the hyper-parameters provided by (Li et al., 2020). Since the predicted parse tree is binary, we also provide the upper bound of F1 scores without tree binarization for each dataset.

English

We list the experimental results of English in Table 2. For the setting with punctuation, DIORA with the all-span objective function outperforms the original DIORA on both sentences of length ≤ 10 and all sentences. For the setting without punctuation, our method performs better than the origin DIORA on sentences of length ≤ 10 but worse on all sentences.

Japanese

We list the experimental results of Japanese in Table 3. For settings both with and without punctuation, DIORA with the all-span objective function performs better on all sentences and sentences of length ≤ 10 than the original DIORA.

3.4 Analysis

To validate that the new objective helps DIORA learn better representation with higher-level spans, we evaluate the F1 scores of spans on different length groups(we consider 5 consecutive lengths as a group). We use the models trained in previous experiments of PTB with the best F1-all scores for both DIORA and DIORA-all. The results are shown in Figure 2.

It can be seen that when there is punctuation, DIORA-all consistently outperforms DIORA and its advantage is particularly large on long spans. Without punctuation, the two methods have similar accuracies except for an outlier at length range [31-35], How punctuation influences learning with the original objective and our new objective is an interesting topic for future research.

4 Conclusion

Deep inside-outside recursive autoencoder (DIORA) is a neural-based model for unsupervised constituency parsing. In this paper, we propose a new training objective function for DIORA that considers all-level

[1]https://nlp.cs.nyu.edu/evalb/

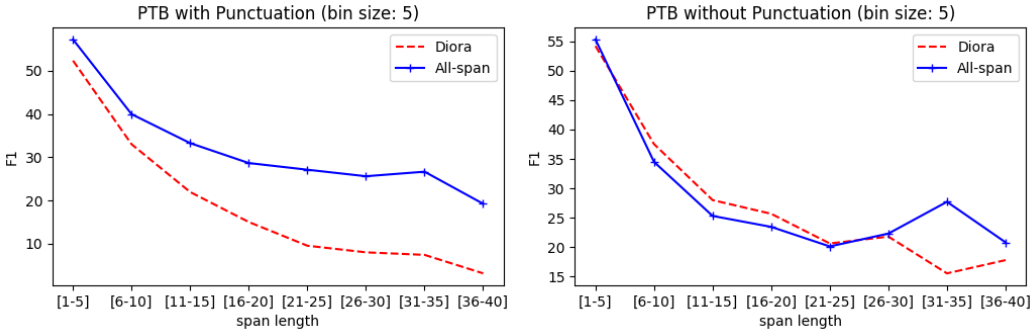

Figure 2: F1 scores on spans of different lengths.

spans instead of leaf-spans only. With the new objective, we show that the model can achieve better overall performance for sentences on English and Japanese.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (61976139).

References

James K Baker. 1979. Trainable grammars for speech recognition. *The Journal of the Acoustical Society of America*, 65(S1):S132–S132.

Alastair Butler, Zhu Hong, Tomoko Hotta, Ruriko Otomo, Kei Yoshimoto, and Zhen Zhou. 2012. Keyaki treebank: phrase structure with functional information for japanese. In *Proceedings of Text Annotation Workshop*.

Andrew Drozdov, Patrick Verga, Mohit Yadav, Mohit Iyyer, and Andrew McCallum. 2019. Unsupervised latent tree induction with deep inside-outside recursive auto-encoders. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1129–1141.

Akiko Eriguchi, Yoshimasa Tsuruoka, and Kyunghyun Cho. 2017. Learning to parse and translate improves neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 72–78, Vancouver, Canada, July. Association for Computational Linguistics.

Daniel Gildea and Martha Palmer. 2002. The necessity of parsing for predicate argument recognition. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 239–246.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 473–483, Vancouver, Canada, July. Association for Computational Linguistics.

Yoon Kim, Chris Dyer, and Alexander Rush. 2019a. Compound probabilistic context-free grammars for grammar induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2369–2385, Florence, Italy, July. Association for Computational Linguistics.

Yoon Kim, Alexander M Rush, Lei Yu, Adhiguna Kuncoro, Chris Dyer, and Gábor Melis. 2019b. Unsupervised recurrent neural network grammars. In *NAACL-HLT (1)*.

Jun Li, Yifan Cao, Jiong Cai, Yong Jiang, and Kewei Tu. 2020. An empirical comparison of unsupervised constituency parsing methods. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3278–3283, Online, July. Association for Computational Linguistics.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.

Yikang Shen, Zhouhan Lin, Chin wei Huang, and Aaron Courville. 2018. Neural language modeling by jointly learning syntax and lexicon. In *International Conference on Learning Representations*.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. 2019. Ordered neurons: Integrating tree structures into recurrent neural networks. In *International Conference on Learning Representations*.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. 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 1556–1566, Beijing, China, July. Association for Computational Linguistics.

Xinyi Wang, Hieu Pham, Pengcheng Yin, and Graham Neubig. 2018. A tree-based decoder for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4772–4777, Brussels, Belgium, October-November. Association for Computational Linguistics.