Efficient Beam Tree Recursion (Appendix)

Anonymous Author(s)
Affiliation
Address
email

1 Organization

In Section 2, we describe the settings of all the tasks and datasets that we have tested our models on. In Section 3, we provide additional results on logical inference and sentiment classification. Then, in Section 4, we present an extended survey of related works. In Section 5, we detail our architecture setup including the sequence-interaction models. In Section 6, we provide our hyperparameters. Last, further details on BT-RvNN (if necessary) can be found in the document titled "Beam Tree Recursive Cells" provided in the supplementary (however, we described the salient aspects of the model in the main paper).

2 Task Details

ListOps: ListOps was introduced by Nangia and Bowman [36] and is a task for solving nested lists of mathematical operations. It is a 10-way classification task. Similar to Chowdhury and Caragea [6], we train our models on the original training set with all samples ≥ 100 sequence lengths filtered out. We use the original development set for validation. We test on the following sets: the original test set (near-IID split); the length generalization splits from Havrylov et al. [20] that include samples of much higher lengths; the argument generalization splits from Anonymous [1] that involve an unseen number of maximum arguments for each operator; and the LRA split (which has both higher sequence length and higher argument number) from Tay et al. [52].

Logical Inference: Logical Inference was introduced by Bowman et al. [3] and is a task that involves classifying fine-grained inferential relations between two given sequences in a form similar to that of formal sentences of propositional logic. Similar to Tran et al. [54], our models were trained on splits with logical connectives ≤ 6. We show the results in OOD test sets with logical connections 10-12. We use the same splits as Shen et al. [44], Tran et al. [54], Chowdhury and Caragea [6].

SST5: SST5 is a fine-grained 5-way sentiment classification task introduced by Socher et al. [50]. We use the original splits.

IMDB: IMDB is a binary sentiment classification task from Maas et al. [31]. We use the same train, validation, and IID test sets as created in Anonymous [1]. We also use the contrast set Gardner et al. [15] and counterfactual set Kaushik et al. [26] as additional test splits.

QQP: QQP[1] [25] is a task of classifying whether two given sequences in a pair are paraphrases of each other or not. As standard Wang et al. [56], we randomly sample $10,000$ samples for validation and IID test set such that for each split $5,000$ samples are maintained to be paraphrases and the other $5,000$ are maintained to be not paraphrases. We also use the adversarial test sets PAWS$_{QQP}$ and PAWS$_{WIKI}$ form Zhang et al. [62].

SNLI: SNLI [2] is a natural language inference (NLI) task. It is a 3-way classification task to classify the inferential relation between two given sequences. We use the same train, development, and IID test set splits as in Chowdhury and Caragea [6]. Any data with a sequence of length ≥ 150 is filtered

[1] https://data.quora.com/First-Quora-Dataset-Release-QuestionPairs

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

Table 1: Mean accuracy and standard deviation on the Logical Inference [3] for ≥ 10 number of operations after training on samples with ≤ 6 operations, and on SST5 [50] and IMDB [31]. **Count.** represents counterfactual test split from Kaushik et al. [26] and **Cont.** represents contrast test split from Gardner et al. [15] The best results are shown in bold. Our models were run 3 times on different seeds. Subscript represents standard deviation. As an example, $90_1 = 90 \pm 0.1$

Model	**Logical Inference Number of Operations**			**SST5**	**IMDB**		
	10	11	12	**IID**	**IID**	**Cont.**	**Count.**
GT-GRC	90.33_{22}	88.43_{18}	85.70_{24}	$51.67_{8.8}$	$85.11_{1}0$	70.63_{21}	81.97_{5}
EGT-GRC	75.79_{61}	73.38_{68}	$69.68_{7.8}$	51.63_{14}	$86.58_{2.7}$	$72_{9.2}$	81.76_{14}
CRvNN	$94.51_{2.9}$	$\mathbf{94.48_{5.6}}$	92.73_{15}	51.75_{11}	$91.47_{1.2}$	$76.98_{5.8}$	$83.68_{7.8}$
OM	94.95_{2}	$93.9_{2.2}$	$93.36_{6.2}$	$52.30_{2.7}$	$\mathbf{91.69_{0.5}}$	$\mathbf{77.80_{15}}$	$\mathbf{85.38_{3.5}}$
BT-GRC	$95.04_{2.3}$	$94.29_{3.8}$	$93.36_{2.4}$	$\mathbf{52.32_{4.7}}$	$91.29_{1.2}$	75.07_{29}	82.86_{23}
BT-GRC OS	$\mathbf{95.43_{4.5}}$	$94.21_{6.6}$	$\mathbf{93.39_{1.5}}$	$51.92_{7.2}$	$90.86_{9.3}$	75.68_{21}	84.77_{11}
EBT-GRC	$94.95_{1.5}$	$93.87_{7.4}$	$93.04_{6.7}$	52.22_{1}	$91.47_{1.2}$	76.16_{17}	84.29_{12}

out from the training set for efficiency. We use also additional test set splits for stress tests. We use the hard test set split from Gururangan et al. [19], the break test set from Glockner et al. [16], and the counterfactual test set from Kaushik et al. [26].

MNLI: MNLI [57] is another NLI dataset, which is similar to SNLI in format. We use the original development sets (match and mismatch) as test sets. We filter out all data with any sequence length ≥ 150 from the training set. Our actual development set is a random sample of $10,000$ data-points from the filtered training set. As additional testing sets, we use the development set of Conjunctive NLI (ConjNLI) [41] and a few of the stress sets from Naik et al. [35]. These stress test sets include - Negation Match (NegM), Negation Mismatch (NegMM), Length Match (LenM), and Length Mismatch (LenMM). NegM and NegMM add tautologies containing "not" terms - this can bias the models to classify contradiction as the inferential relation because the training set contains spurious correlations between existence of "not" related terms and the class of contradiction. LenM and LenMM add tautologies to artificially increase the lengths of the samples without changing the inferential relation class.

3 Additional Results

In Table 1, we show that our EBT-GRC model can keep up fairly well with BT-GRC and BT-GRC OS on logical inference [3] and sentiment classification tasks like SST5 [50], and IMDB [15] while being much more computationally efficient as demonstrated in the main paper.

4 Extended Related Works

RvNN History: Recursive Neural Networks (RvNNs) in the more specified sense of building representations through trees and directed acyclic graphs were proposed in [40, 18]. Socher et al. [48, 49, 50] extended the use of RvNNs in Natural Language Processing (NLP) by considering constituency trees and dependency trees. A few works [63, 51, 28, 64] started adapting Long Shot-term Memory Networks [21] as a cell function for recursive processing. Le and Zuidema [29], Maillard et al. [33] proposed a chart-based method for simulating bottom-up Recursive Neural Networks through dynamic programming. Shi et al. [47], Munkhdalai and Yu [34] explored heuristics-based tree-structured RvNNs.

RvNNs can also be simulated by stack-augmented recurrent neural networks (RNNs) to an extent (similar to how pushdown automata can model context-free grammar [42, 27]). There are multiple works on stack-augmented RNNs [4, 60, 32]. Ordered Memory [44] is one of the more modern such examples. More recently, DuSell and Chiang [12, 13] explored non-deterministic stack augmented RNNs and [9] explored other expressive models. Wu [58] presented a survey of latent structure models.

Choi et al. [5] proposed a greedy search strategy based on easy-first algorithm [17, 30] for auto-parsing structures for recursion utilizing STE gumbel softmax for gradient signals. Peng et al. [38] extended the framework with SPIGOT and Havrylov et al. [20] extended it with reinforcement learning (RL). Anonymous [1] extended it with beam search and soft top-k. Chowdhury and Caragea [6], Zhang et al. [61] introduced different forms of soft-recursion.

Top-down Signal: Similar to us, Teng and Zhang [53] explored bidirectional signal propagation (bottom-up and top-down). However, they sent top-down signal in a sequential manner which can be expensive - either it can get slow without parallelization or memory-wise expensive with parallelization of contextualization of nodes in the same height. Our approach in EBT-GAU also has some kinship with BP-Transformer [59]. BP-Transformer allows message passing between a fixed subset of parent nodes and terminal nodes created using a heuristics-based balanced binary tree. Chart-based models can also create sequence contextualized representations [10, 11] but they can be quite expensive by default [1] needing their own separate techniques [22, 23].

Transformers + RvNNs: There have been several approaches to incorporating RvNN-like inductive biases to Transformers. For instance, Universal Transformer [8] introduced weight-sharing and dynamic halt to Transformers. Csordás et al. [7] extended on universal transformer with geometric attention for locality bias and gating. Shen et al. [46] built on weight-shared transformers with high layer depth and group self-attention. Wang et al. [55], Nguyen et al. [37], Shen et al. [45] added hierarchical structural biases to self-attention. Fei et al. [14] biased pre-trained Transformers to have constituent information in intermediate representations. Hu et al. [22] used Transformer as binary recursive cells in chart-based encoders.

5 Architecture details

5.1 Sentence Encoder Models

For the sentence encoder models the architectural framework we use is the same siamese dual-encoder setup as Anonymous [1].

5.2 Sentence Interaction Models

GAU-Block: Our specific implementation of a GAU-block [24] is detailed below. Our GAU-Block can be defined as GAUBlock(x, p, G). The function arguments are of the following forms: $x \in I\!R^{n\times d}, p \in I\!R^{l\times d}$ and $G \in \{0,1\}^{n\times l}$. x accepts the main sequence of vectors that is to serve as attention queries; p accepts either the sequence of intermediate node representations created from our RvNN (for parent attention) or it accepts the same input as x (for usual cases); p serves as keys and values for attention; G accepts either the adjacency matrix in case of parent attention (where $G_{ij} = 1$ iff p_j is a parent of x_i else $G_{ij} = 0$), otherwise, it accepts just the usual attention mask; either way, G serves as an attention mask.

$$x' = LN(xW_{init} + b_{init}); \quad p' = LN(pW_{init} + b_{init}) \tag{1}$$

$$u = \text{SiLU}(x'W_u + b_u); \quad v = \text{SiLU}(p'W_v + b_v) \tag{2}$$

$$q = z_q \odot \text{SiLU}(x'W_z + b_z) + zb_q; \quad k = z_k \odot \text{SiLU}(p'W_z + b_z) + zb_k \tag{3}$$

$$A = \text{Softmax}(\frac{qk^T + pos}{\sqrt{2d}}, mask = G) \tag{4}$$

$$v' = Av \tag{5}$$

$$o = (u \odot v')W_o + b_o \tag{6}$$

$$g = \text{Sigmoid}([o; x]W_{gate} + b_{gate}) \tag{7}$$

$$out = g \odot o + (1 - g) \odot x \tag{8}$$

Here, $W_{init} \in I\!R^{d\times d}; W_z \in I\!R^{d\times d_h}, W_u, W_v \in I\!R^{d\times 2d}, b_{init}, b_z, b_o \in I\!R^{d}; z_q, zb_q, z_k, zb_k \in I\!R^{d_h}; b_u, b_v \in I\!R^{2d}, W_o, W_{gate} \in I\!R^{2d\times d}$. [;] represents concatenation.

LN is layer normalization. pos is calculated using the technique of Shaw et al. [43] using relative tree height distance for parent attention, or relative positional distance for usual cases.

GAU Sequence Interaction Setup: Let GAUStack represent some arbitrary number of compositions of GAUBlocks (multilayered GAU block). GAUStack has the same function arguments as GAUBlock. Given two sequences (x_1, x_2) and their corresponding attention masks (M_1, M_2) as inputs where $x_1 \in I\!R^{n_1 \times d}, x_2 \in I\!R^{n_2 \times d}, M_1 \in \{0,1\}^{n_1 \times n_1}, M_1 \in \{0,1\}^{n_2 \times n_2}$, the GAU setup can be expressed as:

$$inp = [CLS + seg_1; x_1 + seg_1; SEP; CLS + seg_2, x_2 + seg_2] \tag{9}$$

$$r = \text{GAUStack}(x = inp, p = inp, G = [M_1; M_2]) \tag{10}$$

$$\alpha = \text{Softmax}(\text{GELU}(rW_1 + b_1)W_2 + b_2) \tag{11}$$

$$cls' = \sum_i \alpha_i r \tag{12}$$

$$logits = \text{GELU}(cls' W_1^{logits} + b_1^{logits})W_2^{logits} + b_2^{logits} \tag{13}$$

Here, $CLS, SEP, seg_1, seg_2 \in I\!R^{1 \times d}$ are randomly initialized trainable vectors; seg_1, seg_2 are segment embeddings. $W_1 \in I\!R^{d \times d}, W_2 \in I\!R^{d \times 1}$; $b_1, b_2, b_1^{logits} \in I\!R^d; b_2^{logits} \in I\!R^c; W_1^{logits} \in I\!R^{d \times d}$, $W_2^{logits} \in I\!R^{d \times c}$. c is the number of classes for the task.

EGT-GAU Sequence Interaction Setup: EGT-GAU starts from the same input as above. Let us also assume we have the EGT-GRC(x) module which takes a sequence of vectors $x \in I\!R^{n \times d}$ as the input to recursively process and outputs (cls, p, G) where $cls \in I\!R^{1 \times d}$ is the root representation, $p \in I\!R^{l \times d}$ is the sequence of non-terminal representations from the tree, and $G \in \{0,1\}^{n \times l}$ is the adjacency matrix for parent attention (i.e., $G_{ij} = 1$ iff p_j is a parent of x_i, else $G_{ij} = 0$). Technically, tree height information is also extracted for relative position but we do not express that explicitly for the sake of brevity. With these elements, EGT-GAU can be expressed as below:

$$cls_1, p_1, G_1 = \text{EGT-GRC}(x = x_1); \quad cls_2, p_2, G_2 = \text{EGT-GRC}(x = x_2) \tag{14}$$

$$x_1' = \text{GAUStack}_1(x = x_1, p = p_1, G = G_1); \quad x_2' = \text{GAUStack}_1(x = x_2, p = p_2, G = G_2) \tag{15}$$

$$cls_1' = \text{GELU}(cls_1 W_1^{cls} + b_1^{cls})W_2^{cls} + b_2^{cls}; \quad cls_2' = \text{GELU}(cls_2 W_1^{cls} + b_1^{cls})W_2^{cls} + b_2^{cls} \tag{16}$$

$$inp = [cls_1' + seg_1; x_1' + seg_1; SEP; cls_2' + seg_2, x_2' + seg_2] \tag{17}$$

$$r = \text{GAUStack}_2(x = inp, p = inp, G = [M_1; M_2]) \tag{18}$$

Everything else after eqn. 18 is the same as eqn. 11 to 13. $SEP, seg_1, seg_2 \in I\!R^{1 \times d}$; seg_1, seg_2 are segment embeddings as before. $W_1^{cls}, W_2^{cls} \in I\!R^{d \times d}; b_1^{cls}, b_2^{cls} \in I\!R^d$.

EBT-GAU Sequence Interaction Setup: This setup is similar to that of EGT-GAU but with a few changes. EBT-GAU uses EBT-GRC as a module instead of EGT-GRC. EBT-GAU returns outputs of the form (cls, bp, bG, s) where $cls \in I\!R^{1 \times d}$ is the beam-score-weighted-averaged root representation, $bp \in I\!R^{b \times l \times d}$ are the beams (beam size b) of sequences of non-terminal representations from the tree, $bG \in \{0,1\}^{b \times n \times l}$ are the beams of adjacency matrices for parent attention, and $s \in I\!R^b$ are the softmax-normalized beam scores. Let NGAUStack represent the same function as GAUStack but formalized for batched processing of multiple beams of sequences. With these elements, EBT-GAU can be expressed as:

$$cls_1, bp_1, bG_1, s_1 = \text{EBT-GRC}(x = x_1); \quad cls_2, bp_2, bG_2, s_2 = \text{EBT-GRC}(x = x_2) \tag{19}$$

$$bx_1 = \text{repeat}(x_1, b); \quad bx_2 = \text{repeat}(x_2, b) \tag{20}$$

$$bx_1' = \text{NGAUStack}_1(bx_1, bp_1, bG_1); \quad bx_2' = \text{NGAUStack}_1(bx_2, bp_2, bG_2) \tag{21}$$

$$x_1' = \sum_i s[i] \cdot bx_1'[i]; \quad x_2' = \sum_i s[i] \cdot bx_2'[i] \tag{22}$$

Everything else after eqn. 22 is the same as the equations 16-18 followed by the equations 11 to 13. repeat(x, b) changes $x \in I\!R^{n \times d}$ to $bx \in I\!R^{b \times n \times d}$ by batching the same x for b times.

6 Hyperparameter details

For sentence encoder models, we use the same hyperparameters as [1] (the preprint of the paper is available in the supplementary in anonymized form) for all the datasets. The only new hyperparameter for EBT-GRC is d_s which we set as 64; otherwise the hyperparameters are the same as that of BT-GRC or BT-GRC OS. We discuss the hyperparameters of the sequence interaction models next. For EBT-GAU/EGT-GAU, we used a two-layered weight-shared GAU-Blocks for NGAUStack_1/GAUStack_1 and a three-layered weight-shared GAU-Blocks for GAUStack_2 (for parameter efficiency and regularization). GAU uses a five-layered GAU-Blocks (weights unshared) for GAUStack so that the parameters are similar to that of EBT-GAU or EGT-GAU. We use a dropout of 0.1 after the multiplation with W_o in each GAUBlock layer and a head size d_h of 128 (similar to Hua et al. [24]). For relative position, we set $k = 5$ (k here corresponds the receptive field for relative attention in Shaw et al. [43]) for normal GAUBlocks and $k = 10$ for parent attention (since parent attention is only applied to higher heights, we do not need to initialize weights for negative relative distances). Other hyperparameters are kept same as the sentence encoder models. The hyperparameters of MNLI, SNLI, and QQP are shared. Note that all the natural language tasks are trained with fixed 840B Glove Embeddings [39] as in Anonymous [1]. All models were trained in a single Nvidia RTX A6000. The code is available in the supplementary.

References

[1] Anonymous. Beam tree recursive cells. In *Proceedings of the International Conference on Machine Learning*, 2023.

[2] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL `https://aclanthology.org/D15-1075`.

[3] Samuel R. Bowman, Christopher D. Manning, and Christopher Potts. Tree-structured composition in neural networks without tree-structured architectures. In *Proceedings of the 2015th International Conference on Cognitive Computation: Integrating Neural and Symbolic Approaches - Volume 1583*, COCO'15, page 37–42, Aachen, DEU, 2015. CEUR-WS.org.

[4] Samuel R. Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D. Manning, and Christopher Potts. A fast unified model for parsing and sentence understanding. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1466–1477, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1139. URL `https://aclanthology.org/P16-1139`.

[5] Jihun Choi, Kang Min Yoo, and Sang-goo Lee. Learning to compose task-specific tree structures. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5094–5101. AAAI Press, 2018. URL `https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16682`.

[6] Jishnu Ray Chowdhury and Cornelia Caragea. Modeling hierarchical structures with continuous recursive neural networks. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 1975–1988. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/chowdhury21a.html`.

[7] Róbert Csordás, Kazuki Irie, and Jürgen Schmidhuber. The neural data router: Adaptive control flow in transformers improves systematic generalization. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=KBQP4A_J1K`.

[8] Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal transformers. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=HyzdRiR9Y7`.

[9] Gr'egoire Del'etang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Marcus Hutter, Shane Legg, and Pedro A. Ortega. Neural networks and the chomsky hierarchy. *ArXiv*, abs/2207.02098, 2022.

[10] Andrew Drozdov, Patrick Verga, Mohit Yadav, Mohit Iyyer, and Andrew McCallum. 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, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1116. URL `https://aclanthology.org/N19-1116`.

[11] Andrew Drozdov, Subendhu Rongali, Yi-Pei Chen, Tim O'Gorman, Mohit Iyyer, and Andrew McCallum. Unsupervised parsing with S-DIORA: Single tree encoding for deep inside-outside recursive autoencoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4832–4845, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.392. URL `https://aclanthology.org/2020.emnlp-main.392`.

[12] Brian DuSell and David Chiang. Learning context-free languages with nondeterministic stack RNNs. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 507–519, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.conll-1.41. URL `https://aclanthology.org/2020.conll-1.41`.

[13] Brian DuSell and David Chiang. Learning hierarchical structures with differentiable nondeterministic stacks. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=5LXw_QplBiF`.

[14] Hao Fei, Yafeng Ren, and Donghong Ji. Retrofitting structure-aware transformer language model for end tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2151–2161, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.168. URL `https://aclanthology.org/2020.emnlp-main.168`.

[15] Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.117. URL `https://aclanthology.org/2020.findings-emnlp.117`.

[16] Max Glockner, Vered Shwartz, and Yoav Goldberg. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2103. URL `https://aclanthology.org/P18-2103`.

[17] Yoav Goldberg and Michael Elhadad. An efficient algorithm for easy-first non-directional dependency parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 742–750, Los Angeles, California, June 2010. Association for Computational Linguistics. URL `https://aclanthology.org/N10-1115`.

[18] C. Goller and A. Kuchler. Learning task-dependent distributed representations by backpropagation through structure. In *Proceedings of International Conference on Neural Networks (ICNN'96)*, volume 1, pages 347–352 vol.1, 1996. doi: 10.1109/ICNN.1996.548916.

[19] Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2017. URL https://aclanthology.org/N18-2017.

[20] Serhii Havrylov, Germán Kruszewski, and Armand Joulin. Cooperative learning of disjoint syntax and semantics. 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 1118–1128, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1115. URL https://aclanthology.org/N19-1115.

[21] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9 (8):1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

[22] Xiang Hu, Haitao Mi, Zujie Wen, Yafang Wang, Yi Su, Jing Zheng, and Gerard de Melo. R2D2: Recursive transformer based on differentiable tree for interpretable hierarchical language modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4897–4908, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.379. URL https://aclanthology.org/2021.acl-long.379.

[23] Xiang Hu, Haitao Mi, Liang Li, and Gerard de Melo. Fast-R2D2: A pretrained recursive neural network based on pruned CKY for grammar induction and text representation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2809–2821, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.emnlp-main.181.

[24] Weizhe Hua, Zihang Dai, Hanxiao Liu, and Quoc Le. Transformer quality in linear time. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 9099–9117. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/hua22a.html.

[25] Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. First quora dataset release: Question pairs. In *Quora*, 2017.

[26] Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Sklgs0NFvr.

[27] Donald E. Knuth. On the translation of languages from left to right. *Information and Control*, 8 (6):607 – 639, 1965. ISSN 0019-9958.

[28] Phong Le and Willem Zuidema. Compositional distributional semantics with long short term memory. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 10–19, Denver, Colorado, June 2015. Association for Computational Linguistics. doi: 10.18653/v1/S15-1002. URL https://aclanthology.org/S15-1002.

[29] Phong Le and Willem Zuidema. The forest convolutional network: Compositional distributional semantics with a neural chart and without binarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1155–1164, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1137. URL https://aclanthology.org/D15-1137.

[30] Ji Ma, Jingbo Zhu, Tong Xiao, and Nan Yang. Easy-first POS tagging and dependency parsing with beam search. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 110–114, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL https://aclanthology.org/P13-2020.

[31] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL `https://aclanthology.org/P11-1015`.

[32] Jean Maillard and Stephen Clark. Latent tree learning with differentiable parsers: Shift-reduce parsing and chart parsing. In *Proceedings of the Workshop on the Relevance of Linguistic Structure in Neural Architectures for NLP*, pages 13–18, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-2903. URL `https://aclanthology.org/W18-2903`.

[33] Jean Maillard, Stephen Clark, and Dani Yogatama. Jointly learning sentence embeddings and syntax with unsupervised tree-lstms. *Natural Language Engineering*, 25(4):433–449, 2019. doi: 10.1017/S1351324919000184.

[34] Tsendsuren Munkhdalai and Hong Yu. Neural tree indexers for text understanding. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 11–21, Valencia, Spain, April 2017. Association for Computational Linguistics. URL `https://aclanthology.org/E17-1002`.

[35] Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/C18-1198`.

[36] Nikita Nangia and Samuel Bowman. ListOps: A diagnostic dataset for latent tree learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 92–99, New Orleans, Louisiana, USA, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-4013. URL `https://aclanthology.org/N18-4013`.

[37] Xuan-Phi Nguyen, Shafiq Joty, Steven Hoi, and Richard Socher. Tree-structured attention with hierarchical accumulation. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=HJxK5pEYvr`.

[38] Hao Peng, Sam Thomson, and Noah A. Smith. Backpropagating through structured argmax using a SPIGOT. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1863–1873, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1173. URL `https://aclanthology.org/P18-1173`.

[39] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL `https://aclanthology.org/D14-1162`.

[40] Jordan B. Pollack. Recursive distributed representations. *Artificial Intelligence*, 46(1):77 – 105, 1990. ISSN 0004-3702. doi: https://doi.org/10.1016/0004-3702(90)90005-K. URL `http://www.sciencedirect.com/science/article/pii/000437029090005K`.

[41] Swarnadeep Saha, Yixin Nie, and Mohit Bansal. ConjNLI: Natural language inference over conjunctive sentences. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8240–8252, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.661. URL `https://aclanthology.org/2020.emnlp-main.661`.

[42] M.P. Schützenberger. On context-free languages and push-down automata. *Information and Control*, 6(3):246 – 264, 1963. ISSN 0019-9958.

[43] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2074. URL `https://aclanthology.org/N18-2074`.

[44] Yikang Shen, Shawn Tan, Arian Hosseini, Zhouhan Lin, Alessandro Sordoni, and Aaron C Courville. Ordered memory. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 5037–5048. Curran Associates, Inc., 2019. URL `http://papers.nips.cc/paper/8748-ordered-memory.pdf`.

[45] Yikang Shen, Yi Tay, Che Zheng, Dara Bahri, Donald Metzler, and Aaron Courville. StructFormer: Joint unsupervised induction of dependency and constituency structure from masked language modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7196–7209, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.559. URL `https://aclanthology.org/2021.acl-long.559`.

[46] Zhiqiang Shen, Zechun Liu, and Eric Xing. Sliced recursive transformer. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV*, pages 727–744. Springer, 2022.

[47] Haoyue Shi, Hao Zhou, Jiaze Chen, and Lei Li. On tree-based neural sentence modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4631–4641, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1492. URL `https://aclanthology.org/D18-1492`.

[48] Richard Socher, Christopher D. Manning, and Andrew Y. Ng. Learning continuous phrase representations and syntactic parsing with recursive neural networks. In *In Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop*, 2010.

[49] Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 151–161, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. URL `https://aclanthology.org/D11-1014`.

[50] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL `https://aclanthology.org/D13-1170`.

[51] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 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 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1150. URL `https://aclanthology.org/P15-1150`.

[52] Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena : A benchmark for efficient transformers. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=qVyeW-grC2k`.

[53] Zhiyang Teng and Yue Zhang. Head-lexicalized bidirectional tree LSTMs. *Transactions of the Association for Computational Linguistics*, 5:163–177, 2017. doi: 10.1162/tacl_a_00053. URL `https://aclanthology.org/Q17-1012`.

[54] Ke Tran, Arianna Bisazza, and Christof Monz. The importance of being recurrent for modeling hierarchical structure. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4731–4736, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1503. URL `https://aclanthology.org/D18-1503`.

[55] Yaushian Wang, Hung-Yi Lee, and Yun-Nung Chen. Tree transformer: Integrating tree structures into self-attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1061–1070, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1098. URL `https://aclanthology.org/D19-1098`.

[56] Zhiguo Wang, Wael Hamza, and Radu Florian. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, page 4144–4150. AAAI Press, 2017. ISBN 9780999241103.

[57] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL `https://aclanthology.org/N18-1101`.

[58] Zhaofeng Wu. Learning with latent structures in natural language processing: A survey. *ArXiv*, abs/2201.00490, 2022.

[59] Zihao Ye, Qipeng Guo, Quan Gan, Xipeng Qiu, and Zheng Zhang. Bp-transformer: Modelling long-range context via binary partitioning. *arXiv preprint arXiv:1911.04070*, 2019.

[60] Dani Yogatama, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Wang Ling. Learning to compose words into sentences with reinforcement learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[61] Aston Zhang, Yi Tay, Yikang Shen, Alvin Chan, and SHUAI ZHANG. Self-instantiated recurrent units with dynamic soft recursion. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 6503–6514. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper/2021/file/3341f6f048384ec73a7ba2e77d2db48b-Paper.pdf`.

[62] Yuan Zhang, Jason Baldridge, and Luheng He. PAWS: Paraphrase adversaries from word scrambling. 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 1298–1308, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1131. URL `https://aclanthology.org/N19-1131`.

[63] Xiaodan Zhu, Parinaz Sobihani, and Hongyu Guo. Long short-term memory over recursive structures. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1604–1612, Lille, France, 07–09 Jul 2015. PMLR. URL `https://proceedings.mlr.press/v37/zhub15.html`.

[64] Xiaodan Zhu, Parinaz Sobhani, and Hongyu Guo. DAG-structured long short-term memory for semantic compositionality. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 917–926, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1106. URL `https://aclanthology.org/N16-1106`.