

# Visually Grounded Compound PCFGs

Yanpeng Zhao<sup>ε</sup>

<sup>ε</sup>ILCC, University of Edinburgh  
yanp.zhao@ed.ac.uk

Ivan Titov<sup>εα</sup>

<sup>ε</sup>ILLC, University of Amsterdam  
ititov@inf.ed.ac.uk

## Abstract

Exploiting visual groundings for language understanding has recently been drawing much attention. In this work, we study visually grounded grammar induction and learn a constituency parser from both unlabeled text and its visual groundings. Existing work on this task (Shi et al., 2019) optimizes a parser via REINFORCE and derives the learning signal only from the alignment of images and sentences. While their model is relatively accurate overall, its error distribution is very uneven, with low performance on certain constituents types (e.g., 26.2% recall on verb phrases, VPs) and high on others (e.g., 79.6% recall on noun phrases, NPs). This is not surprising as the learning signal is likely insufficient for deriving all aspects of phrase-structure syntax and gradient estimates are noisy. We show that using an extension of probabilistic context-free grammar model we can do fully-differentiable end-to-end visually grounded learning. Additionally, this enables us to complement the image-text alignment loss with a language modeling objective. On the MSCOCO test captions, our model establishes a new state of the art, outperforming its non-grounded version and, thus, confirming the effectiveness of visual groundings in constituency grammar induction. It also substantially outperforms the previous grounded model, with largest improvements on more ‘abstract’ categories (e.g., +55.1% recall on VPs).<sup>1</sup>

## 1 Introduction

Grammar induction is a task of finding latent hierarchical structure of language. As a fundamental problem in computational linguistics, it has been extensively studied for decades (Lari and Young, 1990; Carroll and Charniak, 1992; Clark, 2001;

Klein and Manning, 2002). Recently, deep learning models have been shown very effective across NLP tasks and have also been applied to grammar induction, greatly advancing the area (Shen et al., 2018, 2019; Kim et al., 2019a,b; Jin et al., 2019). These neural grammar-induction approaches have been generally limited to relying on text, without considering learning signals from other modalities.

In contrast, the crucial aspect of natural language learning is that it is grounded in perceptual experiences (Barsalou, 1999; Fincher-Kiefer, 2001; Bisk et al., 2020). We thus anticipate improved language understanding by leveraging grounded learning. Promising results from grounded learning have been emerging in areas such as representation learning (Bruni et al., 2014; Kiela et al., 2018; Bordes et al., 2019). Typically, they use visual images as perceptual groundings of language and aim at improving *continuous vector representations* of language (e.g., word or sentence embeddings). In this work, we consider a more challenging problem: can visual groundings help us induce *syntactic structure*? We refer to this problem as *visually grounded grammar induction*.

Shi et al. (2019) propose a visually grounded neural syntax learner (VG-NSL) to tackle the task. Specifically, they learn a parser from aligned image-sentence pairs (e.g., image-caption data), where each sentence describes visual content of the corresponding image. The parser is optimized via REINFORCE, where the reward is computed by scoring the alignment of images and constituents. While straightforward, matching-based rewards can, as we will discuss further in the paper, make the parser focus only on more local and short constituents (e.g., 79.6% recall on NPs) and to perform poorly on longer ones (e.g., 26.2% recall on VPs) (Shi et al., 2019). While for the former it outperforms the text-only grammar induction methods, for the latter it substantially underachieves. This may not

<sup>1</sup>Our code is available at <https://git.io/JU0JJ>.

be surprising, as it is not guaranteed that every constituent of a sentence has its visual representation in the aligned image; the reward signals can be noisy and insufficient to capture all aspects of phrase-structure syntax. Consequently, Shi et al. (2019) have to rely on language-specific inductive bias to obtain more informative reward signals. Another issue with VG-NSL is that the parser does not admit tractable estimation of the partition function and the posterior probabilities for constituent boundaries needed to compute the expected reward in closed form. Instead, VG-NSL relies on Monte Carlo policy gradients, potentially suffering from high variance.

To alleviate the first issue, we propose to complement the image-text alignment-based loss with a loss defined on unlabeled text (i.e., its log-likelihood). As re-confirmed with neural models in Shen et al. (2019) and Kim et al. (2019a), text itself can drive induction of rich syntactic knowledge, so additionally optimizing the parser on raw text can be beneficial and complementary to visual grounded learning. To resolve the second issue, we resort to an extension of probabilistic context-free grammar (PCFG) parsing model, compound PCFG (Kim et al., 2019a). It admits tractable estimation of the posteriors, needed in the alignment loss, with dynamical programming and leads to a fully-differentiable end-to-end visually grounded learning. More importantly, the PCFG parser lets us complement the alignment loss with a language modeling objective.

Our key contributions can be summarized as follows: (1) we propose a fully-differentiable end-to-end visually grounded learning framework for grammar induction; (2) we additionally optimize a language modeling objective to complement visually grounded learning; (3) we conduct experiments on MSCOCO (Lin et al., 2014) and observe that our model has a higher recall than VG-NSL for five out of six most frequent constituent labels. For example, it surpasses VG-NSL by 55.1% recall on VPs and by 48.7% recall on prepositional phrases (PPs). Comparing to a model trained purely via visually grounded learning, extending the loss with a language modeling objective improves the overall F1 from 50.5% to 59.4%.

## 2 Background and Motivation

Our model relies on compound PCFGs (Kim et al., 2019a) and generalizes the visually grounded gram-

mar learning framework of Shi et al. (2019). We will describe the relevant aspects of both frameworks in Sections 2.1-2.2, and then discuss their limitations (Section 2.3).

### 2.1 Compound PCFGs

Compound PCFGs extend context-free grammars (CFGs) and, to establish notation, we start by briefly introducing them. A CFG is defined as a 5-tuple  $\mathcal{G} = (S, \mathcal{N}, \mathcal{P}, \Sigma, \mathcal{R})$  where  $S$  is the start symbol,  $\mathcal{N}$  is a finite set of nonterminals,  $\mathcal{P}$  is a finite set of preterminals,  $\Sigma$  is a finite set of terminals,<sup>2</sup> and  $\mathcal{R}$  is a set of production rules in the Chomsky normal form:

$$\begin{aligned} S &\rightarrow A, & A &\in \mathcal{N}, \\ A &\rightarrow BC, & A &\in \mathcal{N}, B, C \in \mathcal{N} \cup \mathcal{P}, \\ T &\rightarrow w, & T &\in \mathcal{P}, w \in \Sigma. \end{aligned}$$

PCFGs extend CPGs by associating each production rule  $r \in \mathcal{R}$  with a non-negative scalar  $\pi_r$  such that  $\sum_{r:A \rightarrow \gamma} \pi_r = 1$ , i.e., the probabilities of production rules with the same left-hand-side nonterminal sum to 1. The strong context-free assumption hinders PCFGs and prevent them from being effective in the grammar induction context. Compound PCFGs (C-PCFGs) mitigate this issue by assuming that rule probabilities follow a compound probability distribution (Robbins, 1951):

$$\pi_r = g_r(\mathbf{z}; \theta), \quad \mathbf{z} \sim p(\mathbf{z}),$$

where  $p(\mathbf{z})$  is a prior distribution of the latent  $\mathbf{z}$ , and  $g_r(\cdot; \theta)$  is parameterized by  $\theta$  and yields a rule probability  $\pi_r$ . Depending on the rule type,  $g_r(\cdot; \theta)$  takes one of these forms:

$$\begin{aligned} \pi_{S \rightarrow A} &= \frac{\exp(\mathbf{u}_A^T f_s([\mathbf{w}_S; \mathbf{z}]))}{\sum_{A' \in \mathcal{N}} \exp(\mathbf{u}_{A'}^T f_s([\mathbf{w}_S; \mathbf{z}]))}, \\ \pi_{A \rightarrow BC} &= \frac{\exp(\mathbf{u}_{BC}^T [\mathbf{w}_A; \mathbf{z}])}{\sum_{B', C' \in \mathcal{N} \cup \mathcal{P}} \exp(\mathbf{u}_{B'C'}^T [\mathbf{w}_A; \mathbf{z}])}, \\ \pi_{T \rightarrow w} &= \frac{\exp(\mathbf{u}_w^T f_t([\mathbf{w}_T; \mathbf{z}]))}{\sum_{w' \in \Sigma} \exp(\mathbf{u}_{w'}^T f_t([\mathbf{w}_T; \mathbf{z}]))}, \end{aligned}$$

where  $\mathbf{u}$  is a parameter vector,  $\mathbf{w}_N$  is a symbol embedding and  $N \in \{S\} \cup \mathcal{N} \cup \mathcal{P}$ .  $[\cdot; \cdot]$  indicates vector concatenation, and  $f_s(\cdot)$  and  $f_t(\cdot)$  encode the input into a vector (parameters are dropped for simplicity).

<sup>2</sup>Strictly, CFGs do not distinguish nonterminals  $\mathcal{N}$  (constituent labels) from preterminals  $\mathcal{P}$  (part-of-speech tags). They are both treated as nonterminals.  $\mathcal{N}, \mathcal{P}, \Sigma$  satisfy  $\mathcal{N} \cap \mathcal{P} = \emptyset$  and  $(\mathcal{N} \cup \mathcal{P}) \cap \Sigma = \emptyset$ .

A C-PCFG defines a mixture of PCFGs (i.e., we can sample a set of PCFG parameters by sampling a vector  $\mathbf{z}$ ). It satisfies the context-free assumption conditioned on  $\mathbf{z}$  and thus admits exact inference for each given  $\mathbf{z}$ . Learning with C-PCFGs involves maximizing the log-likelihood of every observed sentence  $\mathbf{w} = w_1 w_2 \dots w_n$ :

$$\log p_\theta(\mathbf{w}) = \log \int_{\mathbf{z}} \sum_{t \in \mathcal{T}_G(\mathbf{w})} p_\theta(t|\mathbf{z}) p(\mathbf{z}) d\mathbf{z},$$

where  $\mathcal{T}_G(\mathbf{w})$  consists of all parses of the sentence  $\mathbf{w}$  under a PCFG  $\mathcal{G}$ . Though for each given  $\mathbf{z}$  the inner summation over parses can be efficiently computed using the inside algorithm (Baker, 1979), the integral over  $\mathbf{z}$  makes optimization intractable. Instead, C-PCFGs rely on variational inference and maximize the evidence lower bound (ELBO):

$$\log p_\theta(\mathbf{w}) \geq \text{ELBO}(\mathbf{w}; \phi, \theta) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{w})} [\log p_\theta(\mathbf{w}|\mathbf{z})] - \text{KL}[q_\phi(\mathbf{z}|\mathbf{w})||p(\mathbf{z})], \quad (1)$$

where  $q_\phi(\mathbf{z}|\mathbf{w})$  is a variational posterior, a neural network parameterized with  $\phi$ . The expected log-likelihood term is estimated via the reparameterization trick (Kingma et al., 2014); the KL term can be computed analytically when  $p(\mathbf{z})$  and  $q_\phi(\mathbf{z}|\mathbf{w})$  are normally distributed.

## 2.2 Visually grounded neural syntax learner

The visually grounded neural syntax learner (VG-NSL) comprises a parsing model and an image-text matching model. The parsing model is an easy-first parser (Goldberg and Elhadad, 2010). It builds a parse greedily in a bottom-up manner while at the same time producing a semantic representation for each constituent in the parse (i.e., its ‘embedding’). The parser is optimized through REINFORCE (Williams, 1992). The reward encourages merging two adjacent constituents if the merge results in a constituent that is *concrete*, i.e., if its semantic representations is predictive of the corresponding image, as measured with a matching function. We omit details of the parser and how the semantic representations of constituents are computed, as they are not relevant to our approach, and refer the reader to Shi et al. (2019).

However, as we will extend their image-text matching model, we explain this component of their approach more formally. In their work, this loss is used to learn the textual and visual representations. For every constituent  $c^{(i)}$  of a sentence

$\mathbf{w}^{(i)}$ , they define the following triplet hinge loss:

$$h(\mathbf{c}^{(i)}, \mathbf{v}^{(i)}) = \mathbb{E}_{\mathbf{c}'} \left[ m(\mathbf{c}', \mathbf{v}^{(i)}) - m(\mathbf{c}^{(i)}, \mathbf{v}^{(i)}) + \epsilon \right]_+ + \mathbb{E}_{\mathbf{v}'} \left[ m(\mathbf{c}^{(i)}, \mathbf{v}') - m(\mathbf{c}^{(i)}, \mathbf{v}^{(i)}) + \epsilon \right]_+, \quad (2)$$

where  $[\cdot]_+ = \max(0, \cdot)$ ,  $\epsilon$  is a positive margin,  $m(\mathbf{c}, \mathbf{v}) \triangleq \cos(\mathbf{c}, \mathbf{v})$  is the matching function measuring similarity between the constituent representation  $\mathbf{c}$  and the image representation  $\mathbf{v}$ . The expectation is taken with respect to ‘negative examples’,  $\mathbf{c}'$  and  $\mathbf{v}'$ . In practice, for efficiency reasons, a single representation of an image  $\mathbf{v}'$  and a single representation of a constituent (span)  $\mathbf{c}'$  from another example in the same batch are used as the negative examples. Intuitively, an aligned image-constituent pair  $(\mathbf{c}^{(i)}, \mathbf{v}^{(i)})$  should score higher than an unaligned one  $((\mathbf{c}', \mathbf{v}^{(i)})$  or  $(\mathbf{c}^{(i)}, \mathbf{v}')$ .

The total loss for an image-sentence pair  $(\mathbf{v}^{(i)}, \mathbf{w}^{(i)})$  is obtained by summing losses for all constituents in a tree  $t^{(i)}$ , sampled from the parsing model (we write  $c^{(i)} \in t^{(i)}$ ):

$$\hat{s}(\mathbf{v}^{(i)}, \mathbf{w}^{(i)}) = \sum_{c^{(i)} \in t^{(i)}} h(\mathbf{c}^{(i)}, \mathbf{v}^{(i)}). \quad (3)$$

In their work, training alternates between optimizing the parser using rewards (relying on image and text representations) and optimizing the image-text matching model to refine image and text representations (relying on the fixed parsing model). Once trained, the parser can be directly applied to raw text, i.e., images are not used at test time.

## 2.3 Limitations of the VG-NSL framework

While straightforward, there are several practical issues inhibiting the visually grounded learning framework. First, contrastive learning implicitly assumes that every constituent of a sentence has its visual representation in the aligned image. However, it is not guaranteed in practice and would result in noisy reward signals. Besides, the loss in Equation 2 (and a similar component in the reward, see Shi et al. (2019)) focuses on constituents corresponding to short spans. Long spans, independently of their syntactic structure, tend to be sufficiently discriminative to distinguish the aligned image  $\mathbf{v}^{(i)}$  from an unaligned one. This implies that there is not much learning signal for such constituents. The tendency to focus on short spans and those more easily derivable from an image is evident from the results (Shi et al., 2019; Kojima et al., 2020). For example, their parser is accurate for noun phrases

(recall 79.6%), which are often short for captions, but performs poorly on verb phrases (recall 26.2%) which have longer spans, more complex compositionally and also harder to predict from images (see our analysis in Section 4.3.2). While there may be ways to mitigate some of these issues, we believe that any image-text matching loss alone is unlikely to provide sufficient learning signal to accurately capture all aspects of syntax. Instead of resorting to language-specific inductive biases as done by Shi et al. (2019) (i.e., head-initial bias (Baker, 2008) of English), we propose to complement the image-text matching loss with the objective derived from the unaligned text (i.e., log-likelihood), jointly training a parser to both explain the raw language data and the alignment with images.

Moreover, their learning is likely to suffer from large variance in gradient estimation as their parser does not admit tractable estimation of the partition function, and thus they have to rely on sampling decisions. This will be even more of a problem if we would attempt to use it in the joint learning setup. Also note that similar parsing models do not yield linguistically-plausible structures when used in the conventional (i.e., non-grounded) grammar-induction set-ups (Williams et al., 2018; Havrylov et al., 2019).

In the next section, we will use compound PCFGs and describe an improved visually grounded learning framework that can tackle these issues neatly.

### 3 Visually grounded compound PCFGs

We use compound PCFGs (Kim et al., 2019a) and develop visually-grounded compound PCFGs (VC-PCFGs) within the contrastive learning framework. Instead of sampling a tree and computing a point estimate of the image-text matching loss, we can compute the expected image-text matching loss under a tree distribution and use end-to-end contrastive learning (Section 3.1). Since it is inefficient to compute constituent representations relying on the chart, we will introduce an additional textual representation model to encode constituents (Section 3.2). Moreover, VC-PCFGs let us additionally optimize a language modeling objective, complementing the visually grounded contrastive learning (Section 3.3).

### 3.1 End-to-end contrastive learning

In the visually grounded grammar induction framework, the parsing model is optimized through learning signals derived from the alignment of images and constituents, as scored by the image-text matching model. Denoting a set of image representations by  $\mathcal{V} = \{\mathbf{v}^{(i)}\}$  and the corresponding set of sentences by  $\mathcal{W} = \{\mathbf{w}^{(i)}\}$ , the image-text matching model is optimized via contrastive learning:

$$\mathcal{L}(\mathcal{V}, \mathcal{W}; \phi, \theta) = \sum_i s(\mathbf{v}^{(i)}, \mathbf{w}^{(i)}). \quad (4)$$

We define  $s(\mathbf{v}^{(i)}, \mathbf{w}^{(i)})$  as the loss of aligning  $\mathbf{v}^{(i)}$  and  $\mathbf{w}^{(i)}$ . In VG-NSL, it is estimated via point estimation (see Equation 3). While in VC-PCFGs, given an aligned image-sentence pair  $(\mathbf{v}, \mathbf{w})$ , we compute the expected image-sentence matching loss under a tree distribution  $p_\theta(t|\mathbf{w})$ , leading to an end-to-end contrastive learning:

$$s(\mathbf{v}, \mathbf{w}) = \mathbb{E}_{p_\theta(t|\mathbf{w})} \sum_{c \in t} h(\mathbf{c}, \mathbf{v}), \quad (5)$$

where  $h(\mathbf{c}, \mathbf{v})$  is the hinge loss of aligning the unlabeled constituent  $c$  and the image  $\mathbf{v}$  (defined in Equation 2). Minimizing the hinge loss encourages an aligned image-constituent pair to rank higher than any unaligned one. Expanding the right-hand side of Equation 5

$$\begin{aligned} s(\mathbf{v}, \mathbf{w}) &= \sum_{t \in \mathcal{T}_G(\mathbf{w})} p_\theta(t|\mathbf{w}) \sum_{c \in t} h(\mathbf{c}, \mathbf{v}) \\ &= \sum_{c \in \mathbf{w}} \underbrace{\sum_{t \in \mathcal{T}_G(\mathbf{w})} \mathbb{I}_{\{c \in t\}} p_\theta(t|\mathbf{w})}_{p(c|\mathbf{w}): \text{marginal of the span } c} h(\mathbf{c}, \mathbf{v}) \\ &= \sum_{c \in \mathbf{w}} p(c|\mathbf{w}) h(\mathbf{c}, \mathbf{v}), \end{aligned} \quad (6)$$

where  $p(c|\mathbf{w})$  is the conditional probability (i.e., marginal) of the span  $c$  given  $\mathbf{w}$ . It can be efficiently computed with the inside algorithm and automatic differentiation (Eisner, 2016).

### 3.2 Span representation

Estimation of the expected image-text matching scores relies on span representations. Ideally, a span representation should encode semantics of a span with its computation guided by its syntactic structure (Socher et al., 2013). The reliance on the predicted tree structure will result in propagating learning signals derived from the alignment of



images and sentences back to the parser. To realize this desideratum, we could follow the inside algorithm and recursively compose span representations (Le and Zuidema, 2015; Stern et al., 2017; Drozdov et al., 2019), which is, however, time- and memory-inefficient in practice.

Instead, we produce span representations largely independently of the parser, as we will explain below. The only way the parser model influences this representation is through the predicted constituent label: we use its distribution to compute the representation.<sup>3</sup>

Specifically, as a trade-off for a better training efficiency, we adopt a single-layer BiLSTM to encode spans. A mean-pooling layer is applied over the hidden states  $\mathbf{h}$  of the BiLSTM and followed by a label-specific affine transformation  $f_k(\cdot)$  to produce a label-specific span representation  $\mathbf{c}_k$ . Take a span  $c_{i,j} = w_i \dots w_j$  ( $0 < i < j \leq n$ ):

$$\mathbf{c}_k = f_k\left(\frac{1}{j-i+1} \sum_{l=i}^j \mathbf{h}_l\right). \quad (7)$$

The BiLSTM encoding model operates at the span level and encodes semantics of a span. Unlike using a single sentence-level (Bi)LSTM encoder, it guarantees that no information from words outside of the span leaks into its representations. More importantly, it can run in  $\mathcal{O}(n)$  for a sentence of length  $n$  with a parallel implementation. While the produced representation does not reflect the structural decisions made by the parser, it can be sensitive to word order and may be affected by its syntactic structure (Blevins et al., 2018).

In order to compute the representation of unlabeled constituent  $\mathbf{c}$ , we average the label-specific span representation  $\mathbf{c}_k$  under the distribution of labels defined by the parser:

$$\mathbf{c} = \sum_{k=1}^K p(k|\mathbf{c}, \mathbf{w}) \mathbf{c}_k, \quad (8)$$

where  $p(k|\mathbf{c}, \mathbf{w})$  is the probability that the span  $\mathbf{c}$  has label  $k$ , conditioned on having this constituent span in the tree.

To further reduce computation we estimate the matching loss only using the  $\frac{n(n-1)}{4}$  shortest spans for a sentence of length  $n$ . Thus the image-text alignment loss will focus on small constituents.

<sup>3</sup>Intuitively, the key learning signal for the parser in our model comes through the marginals in Equation 6, not through the span representation.

This is the case anyway (see discussion in Section 2.3), so we expect that this simplification would not hurt model performance significantly.

### 3.3 Joint objective

Rather than simply optimizing the contrastive learning objective, we additionally maximize the log-likelihood of text data. As with C-PCFGs, we optimize the ELBO:

$$\mathcal{L}(\mathcal{W}; \phi, \theta) = - \sum_{\mathbf{w} \in \mathcal{W}} \text{ELBO}(\mathbf{w}; \phi, \theta). \quad (9)$$

This learning objective complements contrastive learning. As contrastive learning optimizes a parser by solely matching images and constituents, the parser would only focus on simple and local constituents (e.g., short NPs). Moreover, in practice, since not every constituent can be grounded in an image, contrastive learning would suffer from misleading or ambiguous learning signals.

To summarize, the overall loss function is

$$\mathcal{J}(\phi, \theta) = \mathcal{L}(\mathcal{W}; \phi, \theta) + \alpha \cdot \mathcal{L}(\mathcal{V}, \mathcal{W}; \phi, \theta), \quad (10)$$

where  $\alpha$  is a hyper-parameter balancing the relative importance of the contrastive learning.

### 3.4 Parsing

The parser can be directly used to parse raw text after training, without requiring access to visual groundings. Parsing seeks for the most probable parse  $t^*$  of  $\mathbf{w}$ :

$$t^* = \operatorname{argmax}_t \int_{\mathbf{z}} p_\theta(t|\mathbf{w}, \mathbf{z}) p_\theta(\mathbf{z}|\mathbf{w}) d\mathbf{z}.$$

Still, though the maximum a posterior (MAP) inference over  $p_\theta(t|\mathbf{w})$  can be solved by the CYK algorithm (Kasami, 1966; Younger, 1967), inference becomes intractable when introducing into  $\mathbf{z}$ . The MAP inference is instead approximated by

$$t^* \approx \operatorname{argmax}_t \int_{\mathbf{z}} p_\theta(t|\mathbf{w}, \mathbf{z}) \delta(\mathbf{z} - \boldsymbol{\mu}_\phi(\mathbf{w})) d\mathbf{z},$$

where  $\delta(\cdot)$  is the Dirac delta function and  $\boldsymbol{\mu}_\phi(\mathbf{w})$  is the mean vector of the variational posterior  $q_\phi(\mathbf{z}|\mathbf{w})$ . As  $\delta(\cdot)$  has zero mass everywhere but at the mode  $\boldsymbol{\mu}_\phi(\mathbf{w})$ , it is equivalently solving  $\operatorname{argmax}_t p_\theta(t|\mathbf{w}, \boldsymbol{\mu}_\phi(\mathbf{w}))$ .

## 4 Experiments

### 4.1 Datasets and evaluation

**Datasets:** We use MSCOCO (Lin et al., 2014). It consists of 82,783 training images, 1,000 validation images, and 1,000 test images. Each image is associated with 5 caption sentences. We encode images into 2048-dimensional vectors using the pre-trained ResNet-101 (He et al., 2016). At test time, only captions are used. We follow Shi et al. (2019) and parse test captions with Benepar (Kitaev and Klein, 2018). We use the same data preprocessing<sup>4</sup> as in Shen et al. (2019) and Kim et al. (2019a), where punctuation is removed from all data, and the top 10,000 frequent words in training sentences are kept as the vocabulary.

**Evaluation:** We mainly compare VC-PCFGs with VG-NSL (Shi et al., 2019). To verify the effectiveness of the use of visual groundings, we also compare our model with a C-PCFG trained only on the training captions. All models are run four times with different random seeds and for at most 15 epochs with early stopping (i.e., the image-caption loss / perplexity on the validation captions does not decrease). We report both averaged corpus-level F1 and averaged sentence-level F1 numbers as well as the unbiased standard deviations.

### 4.2 Settings and hyperparameters

We adopt parameter settings suggested by the authors for the baseline models. For VG-NSL we run the authors’ code.<sup>5</sup> We re-implement C-PCFG using automatic differentiation (Eisner, 2016) to speed up training. Our VC-PCFG comprises a parsing model and an image-text matching model. The parsing model has the same parameters as the baseline C-PCFG; the image-text matching model has the same parameters as the baseline VG-NSL. Concretely, the parsing model has 30 nonterminals and 60 preterminals. Each of them is represented by a 256-dimensional vector. The inference model  $q_\phi(\mathbf{z}|\mathbf{w})$  uses a single-layer BiLSTM. It has a 512-dimensional hidden state and relies on 512-dimensional word embeddings. We apply a max-pooling layer over the hidden states of the BiLSTM and then obtain 64-dimensional mean vectors  $\mu_\phi(\mathbf{w})$  and log-variances  $\log \sigma_\phi(\mathbf{w})$  by using an affine layer. The image-text matching model projects visual features into 512-dimensional feature vectors and

<sup>4</sup><https://git.io/JfV6J>.

<sup>5</sup><https://git.io/Jf3nn>.

encodes spans as 512-dimensional vectors. Our span representation model is another single-layer BiLSTM, with the same hyperparameters as in the inference model.  $\alpha$  for visually grounded learning is set to 0.001. We implement VC-PCFG relying on Torch-Struct (Rush, 2020), and optimize it using Adam (Kingma and Ba, 2015) with the learning rate set to 0.01,  $\beta_1 = 0.75$ , and  $\beta_2 = 0.999$ . All parameters are initialized with Xavier uniform initializer (Glorot and Bengio, 2010).

### 4.3 Results and analysis

#### 4.3.1 Main results

Our model outperforms all baselines according to both corpus-level F1 and sentence-level F1 (see Table 1). Notably, it surpasses VG-NSL+HI by 10% F1.<sup>6</sup> The right branching model is a strong baseline on image captions, as observed previously on the WSJ corpus, including in recent work (Shen et al., 2018; Kim et al., 2019a). Comparing with C-PCFG, which is trained solely on captions, VC-PCFG achieves a much higher mean F1 (+5.7% F1), demonstrating the informativeness of visual groundings. However, VC-PCFG suffers from a larger variance presumably because the joint objective is harder to optimize. Visually grounded contrastive learning (w/o LM) has a mean F1 50.5%. It is further improved to 59.4% when additionally optimizing the language modeling objective.

Moreover, we show recall on six frequent constituent labels (NP, VP, PP, SBAR, ADJP, ADVP) in the test captions. Unsurprisingly, VG-NSL is best on NPs because the matching-based reward signals optimize it to focus only on short and concrete NPs (recall 64.3%). It performs poorly on other constituent labels such as VPs (recall 28.1%). In contrast, VC-PCFG exhibits a relatively even performance across constituent labels, e.g., it is most accurate on SBARs and ADVPs and works fairly well on VPs (recall 83.2%). Meanwhile, it improves over C-PCFG for NPs, which are usually short and ‘concrete’, once again confirming the benefits of using visual groundings. Visually grounded contrastive learning (w/o LM) tends to behave like

<sup>6</sup>We run the code of Shi et al. (2019) and train VG-NSL and VG-NSL+HI on the training captions with punctuation removed. This is considered a more challenging setting as punctuation signals the boundaries of constituents and makes it easy for parsers to derive constituents. At test time, as a common practice (Shen et al., 2018, 2019; Kim et al., 2019a), we discard punctuation and ignore trivial single-word and sentence-level spans. We notice that including sentence-level spans can improve the F1 of VG-NSL to around 49%.

Model	NP	VP	PP	SBAR	ADJP	ADVP	C-F1	S-F1
Left Branching	33.2	0.0	0.1	0.0	4.9	0.0	15.1	15.7
Right Branching	37.5	<b>94.5</b>	71.1	<b>97.8</b>	20.9	79.1	51.0	51.8
Random Trees	32.8 $\pm$ 0.5	18.4 $\pm$ 0.4	24.4 $\pm$ 0.3	17.7 $\pm$ 1.7	26.8 $\pm$ 2.6	20.9 $\pm$ 1.5	24.2 $\pm$ 0.3	24.6 $\pm$ 0.2
C-PCFG	43.0 $\pm$ 8.6	85.0 $\pm$ 2.6	78.4 $\pm$ 5.6	90.6 $\pm$ 2.1	36.6 $\pm$ 2.1	<b>87.4</b> $\pm$ 1.0	53.6 $\pm$ 4.7	53.7 $\pm$ 4.6
VG-NSL <sup>†</sup>	79.6 $\pm$ 0.4	26.2 $\pm$ 0.4	42.0 $\pm$ 0.6		22.0 $\pm$ 0.4		50.4 $\pm$ 0.3	
VG-NSL+HI <sup>†</sup>	74.6 $\pm$ 0.5	32.5 $\pm$ 1.5	66.5 $\pm$ 1.2		21.7 $\pm$ 1.1		53.3 $\pm$ 0.2	
VG-NSL <sup>*</sup>	<b>64.3</b> $\pm$ 1.1	28.1 $\pm$ 0.5	32.2 $\pm$ 1.1	16.9 $\pm$ 3.2	13.2 $\pm$ 1.5	5.6 $\pm$ 0.3	41.5 $\pm$ 0.5	41.8 $\pm$ 0.5
VG-NSL+HI <sup>*</sup>	61.0 $\pm$ 0.2	33.5 $\pm$ 1.6	62.7 $\pm$ 0.6	42.0 $\pm$ 5.1	13.9 $\pm$ 0.6	65.9 $\pm$ 2.5	48.8 $\pm$ 0.4	49.4 $\pm$ 0.5
VC-PCFG (ours)	54.9 $\pm$ 14	83.2 $\pm$ 3.9	<b>80.9</b> $\pm$ 7.9	89.0 $\pm$ 2.0	<b>38.8</b> $\pm$ 2.5	86.3 $\pm$ 4.1	<b>59.3</b> $\pm$ 8.2	<b>59.4</b> $\pm$ 8.3
w/o LM	35.6 $\pm$ 3.7	93.4 $\pm$ 2.1	70.1 $\pm$ 2.0	95.9 $\pm$ 3.9	20.6 $\pm$ 0.8	78.0 $\pm$ 2.2	49.7 $\pm$ 2.6	50.5 $\pm$ 2.5

Table 1: Recall on six frequent constituent labels (NP, VP, PP, SBAR, ADJP, ADVP) in the MSCOCO test captions and corpus-level F1 (C-F1) and sentence-level F1 (S-F1) results. The best mean number in each column is in bold. <sup>†</sup> indicates results reported by Shi et al. (2019). <sup>\*</sup> denotes results obtained by running their code. Notice that the results from Shi et al. (2019) are not comparable to ours because they keep punctuation and include trivial sentence-level spans in evaluation.

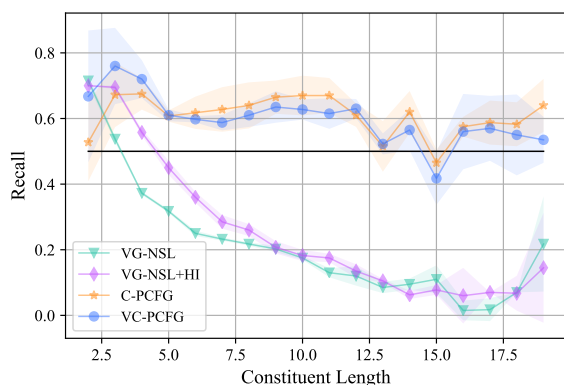


Figure 1: Recall broken down by constituent length.

the right branching baseline. Additionally optimizing the language modeling objective brings a huge improvement for NPs (+19.3% recall).

### 4.3.2 Analysis

We analyze model performance for constituents of different lengths (Figure 1). As expected, VG-NSL becomes weaker as constituent length increases, and the drop is very dramatic. C-PCFG and its grounded version VC-PCFG consistently outperform VG-NSL on constituents longer than four tokens and display a more even performance across constituent lengths. Meanwhile, VC-PCFG beats C-PCFG on constituents of length below 5, confirming that visual groundings are beneficial for short spans. We further plot the distribution over constituent length for different phrase types (Figure 2) and find that around 75% constituents in our dataset are shorter than six tokens, and 60% of them are NPs. Thus, it is not surprising that the im-

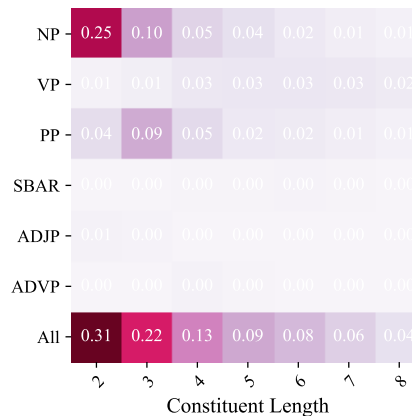


Figure 2: Label distribution over constituent length. All denotes frequencies of constituent lengths. Zero frequencies are due to the limited numerical precision.

provement on NPs, brought by visually grounded learning, has a large impact on the overall performance.

Next, we analyze induced tree structures. We compare model predictions against gold trees, left branching trees, and right branching trees. As there is little performance difference between corpus-level F1 and sentence-level F1, we focus on sentence-level F1 in this analysis. We report self F1 (Williams et al., 2018) to show model consistency across runs. The self F1 is computed by averaging over six model pairs from four different runs. All results are presented in Table 2. Overall, all models have self F1 above 70%, indicating a relatively high consistency. We observe that using the head-initial bias pushes VG-NSL closer to the right-branching baseline, while visual grounded learning

Model	Gold	Left	Right	Self
VG-NSL	41.8	28.3	20.6	84.3
VG-NSL+HI	49.4	24.5	29.2	88.6
C-PCFG	53.7	1.3	53.6	77.3
VC-PCFG	59.4	4.4	48.5	71.1

Table 2: Average sentence-level F1 results against gold trees (Gold), left branching trees (Left), right branching trees (Right), and self F1 (Self) (Williams et al., 2018).

leads to improvements over C-PCFG, forcing VC-PCFG to deviate from the default right-branching behaviour.

Finally, we test VG-NSL+HI and VC-PCFG on 50 manually annotated captions released by Shi et al. (2019). VC-PCFG achieves a mean F1 62.7%, surpassing VG-NSL+HI by 12.1% F1. In Figure 3 we visualize a parse tree predicted by the best run of VC-PCFG. We can see that VC-PCFG identifies most NPs but makes mistakes in PP attachment and consequently fails to identify the VP.

## 5 Related work

**Grammar Induction** has a long history in computational linguistics. Following observations that direct optimization of log-likelihood with the Expectation Maximization algorithm (Lari and Young, 1990) is not effective at producing effective grammars, a number of approaches have been developed, embodying various inductive biases or assumption about the language structure and its relation to surface realizations (Klein and Manning, 2002; Smith and Eisner, 2005; Cohen and Smith, 2009; Spitzkovsky et al., 2010). The recent advances in the area have been brought by flexible neural models (Jin et al., 2019; Kim et al., 2019a,b; Drozdov et al., 2019). All these methods, with the exception of Shi et al. (2019), rely solely on text.

**Visually grounded learning** is motivated by the observation that natural language is grounded in perceptual experiences (Steels, 1998; Barsalou, 1999; Fincher-Kiefer, 2001; Roy, 2002; Bisk et al., 2020). It has been shown effective in word representation learning (Bruni et al., 2014; Silberer and Lapata, 2014; Lazaridou et al., 2015) and sentence representation learning (Kiela et al., 2018; Bordes et al., 2019). All this work uses visual images as perceptual experience of language and exploits visual semantics derived from images to improve continuous vector representations of language. In contrast, we induce structured representations, discrete

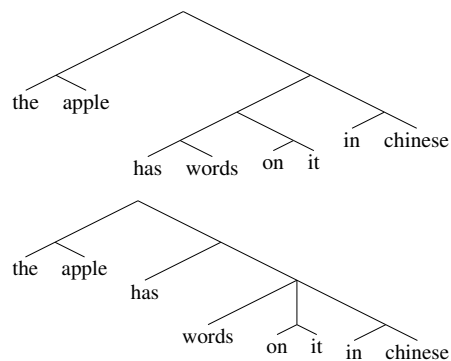


Figure 3: Upper: A parse output by the best run of VC-PCFG. Bottom: The corresponding gold tree.

tree structure of language, by using visual groundings. We propose a model for the task within the contrastive learning framework. Learning involves estimating *concreteness* of spans, which generalizes word-level concreteness (Turney et al., 2011; Kiela et al., 2014).

In the vision and machine learning community, unsupervised **induction of structured image representations** (aka scene graphs or world models) has been receiving increasing attention (Eslami et al., 2016; Burgess et al., 2019; Kipf et al., 2020). However, they typically rely solely on visual signal. An interesting extension of our work would be to consider joint induction of structured representations of images and text while guiding learning by an alignment loss.

## 6 Conclusion

We have presented visually-grounded compound PCFGs (VC-PCFGs) that use compound PCFGs and generalize the visually grounded grammar learning framework. VC-PCFGs exploit visual groundings via contrastive learning, with learning signals derived from minimizing an image-text alignment loss. To tackle the issues of misleading and insufficient learning signals from purely agreement-based learning, we propose to complement the image-text alignment loss with a loss defined on unlabeled text. We resort to using compound PCFGs which enables us to complement the alignment loss with a language modeling objective, resulting in a fully-differentiable end-to-end visually grounded learning. We empirically show that our VC-PCFGs are superior to models that are trained only through visually grounded learning or only relying on text.



## Acknowledgments

We would like to thank anonymous reviewers for their suggestions and comments. The project was supported by the European Research Council (ERC Starting Grant BroadSem 678254) and the Dutch National Science Foundation (NWO VIDI 639.022.518).

## References

- J. K. Baker. 1979. [Trainable grammars for speech recognition](#). *The Journal of the Acoustical Society of America*, 65(S1):S132–S132.
- Mark C Baker. 2008. *The atoms of language: The mind’s hidden rules of grammar*. Basic books.
- Lawrence W. Barsalou. 1999. [Perceptual symbol systems](#). *Behavioral and Brain Sciences*, 22(4):577–660.
- Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, et al. 2020. [Experience grounds language](#). *arXiv preprint arXiv:2004.10151*.
- Terra Blevins, Omer Levy, and Luke Zettlemoyer. 2018. [Deep RNNs encode soft hierarchical syntax](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 14–19, Melbourne, Australia. Association for Computational Linguistics.
- Patrick Bordes, Eloi Zabolocki, Laure Soulier, Benjamin Piwowarski, and Patrick Gallinari. 2019. [Incorporating visual semantics into sentence representations within a grounded space](#). 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 696–707, Hong Kong, China. Association for Computational Linguistics.
- Elia Bruni, Nam Khanh Tran, and Marco Baroni. 2014. [Multimodal distributional semantics](#). *J. Artif. Int. Res.*, 49(1):1–47.
- Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. 2019. [Monet: Unsupervised scene decomposition and representation](#). *arXiv preprint arXiv:1901.11390*.
- Glenn Carroll and Eugene Charniak. 1992. *Two experiments on learning probabilistic dependency grammars from corpora*. AAAI Press.
- Alexander Clark. 2001. [Unsupervised induction of stochastic context-free grammars using distributional clustering](#). In *Proceedings of the ACL 2001 Workshop on Computational Natural Language Learning (ConLL)*.
- Shay Cohen and Noah A. Smith. 2009. [Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction](#). In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 74–82, Boulder, Colorado. Association for Computational Linguistics.
- 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, Minneapolis, Minnesota. Association for Computational Linguistics.
- Jason Eisner. 2016. [Inside-outside and forward-backward algorithms are just backprop \(tutorial paper\)](#). In *Proceedings of the Workshop on Structured Prediction for NLP*, pages 1–17, Austin, TX. Association for Computational Linguistics.
- S. M. Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, koray kavukcuoglu, and Geoffrey E Hinton. 2016. [Attend, infer, repeat: Fast scene understanding with generative models](#). In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3225–3233. Curran Associates, Inc.
- Rebecca Fincher-Kiefer. 2001. [Perceptual components of situation models](#). *Memory & Cognition*, 29(2):336–343.
- Xavier Glorot and Yoshua Bengio. 2010. [Understanding the difficulty of training deep feedforward neural networks](#). In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. PMLR.
- Yoav Goldberg and Michael Elhadad. 2010. [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. Association for Computational Linguistics.
- Serhii Havrylov, Germán Kruszewski, and Armand Joulin. 2019. [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. Association for Computational Linguistics.
- 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.
- Lifeng Jin, Finale Doshi-Velez, Timothy Miller, Lane Schwartz, and William Schuler. 2019. [Unsupervised learning of PCFGs with normalizing flow](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2442–2452, Florence, Italy. Association for Computational Linguistics.
- Tadao Kasami. 1966. [An efficient recognition and syntax-analysis algorithm for context-free languages](#). *Coordinated Science Laboratory Report no. R-257*.
- Douwe Kiela, Alexis Conneau, Allan Jabri, and Maximilian Nickel. 2018. [Learning visually grounded sentence representations](#). 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 408–418, New Orleans, Louisiana. Association for Computational Linguistics.
- Douwe Kiela, Felix Hill, Anna Korhonen, and Stephen Clark. 2014. [Improving multi-modal representations using image dispersion: Why less is sometimes more](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 835–841, Baltimore, Maryland. 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. Association for Computational Linguistics.
- Yoon Kim, Alexander Rush, Lei Yu, Adhiguna Kuncoro, Chris Dyer, and Gábor Melis. 2019b. [Unsupervised recurrent neural network grammars](#). 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 1105–1117, Minneapolis, Minnesota. Association for Computational Linguistics.
- Diederik P Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *Proceedings of the 3rd International Conference on Learning Representations*.
- Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. 2014. [Semi-supervised learning with deep generative models](#). In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3581–3589. Curran Associates, Inc.
- Thomas Kipf, Elise van der Pol, and Max Welling. 2020. [Contrastive learning of structured world models](#). In *International Conference on Learning Representations*.
- Nikita Kitaev and Dan Klein. 2018. [Constituency parsing with a self-attentive encoder](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.
- Dan Klein and Christopher D. Manning. 2002. [A generative constituent-context model for improved grammar induction](#). In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 128–135, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Noriyuki Kojima, Hadar Averbuch-Elor, Alexander Rush, and Yoav Artzi. 2020. [What is learned in visually grounded neural syntax acquisition](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2615–2635, Online. Association for Computational Linguistics.
- K. Lari and S.J. Young. 1990. [The estimation of stochastic context-free grammars using the inside-outside algorithm](#). *Computer Speech and Language*, 4(1):35 – 56.
- Angeliki Lazaridou, Georgiana Dinu, and Marco Baroni. 2015. [Hubness and pollution: Delving into cross-space mapping for zero-shot learning](#). 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 270–280, Beijing, China. Association for Computational Linguistics.
- Phong Le and Willem Zuidema. 2015. [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. Association for Computational Linguistics.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. [Microsoft coco: Common objects in context](#). In *Computer Vision – ECCV 2014*, pages 740–755, Cham. Springer International Publishing.
- Herbert Robbins. 1951. [Asymptotically subminimax solutions of compound statistical decision problems](#). In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 131–149, Berkeley, Calif. University of California Press.

- Deb K. Roy. 2002. [Learning visually grounded words and syntax for a scene description task](#). *Computer Speech and Language*, 16(3):353 – 385. Spoken Language Generation.
- Alexander Rush. 2020. [Torch-struct: Deep structured prediction library](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 335–342, Online. Association for Computational Linguistics.
- 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 Sordani, and Aaron Courville. 2019. [Ordered neurons: Integrating tree structures into recurrent neural networks](#). In *International Conference on Learning Representations*.
- Haoyue Shi, Jiayuan Mao, Kevin Gimpel, and Karen Livescu. 2019. [Visually grounded neural syntax acquisition](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1842–1861, Florence, Italy. Association for Computational Linguistics.
- Carina Silberer and Mirella Lapata. 2014. [Learning grounded meaning representations with autoencoders](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 721–732, Baltimore, Maryland. Association for Computational Linguistics.
- Noah A. Smith and Jason Eisner. 2005. [Guiding unsupervised grammar induction using contrastive estimation](#). In *In Proc. of IJCAI Workshop on Grammatical Inference Applications*, pages 73–82.
- Richard Socher, John Bauer, Christopher D. Manning, and Andrew Y. Ng. 2013. [Parsing with compositional vector grammars](#). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 455–465, Sofia, Bulgaria. Association for Computational Linguistics.
- Valentin I. Spitkovsky, Hiyan Alshawi, Daniel Jurafsky, and Christopher D. Manning. 2010. [Viterbi training improves unsupervised dependency parsing](#). In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 9–17, Uppsala, Sweden. Association for Computational Linguistics.
- Luc Steels. 1998. [The origins of syntax in visually grounded robotic agents](#). *Artificial Intelligence*, 103(1):133 – 156. Artificial Intelligence 40 years later.
- Mitchell Stern, Jacob Andreas, and Dan Klein. 2017. [A minimal span-based neural constituency parser](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 818–827, Vancouver, Canada. Association for Computational Linguistics.
- Peter Turney, Yair Neuman, Dan Assaf, and Yohai Cohen. 2011. [Literal and metaphorical sense identification through concrete and abstract context](#). In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 680–690, Edinburgh, Scotland, UK. Association for Computational Linguistics.
- Adina Williams, Andrew Drozdov, and Samuel R. Bowman. 2018. [Do latent tree learning models identify meaningful structure in sentences?](#) *Transactions of the Association for Computational Linguistics*, 6:253–267.
- Ronald J. Williams. 1992. [Simple statistical gradient-following algorithms for connectionist reinforcement learning](#). *Machine Learning*, 8(3):229–256.
- Daniel H. Younger. 1967. [Recognition and parsing of context-free languages in time n<sup>3</sup>](#). *Information and Control*, 10(2):189 – 208.