Dependency Induction Through the Lens of Visual Perception

Ruisi Su[1] **Shruti Rijhwani**[2] **Hao Zhu**[2] **Junxian He**[2]
Xinyu Wang[2] **Yonatan Bisk**[2] **Graham Neubig**[2]
[1]Electrical and Computer Engineering, Carnegie Mellon University
[2]Language Technologies Institute, Carnegie Mellon University
ruisis@alumni.cmu.edu, zhuhao@cmu.edu

Abstract

Most previous work on grammar induction focuses on learning phrasal or dependency structure purely from text. However, because the signal provided by text alone is limited, recently introduced visually grounded syntax models make use of multimodal information leading to improved performance in constituency grammar induction. However, as compared to dependency grammars, constituency grammars do not provide a straightforward way to incorporate visual information without enforcing language-specific heuristics. In this paper, we propose an unsupervised grammar induction model that leverages word concreteness and a structural vision-based heuristic to jointly learn constituency-structure and dependency-structure grammars. Our experiments find that concreteness is a strong indicator for learning dependency grammars, improving the direct attachment score (DAS) by over 50% as compared to state-of-the-art models trained on pure text. Next, we propose an extension of our model that leverages both word concreteness and visual semantic role labels in constituency and dependency parsing. Our experiments show that the proposed extension outperforms the current state-of-the-art visually grounded models in constituency parsing even with a smaller grammar size.[1]

1 Introduction

Grammar induction aims to discover the underlying grammatical structure of a language from strings of symbols. Previous work has focused on two grammar formalisms: constituency and dependency grammars. Probabilistic context-free grammars (Charniak, 1996; Clark, 2001, PCFG) have been used for modeling probabilistic rules in a constituency grammar, including more recent approaches that combine PCFGs with neural models (Kim et al., 2019). The dependency grammar approach to syntactic modeling allows for a binary probabilistic treatment between each word and its head. This binary relationship introduces flexibility for modeling languages with rich morphology and relatively free word order languages (Jurafsky and Martin, 2009). A commonly used model for dependency induction is the dependency model with valence (Klein and Manning, 2004, DMV).

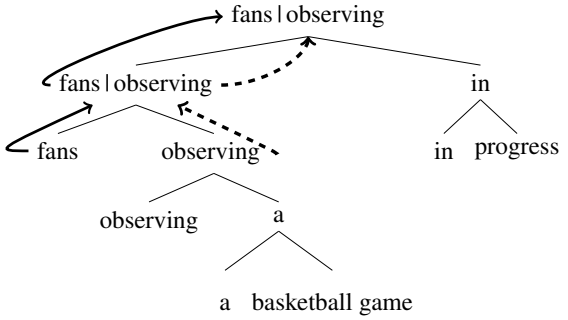


Figure 1: A learned tree structure with its lexicalized heads. Solid lines represent the constituents. Arrows represent the propagation path of the lexical heads from the constituents to the root. ⇢ and → are the paths of the neural L-PCFG which uses pure text and our proposed CONCRETE L-PCFG which uses word concreteness, respectively. neural L-PCFG incorrectly predicts *observing* as the root of the sentence, while CONCRETE L-PCFG selects the correct root *fans*.

The two formalisms are complementary and unified models have been shown to achieve higher induction accuracy than models that are trained for either constituency or dependency parsing alone. For instance, the lexicalized PCFG (Collins, 2003) is a supervised model that jointly learns the two formalisms by associating a word and a part-of-speech tag with each non-terminal in the tree of the PCFG. A more recent variant, the neural L-PCFG (Zhu et al., 2020) is an unsupervised model that uses lexical information to jointly learn both constituency-structure and dependency-structure grammars, achieving state-of-the-art performance in both formalisms.

[1]Code is available at https://github.com/ruisi-su/concrete_dep.

Proceedings of the 25th Conference on Computational Natural Language Learning (CoNLL), pages 17–26
November 10–11, 2021.

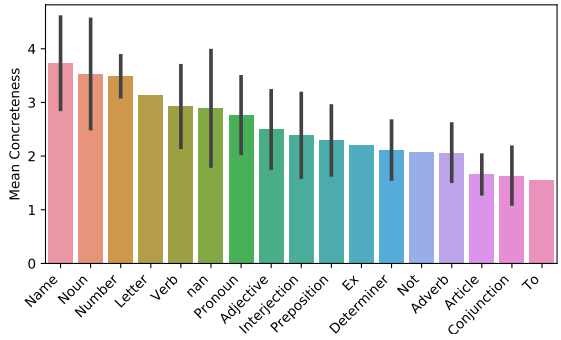


Figure 2: Mean word concreteness by dominant part-of-speech tags (Brysbaert et al., 2013) with standard deviations of the English MSCOCO captioning data (Lin et al., 2014). We exclude words that are unclassified or lack a dominant tag.

Besides the use of lexical information, grammar induction also benefits from a coarse understanding of the similarities in the syntactic structures shared by many human languages. Recent works showed that biases based on such syntactic universals can be applied to the posterior distribution of the dependency structures (Naseem et al., 2010; Cai et al., 2017; Li et al., 2019). Such universal linguistic rules can be helpful when part-of-speech tags are given to the grammar induction models. However, explicitly providing a ruleset can make non-terminals sensitive to their context, which is slightly contradictory to context-free grammar based methods, such as PCFGs. Thus, a more abstract method is needed to provide both guidance and flexibility to grammar induction models.

Nevertheless, inducing grammars entirely from pure text is a tall order, and arguably one not even tackled by humans themselves – humans learn language from the surrounding world (Bisk et al., 2020). Recent work, such as the Visually Grounded Neural Syntax Learner (Shi et al., 2019, VGNSL) show improved performance on constituency grammar induction by learning a shared embedding space for text and its corresponding image. However, Kojima et al. (2020) note that the improvement in performance is not a result of learning complex syntactic rules, and is rather of the model indirectly learning word *concreteness*, a concept that evaluates the degree to which a word refers to a perceptible entity (Brysbaert et al., 2013).

However, concisely describing the effect of word concreteness on the constituency structure of a sentence is non-trivial without resorting to language-specific heuristics such as the head-initial bias of the VGNSL (Shi et al., 2019). On the other hand, describing the effect of concreteness on *dependency* structure is more straightforward. In most languages, content words, such as nouns and verbs, tend to be more *concrete* as they are learned through experience via modalities (Brysbaert et al., 2013). Our analysis on the English MSCOCO dataset, shown in Figure 2, affirms this is true for English. Content words are also more likely to be heads than dependents in many dependency specifications (Nivre et al., 2016).

Motivated by these observations, in this paper, we incorporate the concept of concreteness in the unsupervised neural L-PCFG model (Zhu et al., 2020) at two levels: the word level and the phrase level. First, at the word level, we incorporate a prior to upweight dependency structures that have heads with higher word concreteness scores, which are human-rated numerical values for common English words (Brysbaert et al., 2013). Our experiments on the English MSCOCO dataset show that, at the word level, the word concreteness priors greatly improve the dependency induction performance over the neural L-PCFG, increasing the directed attachment accuracy (DAS) by 50% (Section 4). Additionally, we investigate the effect on predicted roots and find that the concreteness priors serve as a "short-cut" to bias perceptible entities towards becoming heads in the dependency parse. In the example in Figure 1, the CONCRETE L-PCFG selects the entity *fans* as the root word for the sentence, because it is more perceptible than the action *observing*. Adding word concreteness priors to the root encourages the subsequently learned rules to select the lexical head with higher word concreteness.

At the phrase level, we present a vision-based heuristic to exploit concreteness by connecting the word referents of the corresponding perceptible entities. These perceptible entities are extracted from images and aligned using a unified vision-language pre-training model (Zhou et al., 2020, VLP). Our experiments show the vision-based heuristic improves the model's constituency parsing performance. Finally, we present a model that combines the concreteness priors at both the word and phrase levels, which further improves the F1 score by 12% over the neural L-PCFG for the constituency parsing task.

In summary, this paper demonstrates that concreteness interacts with the neural L-PCFG differently at various levels: word-level concreteness

helps dependency induction in differentiating between content and function words, and the phrase-level vision-based heuristic helps pruning grammars that produce incorrect spans. Both levels help the neural L-PCFG to conceptualize the relationships between words that are highly associated with visually perceptible entities.

2 Background

This section briefly discusses the neural L-PCFG (Zhu et al., 2020), which jointly learns constituency-structure and dependency-structure grammars. We encourage readers to refer to the original paper for a more detailed description.

The neural L-PCFG is a neural parameterization of the lexicalized PCFG (Collins, 2003, L-PCFG) and it demonstrates improved performance on unsupervised constituency and dependency parsing. The phrase structures from a context-free grammar (CFG) are defined as a five-tuple $\mathcal{T} = (S, \mathcal{N}, \mathcal{P}, \Sigma, \mathcal{R})$, including the following set of rules $\mathcal{R}$:

$$\begin{aligned} S &\rightarrow A\,[\alpha]\,, && A \in \mathcal{N} \\ A\,[\alpha] &\rightarrow B\,[\alpha]\,C\,[\beta]\,, && A \in \mathcal{N}, B, C \in \mathcal{N} \cup \mathcal{P} \\ A\,[\alpha] &\rightarrow B\,[\beta]\,C\,[\alpha]\,, && A \in \mathcal{N}, B, C \in \mathcal{N} \cup \mathcal{P} \\ T\,[\alpha] &\rightarrow \alpha, && T \in \mathcal{P} \end{aligned}$$

where $\mathcal{N}$ and $\mathcal{P}$ represent the set of non-terminals and preterminals, respectively, and $\alpha, \beta \in \Sigma$ where Σ is the set of lexical items in grammar $\mathcal{T}$. The branching rules' scores rely on four components: (1) the probability of the root to the non-terminal A, (2) the word emission probability, (3) the probability of the headedness direction and the head-inheriting child conditioned on the parent non-terminal and head words, and (4) the probability of the non-inheriting child conditioned on the headedness direction, parent non-terminal, and head-inheriting child non-terminals.

From Zhu et al. (2020), the neural L-PCFG is a parameter-sharing method that allows more flexible parameterization than traditional L-PCFGs by conditioning the probabilities of production rules on the representations of non-terminals, preterminals, and lexical items. The neural L-PCFG also utilizes the compound probability distribution stemming from the compound PCFG (Kim et al., 2019), which showed promising results in estimating the parameters of the model based on natural linguistic differences in the inputting sentences. The latent compound variable $\boldsymbol{z}$ is sampled from a standard spherical Gaussian distribution, whose probability is denoted as $p_{\mathcal{N}(\mathbf{0},\boldsymbol{I})}(\boldsymbol{z})$. The log likelihood of a sentence $\boldsymbol{x}$ is obtained by marginalizing the compound variable:

$$\log p(\boldsymbol{x}) = \log \int_{\boldsymbol{z}} p_{\boldsymbol{z}}(\boldsymbol{x}) p_{\mathcal{N}(\mathbf{0},\boldsymbol{I})}(\boldsymbol{z})\, dz \qquad (1)$$

Because it is intractable to integrate over $\boldsymbol{z}$ in Equation 1, the evidence lower bound (ELBo) of the log likelihood is estimated by Monte-Carlo sampling and optimized during training of the neural L-PCFG (Zhu et al., 2020) using the mean and the variance vectors predicted by an inference network. The neural L-PCFG is trained to maximize the log likelihood of the entire corpus.

During inference, the most probable tree is approximated using the mean vector $\boldsymbol{\mu} = f_{\mu}(\boldsymbol{x})$ (predicted by the inference network) in a Dirac delta distribution $\delta(\boldsymbol{z} - \boldsymbol{\mu})$ instead of the real distribution $p(\boldsymbol{z}|\boldsymbol{x})$, which would be intractable:

$$\begin{aligned} \hat{t} &\approx \arg\max_{t \in \mathcal{T}_{\boldsymbol{x}}} \int_{\boldsymbol{z}} p_{\boldsymbol{z}} \delta(\boldsymbol{z} - \boldsymbol{\mu})\, dz \\ &= \arg\max_{t \in \mathcal{T}_{\boldsymbol{x}}} p_{\boldsymbol{\mu}}(t) \end{aligned}$$

The probability distribution over sentences $\boldsymbol{x}$ is then defined by:

$$p_{\boldsymbol{z}}(\boldsymbol{x}) = \sum_{t \in \mathcal{T}_{\boldsymbol{x}}} p_{\boldsymbol{z}}(t) = \frac{1}{\mathcal{Z}(\mathcal{T}, \boldsymbol{z})} \sum_{t \in \mathcal{T}_{\boldsymbol{x}}} \tilde{p}_{\boldsymbol{z}}(t)$$

in which $\mathcal{Z}(\mathcal{T}, \boldsymbol{z})$ is the normalizing factor, and $\sum_{t \in \mathcal{T}_{\boldsymbol{x}}} p_{\boldsymbol{z}}(t)$ is the sum of normalized probabilities of trees that might have generated $\boldsymbol{x}$. The unnormalized probability is calculated by exponentiating a scoring function: $\tilde{p}_{\boldsymbol{z}}(t) \propto \exp G_{\theta}(t, \boldsymbol{z})$, which assumes each rule is independent of the other rules in the parse tree t.

3 Dependency Induction Method

We propose three model variants: the word concreteness model, the vision-based heuristic model, and a model that combines both concreteness and the heuristic for inducing dependency structures and constituency structures.

3.1 Proposed Model

We extend the neural L-PCFG model to incorporate the concept of concreteness by adding two concrete-

(a) The example image

source: (a) kitchen (with) (a) stove oven (and) refrigerator	
target: overflowing refrigerator kitchen water	
alignment pair	scores
kitchen:kitchen	0.625
refrigerator:refrigerator	0.363
stove:overflowing	0.030
oven:water	0.006

(b) The example input and output of the alignment method.

Figure 3: An example of an image-caption pair from the MSCOCO test split. Figure 3b shows the source (captions) and target (semantic role labels) for the alignment method of Figure 3a. Words in parentheses are the stop words removed from the original caption. We use the SpaCy default list of stop words for English (Honnibal et al., 2020). The labels are ordered by the first term as the predicted activity followed by the set of the entities involved in the activity (Yatskar et al., 2016). The alignment pairs are ordered from the highest to the lowest scores.

ness priors to the context-free scoring function:

$$G_\theta(t, \boldsymbol{z}) = \sum_{i=1}^{k} g_\theta(r_i, \boldsymbol{z}) + \lambda_c h_i + \lambda_v \mathbf{1}_{\mathbf{v}}(s_i) \quad (2)$$

For a parsed tree t, $g_\theta(r_i, \boldsymbol{z})$ is the original scoring function of the neural L-PCFG (Section 2) that assigns a log-likelihood to rule r_i. λ_c is the hyperparameter used to control the effect of **the word-level concreteness prior** h_i of the current head word i, detailed in Section 3.2. Similarly, λ_v is the hyperparameter controlling the effort of **the phrase-level concreteness prior**. $\mathbf{1}_{\mathbf{v}}(s_i)$ is the indicator function $[s_i \in \mathcal{V}]$ that the parsed constituent s_i is in the set of the rewarded spans $\mathcal{V}$ generated using the vision-based coupling heuristic in Section 3.3.

3.2 Concrete L-PCFG

The first variant of the model we experiment with is denoted by CONCRETE L-PCFG. In this variant, we set $\lambda_v = 0$ in Equation 2, thus *only* incorporating the word concreteness prior into the neural L-PCFG. The CONCRETE L-PCFG uses the concreteness score derived from Brysbaert et al. (2013), normalized (the raw scores are on a 5-point rating scale) to lie between 0 and 1. Given an observed sentence $\boldsymbol{x}$, h_i adds the concreteness score of the root word x_j, specifically the head of the constituent generated from the rule that spans from the beginning to the end of the sentence. Table 1 shows the concreteness scores of the possible lexical heads in Equation 2 on an example sentence.

fans	observe	a	basketball	game	in	progress
0.942	0.526	0.292	0.994	0.9	0.6	0.424

Table 1: The concreteness score for each word in an example phrase. Perceptible entities like *fans* and *game* have higher scores than *in* and *observe*. Note that the concreteness scores are not marginalized by the words in a sentence.

3.3 Vision-based Coupling Heuristic

For the second variant of our model, we introduce the COUPLING heuristic: instead of upweighting the root word with high word concreteness, we incentivize rules that might have generated spans according to the argument-predicate relationship extracted from the visual information. Specifically, this objective aims to reward spans that contain both an argument and predicate in a relation with the head assigned to the predicate. For example, given a caption, *a girl is eating a slice of cheesecake*, and the predicted semantic role labels from the corresponding image are the agent *woman*, the activity *eat*, and the item *cake*. COUPLING sets $\lambda_c = 0$ in Equation 2. We use a VLP (Zhou et al., 2020) fine-tuned on a subset of MSCOCO and a subset of imSitu (Yatskar et al., 2016) to generate semantic role labels for the image corresponding to the text caption (see Figure 3). In Equation 2, we use s_i to denote the span generated by the rule r_i, and $\mathcal{V}$ to denote the set of the rewarded spans. The rewarded spans are selected following the procedure described below.

Label-Caption Alignment For a given image, its semantic role labels consist of an action, its participants (such as actors, objects, substances, lo-

Algorithm 1 Rewarded Span Generation

Inputs:
$caption, labels, aligns$
Initialize:
$\mathcal{V} \leftarrow \varnothing$, $F = zeros(labels)$
for l_1 in $labels$ **do**
  for a in $aligns$ **do**
    Get cap, l_2, $score$ from a
    if $l_1 == l_2$ **then**
      $F_i = index(cap)$
    end if
  end for
end for
compute distance D between F_0 and rest of F
F_0 is the predicate
$A \leftarrow argsort(D)$
for $i = 1, \ldots, N$ **do**
  $d \leftarrow D[A[i]]$
  if $d < 0$ **then**
    $s \leftarrow (F[A[i]], F[0], F[0])$
  else$s \leftarrow (F[0], F[A[i]], F[0])$
  end if
  $\mathcal{V} \leftarrow \mathcal{V} + s$
end for

cations, and etc.), and the roles these participants play in the activity (Yatskar et al., 2016). We perform alignment between the captions of the dataset and the predicted semantic labels to locate the predicted perceived entities from the VLP model in the caption. Since the VLP model is pretrained with an unsupervised learning objective, using the vision-based heuristic on the neural L-PCFG remains an unsupervised task. The semantic labels are preprocessed before the alignment by removing the role labels. The first entity is always the main activity followed by the participants in the order they appear in the semantic labeling predictions. We also removed stop words from the captions to generate more reliable alignment pairs. This is largely because a misalignment, such as aligning determiners with nouns, can potentially be detrimental to use the COUPLING heuristic on the neural L-PCFG. An example of the alignment inputs and the generated outputs with the corresponding alignment scores are described in Figure 3. We use the Dice alignment method (Melamed, 1997) to find co-occurrences of the caption words and semantic labels and generate alignment pairs between them.

Because semantic role labels explain the situation depicted in the image, it is reasonable to couple the words of the corresponding caption that realizes the role labels in a similar way: couple the predicate representative, *eat*, with the representatives of its arguments, *girl* and *cheesecake*.

Rewarded Span Generation For each caption, we use the alignment pairs to generate a set of rewarded spans $\mathcal{V}$ in the third term of Equation 2. Each generated span couples only one argument with the predicate: the start and end of the span are based on the order in which the predicate and the alignment appeared in the caption, and the predicate word is always the head word of the span. Algorithm 1 details the procedure for generating the rewarded spans.

4 Experiments

We run experiments for the three model variants we propose in the previous section: the word concreteness prior, the vision-based heuristic, and the combined method. For word concreteness, our experiments aim to answer the following questions: (1) "does incorporating concreteness in the learning process improve dependency induction?" and (2) "once concreteness has been used in learning, can the model still achieve good performance when concreteness information is no longer available during inference?". In particular, the second question asks whether the model can internalize the concept of concreteness during training and accurately induce dependency structure even when concreteness is not explicitly provided. Additionally, we experiment with the vision-based heuristic model to explore (3) whether structured priors at the phrase-level can improve constituency parsing, and (4) whether it is also effective on improving dependency parsing. Finally, we combine the priors to investigate the utility of concreteness at both the word and the phrase levels for further improving the performance of the neural L-PCFG. This combination allows more effect of the concreteness priors when training the neural L-PCFG to jointly learn the constituency and the dependency structures. We experiment on the combined model that applies both priors either at the root level (as in CONCRETE L-PCFG), or at the rule-level that produces the parsed tree $\boldsymbol{t}$ (as in COUPLING) to fully understand the influence of the priors and the effect of where they are incorporated in the neural L-PCFG grammars.

4.1 Dataset

We use the MSCOCO dataset because it has a large number of visually concrete concepts, which provides a good testbed for our model. Although the MSCOCO dataset is not commonly used in previ-

ous works on dependency induction, MSCOCO has been used in previous work on visually informed constituency induction (Shi et al., 2019; Zhao and Titov, 2020). Using the same splits as VGNSL, the dataset contains 82,783 images for training, 1,000 for validation and testing, respectively. Each image has five corresponding captions. As a preprocessing step, we tokenize and lemmatize the captions (hyphenated words are preserved as a single token) using SpaCy (Honnibal et al., 2020). We use the Berkeley constituent parser (Kitaev and Klein, 2018) to create the gold trees, and we use the Stanford Parser for generating the gold dependencies (de Marneffe and Manning, 2008) for the validation and test sets.

Concreteness scores Each word of the processed sentence in the MSCOCO dataset is matched with its lemma's concreteness rating from Brysbaert et al. (2013). If a word's lemma is absent from the concreteness rating, the word has a concreteness of zero.

4.2 Baseline Methods

We compare our proposed CONCRETE L-PCFG model and the COUPLING heuristic to several existing state-of-the-art methods:

- **DMV**: The dependency model with valence (Klein and Manning, 2004, DMV) is trained on POS tags obtained from an unsupervised tagging model following He et al. (2018) using sentences that have fewer than 20 tokens. This allows the model to run faster while retaining high coverage of the data (97.6% of the training set).

- **HI+FastText+IN**: The results from VGNSL with head-initial bias, FastText embeddings, and normalized image features (Shi et al., 2019).

- **1, $\mathbf{S_{MHI}}$, $\mathbf{C_{MX}}$-IN**: The results from Kojima et al. (2020) that reduces the image dimension to a single-dimension, uses mean and head-initial inductive bias for the score function ($\mathbf{S_{MHI}}$), uses max pooling for the combine function ($\mathbf{C_{MX}}$), and **-IN** removes normalized image features.

- **VC-PCFG**: The results of the visually-grounded compound PCFGs with the language modeling objective (Zhao and Titov, 2020) trained with 30 non-terminals, 60 preterminals, and 512-dimensional hidden states for the LSTM inference network.

- **neural L-PCFG**: The state-of-the-art neural L-PCFG (Zhu et al., 2020), i.e., the base model described in Section 2.

4.3 Our Methods

To compare with previous work, we use pre-trained FastText (Joulin et al., 2016) embeddings for the MSCOCO dataset on all variants of our model described in Section 3. We initialized the preterminals embeddings with centroids obtained using K-means clustering on the pre-trained word embeddings following Zhu et al. (2020). This initialization allows the model to have a preliminary understanding of word meanings before training.

As described in Section 3, we experiment with three variants of our proposed model. The first of these variants is CONCRETE L-PCFG, described in Section 3.2. We experimented with different values of λ_c in Equation 2 between 1.0 and 3.0, to vary the amplitude of the concreteness prior, but found that the model's performance is not strongly affected by this hyperparameter. Because the neural L-PCFG benefits from a larger grammar (Zhu et al., 2020), we increase the number of non-terminals and preterminals to 20 and 25, respectively, when training on the MSCOCO dataset as compared to the original paper. All other hyperparameters remain the same as Zhu et al. (2020).

We conduct experiments with the COUPLING variant of the model, described in Section 3.3. We train the model with 15 non-terminals, 20 preterminals, and decrease the hidden states of the LSTM inference network from 512 dimensions to 128. This is because, in our preliminary experiments, the effect of the vision-based heuristic was more apparent with a smaller inference network. We also run experiments with preterminals of 10 to further limit the strength of the neural LPCFG to uncover the difference, if any, in performance between dependency parsing and constituency parsing. One advantage of decreasing the grammar size is the reduction of parameters for faster training and lower memory requirements.Two values, 1.0 and 2.0, are used for λ_v in Equation 2.

Finally, we combine the two priors from Section 3.2 and Section 3.3 in the COMBINED_ROOT and COMBINED_NON_ROOT models, which perform the combination at the root-level and the rule-level, respectively. The combined models are trained with 15 non-terminals, 20 preterminals, and with the hidden states of 128 dimensions. Hyperparameters λ_c and λ_v are set to 0.25 to avoid overpow-

	PT	NT	Corpus F1	Sent F1	DAS	UAS
Small grammar system						
NEURAL L-PCFG	10	15	53.04	53.21	16.55	47.41
COUPLING	10	15	**54.84**	**55.01**	26.22	**53.23**
Large grammar system						
NEURAL L-PCFG	20	15	56.63	56.46	13.76	42.08
COUPLING	20	15	58.52	58.6	18.27	45.79
COMBINED_ROOT	20	15	51.01	50.91	17.94	46.15
COMBINED_NON_ROOT	20	15	**68.73**	**69.23**	18.22	47.12
NEURAL L-PCFG	25	20	58.02	58.63	15.63	42.33
CONCRETE L-PCFG	25	20	54.81	55.21	**31.42**	**52.23**
Baselines						
DMV	-	-	-	-	18.79	42.47
HI+FastText+IN	-	-	54.4	-	-	-
1, S_{MHI}, C_{MX}-IN	-	-	57.5	-	-	-
VC-PCFG**	60	30	59.3	59.4	-	-

Table 2: Performance of constituency parsing (F1) and dependency parsing (DAS/UAS) of our proposed models as compared to the baselines. The best setup of each model is reported in this table, selected based on validation set performance. The best scores for each task are **highlighted**. PT and NT are the number of preterminals and non-terminals of the neural L-PCFG, respectively. Priors are added at the root level for the COMBINED_ROOT model. For the NON_ROOT model, the priors are added at each rule-level. ** indicates the VC-PCFG (Zhao and Titov, 2020), which is trained with a much larger grammar system.

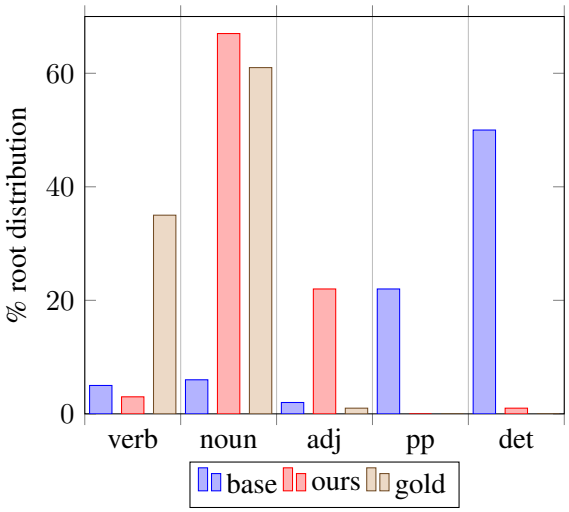


Figure 4: Root distribution (%) by part-of-speech tags of the parse trees predicted with neural L-PCFG (base) and CONCRETE L-PCFG (ours), compared to the gold distribution. POS categories with root distribution$<$ .05for all setups are removed for clarity.

ering the neural L-PCFG with two sets of priors.

We evaluate all models using directed and undirected attachment score (DAS and UAS) to measure dependency parsing, and F1 score for constituency parsing. We select the model checkpoint that maximizes the F1 score of the validation set. We report both corpus-level F1 and sentence-level F1 numbers in our experiments.

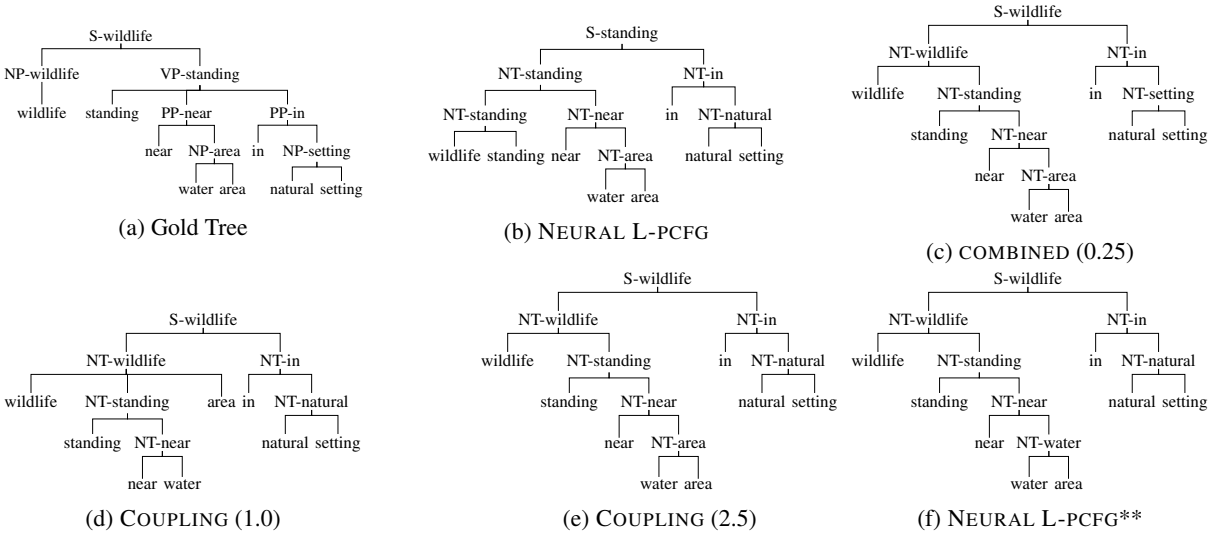
4.4 Results

Table 2 shows our three model variants compared with the baseline models. Both the CONCRETE L-PCFG and the neural L-PCFG show comparable results in F1 with the VGNSL from Shi et al. (2019) and from Kojima et al. (2020). In dependency parsing, CONCRETE L-PCFG doubles the DAS as compared to the neural L-PCFG, while also outperforming DMV by a large margin.

For further analysis of the performance improvement, we look at the part-of-speech (POS) distributions of the sentence roots in MSCOCO (Figure 4). For the text-only neural L-PCFG, the model has a tendency to select prepositions and determiners as heads. This is intuitive because function words are far more common in the data, and there are no constraints against choosing them as the root of the tree. In contrast, CONCRETE L-PCFG chooses nouns as the root word more often than function words, which is closer to the gold distribution, thus indicating the utility of the concreteness prior that upweights perceptible entities (typically nouns).

In Table 2, we observe that with smaller grammar systems, the proposed COUPLING method achieves better dependency induction performance as compared to the neural L-PCFG of the same grammar size. With the large grammar system, the best COUPLING model achieves comparable performance in constituency parsing (F1) with the neural L-PCFG, and all models improve performance in dependency parsing (DAS/UAS) from the baseline neural L-PCFG on MSCOCO. In addition, although the COMBINED_ROOT shows slightly lower performance in dependency parsing compared with the COMBINED_NON_ROOT, the COMBINED_ROOT still improves over the neural L-PCFG. This shows that enforcing the priors at the root level might be too late—the model has likely already selected incorrect heads in lower-level spans.

Finally, we see that the COMBINED method outperforms all models in terms of constituency parsing performance by a large margin. We depict the predicted tree structures for an example sentence with various models in Figure 5. In this example, (b) *standing* is incorrectly selected as the root of the sentence. Interestingly, (c) is almost identical



(a) Gold Tree (b) NEURAL L-PCFG (c) COMBINED (0.25)

(d) COUPLING (1.0) (e) COUPLING (2.5) (f) NEURAL L-PCFG**

Figure 5: The predicted tree structures for each model. ** indicates the tree structure was predicted by both CONCRETE L-PCFG and neural L-PCFG with the same model size. Note that (b) is the baseline model in the vision-based heuristic experiments, and (f) is the baseline model in the concreteness experiments.

to (e) except the correct head is predicted by (c) in the subtree *natural setting*. It is possible that due to the simplicity of the constituency structures in MSCOCO, the neural L-PCFG finds a set of optimal latent rules which explains most of the noun-phrases in the corpus but fails to identify the correct root for dependency parsing. This indicates that learning the correct lexical head is rather difficult for the neural L-PCFG and it requires additional information such as the concreteness priors to improve on dependency parsing.

5 Discussion and Conclusion

Overall, our results show that the substantial increase in dependency induction performance using our method is due to the added word concreteness prior. We demonstrate that concreteness is a good approximation for understanding subcategories of words in a dependency parsing model. However, we recognize that this approximation is insufficient for verb phrases, because an action described by a verb is often coupled with visible entities (Alikhani and Stone, 2019). For example, in a sentence *many people gather around a building with clocks*, our approach incorrectly upweights *people* to be the root rather than *gather*, because the concreteness of the former is higher than the latter.

Similar to Alikhani and Stone (2019), our analysis in Figure 4 shows that MSCOCO includes a large number of noun phrases (almost double the number of verb phrases). This disproportional ratio might contribute to the lower DAS for the DMV and text-only neural L-PCFG models. Because MSCOCO contains shorter sentences with a simpler syntactic structure as compared to the Penn Treebank, neural L-PCFG converges to a local optimum that achieves high constituency accuracy but low dependency accuracy (Table 2). By adding concreteness, our results show that dependency accuracy can be significantly improved while still maintaining a high constituency accuracy.

Our experiments with the proposed vision-based heuristic show that both dependency and constituency accuracy measures can be improved by leveraging the joint learning setting of the neural L-PCFG and the visually derived information. Unlike previous work on visually grounded models, our COUPLING heuristic selects what is visually significant from an image in the language modeling objective. However, certain limitations exist in our simple heuristic, including noise propagation from the semantic role labeling model. Because MSCOCO has a large portion of noun phrases, the action predicted by the semantic role labeling model from the image is often misaligned with a noun term in the caption and vice versa. This could mislead the downstream grammar induction task and hurt the constituency accuracy. For example, the left constituent from the root in (d) of Figure 5 is grammatically incorrect. Given these observations, we removed the rewards to spans that couple the arguments when predicate is not aligned in the COMBINED model.

In future work, we plan to explore more ele-

gant ways to parameterize the priors in the neural L-PCFG, instead of using the hyperparameter λ. Nonetheless, different from other visually grounded models, which jointly learn the textual representations with the paired image,our approach directly influences the language model by encoding the visual information completely in the text domain. One advantage of our approach is that the visual information is directly applied to optimize a language modeling objective, allowing explainability of which parts of the image are helpful for learning the syntax of the corresponding caption.

Our work provides a basis for solving higher-level syntax ambiguities. For example, the sentence *the girl will put the orange on the tray in the bowl* can be decomposed in two ways: [the girl will put][the orange on the tray][in the bowl] and [the girl will put the orange][on the tray in the bowl] (Coco and Keller, 2015). Although both analyses are grammatically correct, only one is correct when a visual reference (i.e., an image) is provided. Future dependency induction models could aim at untangling such ambiguity using more complex visual information.

Finally, although we show that concreteness is useful for dependency induction, using concreteness is seemingly an over-simplification for modeling semantic dependency relationships. In human language, root assignment can involve a priority mechanism: when a verb is present in a sentence, it immediately becomes the root despite how concrete other words are. Our work suggests that further investigation into modeling such priority mechanisms to improve the grammar induction performance even more.

Acknowledgements

This work was supported in part by the DARPA GAILA project (award HR00111990063). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on. We would also like to thank the reviewers for their thoughtful comments.

References

Malihe Alikhani and Matthew Stone. 2019. "caption" as a coherence relation: Evidence and implications. In *Proceedings of the Second Workshop on Shortcomings in Vision and Language*, pages 58–67, Minneapolis, Minnesota. Association for Computational Linguistics.

Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, Nicolas Pinto, and Joseph Turian. 2020. Experience grounds language. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Marc Brysbaert, Amy Warriner, and Victor Kuperman. 2013. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior research methods*, 46.

Jiong Cai, Yong Jiang, and Kewei Tu. 2017. CRF autoencoder for unsupervised dependency parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1638–1643, Copenhagen, Denmark. Association for Computational Linguistics.

Eugene Charniak. 1996. Tree-bank grammars. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2*, AAAI'96, page 1031–1036. 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)*.

Moreno I. Coco and Frank Keller. 2015. The interaction of visual and linguistic saliency during syntactic ambiguity resolution. *Quarterly Journal of Experimental Psychology*, 68(1):46–74. PMID: 25176109.

Michael Collins. 2003. Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29(4):589–637.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The Stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8, Manchester, UK. Coling 2008 Organizing Committee.

Junxian He, Graham Neubig, and Taylor Berg-Kirkpatrick. 2018. Unsupervised learning of syntactic structure with invertible neural projections. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1292–1302.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models.

Daniel Jurafsky and James H. Martin. 2009. *Speech and Language Processing (2Nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Yoon Kim, Chris Dyer, and Alexander Rush. 2019. 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.

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)*, Melbourne, Australia. Association for Computational Linguistics.

Dan Klein and Christopher Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 478–485, Barcelona, Spain.

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.

Bowen Li, Jianpeng Cheng, Yang Liu, and Frank Keller. 2019. Dependency grammar induction with a neural variational transition-based parser. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6658–6665.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312.

I. Dan Melamed. 1997. A word-to-word model of translational equivalence.

Tahira Naseem, Harr Chen, Regina Barzilay, and Mark Johnson. 2010. Using universal linguistic knowledge to guide grammar induction. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1234–1244, Cambridge, MA. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

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*.

Mark Yatskar, Luke Zettlemoyer, and Ali Farhadi. 2016. Situation recognition: Visual semantic role labeling for image understanding. In *Conference on Computer Vision and Pattern Recognition*.

Yanpeng Zhao and Ivan Titov. 2020. Visually grounded compound PCFGs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4369–4379, Online. Association for Computational Linguistics.

Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. 2020. Unified vision-language pre-training for image captioning and vqa. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):13041–13049.

Hao Zhu, Yonatan Bisk, and Graham Neubig. 2020. The return of lexical dependencies: Neural lexicalized PCFGs. *Transactions of the Association for Computational Linguistics*, 8:647–661.