What is Learned in Visually Grounded Neural Syntax Acquisition

Noriyuki Kojima, Hadar Averbuch-Elor,
Alexander Rush and **Yoav Artzi**
Department of Computer Science and Cornell Tech, Cornell University
{nk654,he93,arush}@cornell.edu
{yoav}@cs.cornell.edu

Abstract

Visual features are a promising signal for learning bootstrap textual models. However, black-box learning models make it difficult to isolate the specific contribution of visual components. In this analysis, we consider the case study of the Visually Grounded Neural Syntax Learner (Shi et al., 2019), a recent approach for learning syntax from a visual training signal. By constructing simplified versions of the model, we isolate the core factors that yield the model's strong performance. Contrary to what the model might be capable of learning, we find significantly less expressive versions produce similar predictions and perform just as well, or even better. We also find that a simple lexical signal of noun concreteness plays the main role in the model's predictions as opposed to more complex syntactic reasoning.

1 Introduction

Language analysis within visual contexts has been studied extensively, including for instruction following (e.g., Anderson et al., 2018b; Misra et al., 2017, 2018; Blukis et al., 2018, 2019), visual question answering (e.g., Fukui et al., 2016; Hu et al., 2017; Anderson et al., 2018a), and referring expression resolution (e.g., Mao et al., 2016; Yu et al., 2016; Wang et al., 2016). While significant progress has been made on such tasks, the combination of vision and language makes it particularly difficult to identify what information is extracted from the visual context and how it contributes to the language understanding problem.

Recently, Shi et al. (2019) proposed using alignments between phrases and images as a learning signal for syntax acquisition. This task has been long-studied from a text-only setting, including recently using deep learning based approaches (Shen et al., 2018a, 2019; Kim et al., 2019; Havrylov et al., 2019; Drozdov et al., 2019, inter alia). While the introduction of images provides a rich new signal for the task, it also introduces numerous challenges, such as identifying objects and analyzing scenes.

In this paper, we analyze the Visually Grounded Neural Syntax Learner (VG-NSL) model of Shi et al. (2019). In contrast to the tasks commonly studied in the intersection of vision and language, the existence of an underlying syntactic formalism allows for careful study of the contribution of the visual signal. We identify the key components of the model and design several alternatives to reduce the expressivity of the model, at times, even replacing them with simple non-parameterized rules. This allows us to create several model variants, compare them with the full VG-NSL model, and visualize the information captured by the model parameters.

Broadly, while we would expect a parsing model to distinguish between tokens and phrases along multiple dimensions to represent different syntactic roles, we observe that the model likely does not capture such information. Our experiments show that significantly less expressive models, which are unable to capture such distinctions, learn a similar model of parsing and perform equally and even better than the original VG-NSL model. Our visualizations illustrate that the model is largely focused on acquiring a notion of noun concreteness optimized for the training data, rather than identifying higher-level syntactic roles. Our code and experiment logs are available at https://github.com/lil-lab/vgnsl_analysis.

2 Background: VG-NSL

VG-NSL consists of a greedy bottom-up parser made of three components: a token embedding function (ϕ), a phrase combination function (combine), and a decision scoring function (score). The model is trained using a reward signal computed by matching constituents and images.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 2615–2635
July 5 - 10, 2020.

Algorithm 1 VG-NSL greedy bottom-up parser

Input: A sentence $\bar{x} = \langle x_1, \ldots, x_n \rangle$.
Definitions: $\phi(\cdot)$ is a token embedding function; $\text{combine}(\cdot)$ and $\text{score}(\cdot)$ are learned functions defined in Section 2.

$\mathcal{C}, \mathcal{T} \leftarrow \{[i,i]\}_{i=1}^{n}$
$\mathbf{x}_{[i,i]} \leftarrow \phi(x_i) \quad \forall i = 1, \ldots, n$
while $[1,n] \notin \mathcal{T}$ **do**
  $i, k, j = \underset{[i,k],[k+1,j] \in \mathcal{C}}{\text{argmax}} \ \text{score}(\mathbf{x}_{[i,k]}, \mathbf{x}_{[k+1,j]})$
  $\mathbf{x}_{[i,j]} \leftarrow \text{combine}(\mathbf{x}_{[i,k]}, \mathbf{x}_{[k+1,j]})$
  $\mathcal{T} \leftarrow \mathcal{T} \cup \{[i,j]\}$
  $\mathcal{C} \leftarrow (\mathcal{C} \cup \{[i,j]\}) \setminus \{[i,k],[k+1,j]\}$
return $\mathcal{T}$

Given a sentence $\bar{x}$ with n tokens $\langle x_1, \ldots, x_n \rangle$, the VG-NSL parser (Algorithm 1) greedily constructs a parse tree by building up a set of constituent spans $\mathcal{T}$, which are combined spans from a candidate set $\mathcal{C}$. Parsing starts by initializing the candidate set $\mathcal{C}$ with all single-token spans. At each step, a score is computed for each pair of adjacent candidate spans $[i,k]$ and $[k+1,j]$. The best span $[i,j]$ is added to $\mathcal{T}$ and $\mathcal{C}$, and the two sub-spans are removed from $\mathcal{C}$. The parser continues until the complete span $[1,n]$ is added to $\mathcal{T}$.

Scoring a span $[i,j]$ uses its span embedding $\mathbf{x}_{[i,j]}$. First, a d-dimensional embedding for each single-token span is computed using ϕ. At each step, the score of all potential new spans $[i,j]$ are computed from the candidate embeddings $\mathbf{x}_{[i,k]}$ and $\mathbf{x}_{[k+1,j]}$. The VG-NSL scoring function is:

$$\text{score}(\mathbf{x}_{[i,k]}, \mathbf{x}_{[k+1,j]}) = \text{MLP}_s([\mathbf{x}_{[i,k]}; \mathbf{x}_{[k+1,j]}]) \ ,$$

where MLP_s is a two-layer feed-forward network. Once the best new span is found, its span embedding is computed using a deterministic combine function. VG-NSL computes the d-dimensional embedding of the span $[i,j]$ as the L2-normalized sum of the two combined sub-spans:

$$\text{combine}(\mathbf{x}_{[i,k]}, \mathbf{x}_{[k+1,j]}) = \frac{\mathbf{x}_{[i,k]} + \mathbf{x}_{[k+1,j]}}{\left\| \mathbf{x}_{[i,k]} + \mathbf{x}_{[k+1,j]} \right\|_2} \ .$$

Learning the token embedding function ϕ and scoring model MLP_s relies on a visual signal from aligned images via a reward signal derived from matching constituents and the image. The process alternates between updating the parser parameters and an external visual matching function, which is estimated by optimizing a hinge-based triplet ranking loss similar to the image-caption retrieval loss of Kiros et al. (2014). The parser parameters are estimated using a policy gradient method based on the learned visual matching function, which encourages constituents that match with the corresponding image. This visual signal is the only objective used to learn the parser parameters. After training, the images are no longer used and the parser is text-only.

3 Model Variations

We consider varying the parameterization of VG-NSL, i.e., ϕ, combine, and score, while keeping the same inference algorithm and learning procedure. Our goal is to constrain model expressivity, while studying its performance and outputs.

Embedding Bottleneck We limit the information capacity of the parsing model by drastically reducing its dimensionality from $d = 512$ to 1 or 2. We reduce dimensionality by wrapping the token embedding function with a bottleneck layer $\phi_B(x) = \text{MLP}_B(\phi(x))$, where MLP_B is a two-layer feed-forward network mapping to the reduced size. This bottleneck limits the expressiveness of phrase embeddings throughout the parsing algorithm. During training, we compute both original and reduced embeddings. The original embeddings are used to compute the visual matching reward signal, whereas the reduced embeddings are used by score to determine parsing decisions. At test time, only the reduced embeddings are used. In the case of $d = 1$, the model is reduced to using a single criteria. The low dimensional embeddings are also easy to visualize, and to characterize the type of information learned.

Simplified Scoring We experiment with simplified versions of the score function. Together with the lower-dimensional representation, this enables controlling and analyzing the type of decisions the parser is capable of. As we control the information the embeddings can capture, simplifying the scoring function makes sure it does not introduce additional expressivity. The first variation uses a weighted sum with parameters $\mathbf{u}, \mathbf{v}$:

$$\text{score}_{\text{WS}}(\mathbf{x}_{[i,k]}, \mathbf{x}_{[k+1,j]}) = \mathbf{u} \cdot \mathbf{x}_{[i,k]} + \mathbf{v} \cdot \mathbf{x}_{[k+1,j]} \ .$$

This formulation allows the model to learn structural biases, such as the head-initial (HI) bias common in English (Baker, 1987). The second is a non-parameterized mean, applicable for $d = 1$ only:

$$\text{score}_{\text{M}}(\mathbf{x}_{[i,k]}, \mathbf{x}_{[k+1,j]}) = \frac{\mathbf{x}_{[i,k]} + \tau \mathbf{x}_{[k+1,j]}}{1 + \tau} \ ,$$

where τ is a hyper-parameter that enables upweighting the right constituent to induce a HI inductive

bias. We experiment with unbiased $\tau = 1$ (score_M) and HI-biased $\tau = 20$ (score_MHI) scoring.

Reduced Dimension Combine In lower dimensions, the combine function no longer produces useful outputs, i.e., in $d = 1$ it always gives 1 or -1. We therefore consider mean or max pooling:

$$\text{combine}_\text{ME}(\mathbf{x}_{[i,k]}, \mathbf{x}_{[k+1,j]}) = \frac{\mathbf{x}_{[i,k]} + \mathbf{x}_{[k+1,j]}}{2}$$

$$\text{combine}_\text{MX}(\mathbf{x}_{[i,k]}, \mathbf{x}_{[k+1,j]}) = \max(\mathbf{x}_{[i,k]}, \mathbf{x}_{[k+1,j]}) \ .$$

The mean variant computes the representation of a new span as an equal mixture of the two sub-spans, while the max directly copies to the new span representation information only from one of the spans. The max function is similar to how head rules lexicalize parsers (Collins, 1996).

4 Experimental Setup

We train VG-NSL and our model variants using the setup of Shi et al. (2019), including three training extensions: (a) **+HI**: adding a head-initial inductive bias to the training objective; (b) **+FastText**: the textual representations are partially initialized with pre-trained FastText (Joulin et al., 2016); and (c) **-IN**: [1] disabling the normalization of image features. We follow the Shi et al. (2019) setup. We train all VG-NSL variants on 82,783 images and 413,915 captions from the MSCOCO (Lin et al., 2014) training set. We evaluate unsupervised constituency parsing performance using 5,000 non-overlapping held-out test captions. We use additional 5,000 non-overlapping validation captions for model selection, as well as for our analysis and visualization in Section 5. We generate binary gold-trees using Benepar (Kitaev and Klein, 2018), an off-the-shelf supervised constituency parser.

We notate model variations as $d, \text{score}, \text{combine}$. For example, $1, \text{s}_\text{WS}, \text{c}_\text{ME}$ refers to dimensionality $d = 1$, weighted sum scoring function (s_WS), and mean pooling combine (c_ME). We train five models for each variation, and select the best checkpoint for each model by maximizing the parse prediction agreement on the validation captions between five models. The agreement is measured by the self-F_1 agreement score (Williams et al., 2018). This procedure is directly adopted from Shi et al. (2019). We use the hyper-parameters from the original implementation without further tuning.

[1] The authors of Shi et al. (2019) suggested this ablation as particularly impactful on the learning outcome.

Model	NP	VP	PP	ADJP	Avg. F_1
Shi2019	79.6	26.2	42.0	22.0	50.4 ± 0.3
Shi2019*	80.5	26.9	45.0	21.3	51.4 ± 1.1
$1, \text{s}_\text{WS}, \text{c}_\text{ME}$	77.2	17.0	53.4	18.2	49.7 ± 5.9
$2, \text{s}_\text{WS}, \text{c}_\text{ME}$	**80.8**	19.1	52.3	17.1	51.6 ± 0.6
+HI					
Shi2019	74.6	32.5	66.5	21.7	53.3 ± 0.2
Shi2019*	73.1	33.9	64.5	22.5	51.8 ± 0.3
$1, \text{s}_\text{WS}, \text{c}_\text{ME}$	74.0	35.2	62.0	24.2	51.8 ± 0.4
$2, \text{s}_\text{WS}, \text{c}_\text{ME}$	73.8	30.2	63.7	21.9	51.3 ± 0.1
+HI+FastText					
Shi2019	78.8	24.4	65.6	22.0	54.4 ± 0.3
Shi2019*	77.3	23.9	64.3	21.9	53.3 ± 0.1
$1, \text{s}_\text{WS}, \text{c}_\text{ME}$	76.6	21.9	68.7	20.6	53.5 ± 1.4
$2, \text{s}_\text{WS}, \text{c}_\text{ME}$	77.5	22.8	66.3	19.3	53.6 ± 0.2
+HI+FastText-IN					
Shi2019*	78.3	26.6	67.5	22.1	54.9 ± 0.1
$1, \text{s}_\text{M}, \text{c}_\text{MX}$	79.6	29.0	38.3	23.5	49.7 ± 0.2
$1, \text{s}_\text{MHI}, \text{c}_\text{MX}$	77.6	**45.0**	72.3	**24.3**	$\mathbf{57.5 \pm 0.1}$
$1, \text{s}_\text{M}, \text{c}_\text{ME}$	80.0	26.9	62.2	23.2	54.3 ± 0.2
$1, \text{s}_\text{MHI}, \text{c}_\text{ME}$	76.5	20.5	63.6	22.7	52.2 ± 0.3
$1, \text{s}_\text{WS}, \text{c}_\text{ME}$	77.7	26.3	**72.5**	22.0	55.5 ± 0.1
$2, \text{s}_\text{WS}, \text{c}_\text{ME}$	78.5	26.3	69.5	21.1	55.2 ± 0.1

Table 1: Test results. We report the results from Shi et al. (2019) as Shi2019 and our reproduction (Shi2019*). We report mean F_1 and standard deviation for each system and recall for four phrasal categories. Our variants are specified using a representation embedding ($d \in \{1, 2\}$), a score function (s_M: mean, s_MHI: mean+HI, s_WS: weighted sum), and a combine function (c_MX: max, c_ME: mean).

We evaluate using gold trees by reporting F_1 scores on the ground-truth constituents and recall on several constituent categories. We report mean and standard deviation across the five models.

5 Experiments

Quantitative Evaluation Table 1 shows our main results. As the table illustrates, The model variations achieve F_1 scores competitive to the scores reported by Shi et al. (2019) across training setups. They achieve comparable recall on different constituent categories, and robustness to parameter initialization, quantified by self-F_1, which we report in an expanded version of this table in Appendix A. The model variations closest to the original model, $1, \text{s}_\text{WS}, \text{c}_\text{ME}$ and $2, \text{s}_\text{WS}, \text{c}_\text{ME}$, yield similar performance to the original model across different evaluation categories and metrics, especially in the +HI and +HI+FastText settings. Most remarkably, our simplest variants, which use $1d$ embeddings and a non-parameterized scoring function, are still competitive ($1, \text{s}_\text{M}, \text{c}_\text{ME}$) or even outperform ($1, \text{s}_\text{MHI}, \text{c}_\text{MX}$) the original VG-NSL.

Our simplified model variations largely learn the

Training Setting	$1, s_{WS}, c_{ME}$	$2, s_{WS}, c_{ME}$	U
Basic Setting	72.0	77.5	87.5
+HI	78.2	80.3	91.8
+HI+FastText	80.5	83.1	92.3
+HI+FastText-IN	85.6	86.4	92.8

Table 2: Self-F_1 agreement between two of our variations and the original VG-NSL model. We also report the upper bound scores (U) calculated by directly comparing two separately trained sets of five original VG-NSL models.

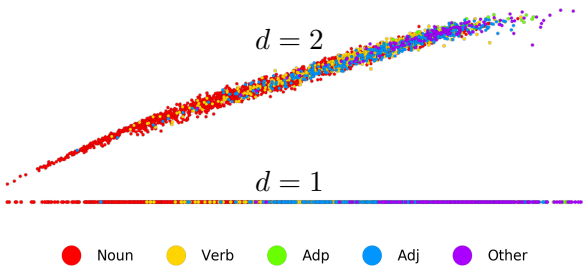


Figure 1: Token embedding visualization for $2, s_{WS}, c_{ME}$ (top) and $1, s_{WS}, c_{ME}$ (bottom) colored by universal POS tags (Petrov et al., 2012). Appendix A includes an expanded version of this figure.

same parsing model as the original. Table 2 shows self-F_1 agreement by comparing constituents predicted by our models in each training setting with the original model. We compute this agreement measure by training two sets of five models on the training data, and selecting checkpoints using the validation captions for each of our model variants and the original VG-NSL model. We parse the same validation captions using each model and generate ten parse trees for each caption, one for each model (i.e., five for each distinct set). We calculate self-F_1 agreement between models by comparing parse trees from model variants to parse trees from the original VG-NSL. We permute all 25 (five by five) combinations of variant/VG-NSL pairs and obtain self-F_1 agreement between the model variant and the original VG-NSL by averaging scores from each pair. For the upper-bound agreement calculation, we train two distinct sets of five original VG-NSL models. Our parsing model is very similar but not exactly identical: there is roughly a six points F1 agreement gap in the best case compared to the upper bound. We consider these numbers a worst-case scenario because self-F_1 agreement measures on the validation data are used twice. First, for model selection to eliminate the variance of each five-model set, and second for the variant agreement analysis.

Expressivity Analysis We analyze the embeddings of the two variants closest to the original model, $1, s_{WS}, c_{ME}$ and $2, s_{WS}, c_{ME}$, to identify the information they capture. Both behave similarly to the original VG-NSL. Figure 1 visualizes the token embedding space for these variants. Interestingly, the distribution of the $2d$ token embeddings seems almost linear, suggesting that the additional dimension is largely not utilized during learning, and that both have a strong preference for separating nouns from tokens belonging to other parts of speech. It seems only one core visual signal is used in the model and if this factor is captured, even a $1d$ model can propagate it through the tree.

We hypothesize that the core visual aspect learned, which is captured even in the $1d$ setting, is noun concreteness. Table 3 shows that the reduced token embeddings have strong correlations with existing estimates of concreteness. Figure 2 shows the ordering of example nouns according to our $1d$ learned model representation. We observe that the concreteness estimated by our model correlates with nouns that are relatively easier to ground visually in MSCOCO images. For example, nouns like "giraffe" and "elephant" are considered most concrete. These nouns are relatively frequent in MSCOCO (e.g., "elephant" appears 4,633 times in the training captions) and also have a low variance in their appearances. On the other hand, nouns with high variance in images (e.g., "traveller") or abstract nouns (e.g., "chart", "spot") are estimated to have low concreteness. Appendix A includes examples of concreteness.

We quantify the role of concreteness-based noun identification in VG-NSL by modifying test-time captions to replace all nouns with the most concrete token (i.e., "elephant"), measured according

Model	$1, s_{WS}, c_{ME}$
Turney et al. (2011)	0.73
Brysbaert et al. (2014)	0.75
Hessel et al. (2018)	0.89
Shi2019*	0.94

Table 3: Pearson correlation coefficient of concreteness estimates between our $1, s_{WS}, c_{ME}$ variant and existing concreteness estimates, including reproduced estimates derived from VG-NSL by Shi et al. (2019).

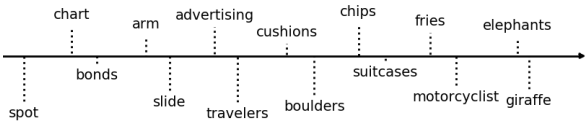


Figure 2: Noun distribution using the $1d$ representation from the $1, s_{WS}, c_{ME}$ variant. The nouns are sorted by their representation value in increasing order from left.

Training Setting	Token	$1, s_{WS}, c_{ME}$	Shi2019*
Basic Setting	herd	$49.5 \Rightarrow 36.3$	$51.0 \Rightarrow 47.6$
*Basic Setting**	cat	$52.4 \Rightarrow 56.9$	$51.0 \Rightarrow 57.2$
+HI	elephant	$51.7 \Rightarrow 63.7$	$51.6 \Rightarrow 59.8$
+HI+FastText	motorcycle	$52.9 \Rightarrow 59.9$	$52.9 \Rightarrow 60.7$
+HI+FastText-IN	elephant	$55.0 \Rightarrow 62.9$	$54.6 \Rightarrow 60.2$

Table 4: F_1 scores evaluated before and after replacing nouns in captions with the most concrete token predicted by models using the $1, s_{WS}, c_{ME}$ configuration. The replacement occurs during test time only as described in Section 5. In *Basic Setting**, we remove one model from $1, s_{WS}, c_{ME}$ which has a significantly low F_1 agreement (54.2) to the rest of four models using the $1, s_{WS}, c_{ME}$ configuration.

to the $1d$ token embeddings learned by our model. We pick the most concrete noun for each training configuration using mean ranking across token embeddings of the five models in each configuration. For example, instead of parsing the original caption "girl holding a picture," we parse "elephant holding an elephant." This uses part-of-speech information to resolve the issue where nouns with low concreteness are treated in the same manner as other part-of-speech tokens. We compare the output tree to the original gold ones for evaluation. We observe that the F_1 score, averaged across the five models, significantly improves from 55.0 to 62.9 for $1, s_{WS}, c_{ME}$ and from 54.6 to 60.2 for the original VG-NSL before and after our caption modification. The performance increase shows that noun identification via concreteness provides an effective parsing strategy, and further corroborates our hypothesis about what phenomena underlie the strong Shi et al. (2019) result. Table 4 includes the results for the other training settings.

6 Conclusion and Related Work

We studied the VG-NSL model by introducing several significantly less expressive variants, analyzing their outputs, and showing they maintain, and even improve performance. Our analysis shows that the visual signal leads VG-NSL to rely mostly on estimates of noun concreteness, in contrast to more complex syntactic reasoning. While our model variants are very similar to the original VG-NSL, they are not completely identical, as reflected by the self-F_1 scores in Table 2. Studying this type of difference between expressive models and their less expressive, restricted variants remains an important direction for future work. For example, this can be achieved by distilling the original model to the less expressive variants, and observing both the agreement between the models and their performance. In our case, this requires further development of distillation methods for the type of reinforcement learning setup VG-NSL uses, an effort that is beyond the scope of this paper.

Our work is related to the recent inference procedure analysis of Dyer et al. (2019). While they study what biases a specific inference algorithm introduces to the unsupervised parsing problem, we focus on the representation induced in a grounded version of the task. Our empirical analysis is related to Htut et al. (2018), who methodologically, and successfully replicate the results of Shen et al. (2018a) to study their performance. The issues we study generalize beyond the parsing task. The question of what is captured by vision and language models has been studied before, including for visual question answering (Agrawal et al., 2016, 2017; Goyal et al., 2017), referring expression resolution (Cirik et al., 2018), and visual navigation (Jain et al., 2019). We ask this question in the setting of syntactic parsing, which allows to ground the analysis in the underlying formalism. Our conclusions are similar: multi-modal models often rely on simple signals, and do not exhibit the complex reasoning we would like them to acquire.

Acknowledgements

Special thanks to Freda Shi for code release and prompt help in re-producing the experiments of Shi et al. (2019). This work was supported by the NSF (CRII-1656998, IIS-1901030), a Google Focused Award, and the generosity of Eric and Wendy Schmidt by recommendation of the Schmidt Futures program. We thank Jack Hessel, Forrest Davis, and the anonymous reviewers for their helpful feedback.

References

Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. 2016. Analyzing the behavior of visual question answering models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1955–1960.

Aishwarya Agrawal, Aniruddha Kembhavi, Dhruv Batra, and Devi Parikh. 2017. C-VQA: A compositional split of the visual question answering (VQA) v1.0 dataset. *CoRR*, abs/1704.08243.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018a. Bottom-up and top-down attention

for image captioning and visual question answering. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 6077–6086.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. 2018b. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 3674–3683.

Mark C. Baker. 1987. *The atoms of language: The mind's hidden rules of grammar*. Basic books.

Valts Blukis, Nataly Brukhim, Andrew Bennett, Ross A. Knepper, and Yoav Artzi. 2018. Following high-level navigation instructions on a simulated quadcopter with imitation learning. In *Proceedings of the Robotics: Science and Systems Conference*.

Valts Blukis, Eyvind Niklasson, Ross A. Knepper, and Yoav Artzi. 2019. Learning to map natural language instructions to physical quadcopter control using simulated flight. In *Proceedings of the Conference on Robot Learning*.

Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior research methods*, 46(3):904–911.

Volkan Cirik, Louis-Philippe Morency, and Taylor Berg-Kirkpatrick. 2018. Visual referring expression recognition: What do systems actually learn? In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 781–787.

Michael John Collins. 1996. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 184–191.

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 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1129–1141.

Chris Dyer, Gábor Melis, and Phil Blunsom. 2019. A critical analysis of biased parsers in unsupervised parsing. *arXiv preprint arXiv:1909.09428*.

Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. 2016. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 457–468.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 6325–6334.

Serhii Havrylov, Germán Kruszewski, and Armand Joulin. 2019. Cooperative learning of disjoint syntax and semantics. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1118–1128.

Jack Hessel, David Mimno, and Lillian Lee. 2018. Quantifying the visual concreteness of words and topics in multimodal datasets. In *Proceedings of North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2194–2205.

Phu Mon Htut, Kyunghyun Cho, and Samuel Bowman. 2018. Grammar induction with neural language models: An unusual replication. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 4998–5003.

Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. 2017. Learning to reason: End-to-end module networks for visual question answering. In *The IEEE International Conference on Computer Vision*, pages 804–813.

Vihan Jain, Gabriel Magalhaes, Alexander Ku, Ashish Vaswani, Eugene Ie, and Jason Baldridge. 2019. Stay on the path: Instruction fidelity in vision-and-language navigation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1862–1872.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Yoon Kim, Alexander M Rush, Lei Yu, Adhiguna Kuncoro, Chris Dyer, and Gábor Melis. 2019. Unsupervised recurrent neural network grammars. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1105–1117.

Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. 2014. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 2676–2686.

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 *European conference on computer vision*, pages 740–755.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L. Yuille, and Kevin Murphy. 2016. Generation and comprehension of unambiguous object descriptions. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 11–20.

Dipendra Misra, Andrew Bennett, Valts Blukis, Eyvind Niklasson, Max Shatkhin, and Yoav Artzi. 2018. Mapping instructions to actions in 3D environments with visual goal prediction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2667–2678.

Dipendra Misra, John Langford, and Yoav Artzi. 2017. Mapping instructions and visual observations to actions with reinforcement learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1004–1015.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation*, pages 2089–2096.

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

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

Haoyue Shi, Jiayuan Mao, Kevin Gimpel, and Karen Livescu. 2019. Visually grounded neural syntax acquisition. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1842–1861.

Peter D Turney, Yair Neuman, Dan Assaf, and Yohai Cohen. 2011. Literal and metaphorical sense identification through concrete and abstract context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 680–690.

Mingzhe Wang, Mahmoud Azab, Noriyuki Kojima, Rada Mihalcea, and Jia Deng. 2016. Structured matching for phrase localization. In *The European Conference on Computer Vision*, pages 696–711.

Adina Williams, Andrew Drozdov*, and Samuel R Bowman. 2018. Do latent tree learning models identify meaningful structure in sentences? In *Transactions of the Association for Computational Linguistics*, volume 6, pages 253–267. MIT Press.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. 2016. Modeling context in referring expressions. In *The European Conference on Computer Vision*, pages 69–85.

A Additional Results and Visualizations

Table 5 is an extended version of Table 1 from Section 5. We include standard deviation for the phrasal category recall and self-F_1 scores evaluated across different parameter initializations. Figure 3 is a larger version of Figure 1 from Section 5. It visualizes the token embeddings of $1, \mathrm{s_{WS}}, \mathrm{c_{ME}}$ and $2, \mathrm{s_{WS}}, \mathrm{c_{ME}}$ for all universal parts-of-speech categories (Petrov et al., 2012). Figures 4 and 5 show several examples visualizing our learned representations with the $1, \mathrm{s_{WS}}, \mathrm{c_{ME}}$ variant, the $1d$ variant closest to the original model, as a concreteness estimate. Figure 4 shows the most concrete nouns, and Figure 5 shows the least concrete nouns. We selected nouns from the top (bottom) 5% of the data as most (least) concrete. We randomly selected image-caption pairs for these nouns.

At the end of the supplementary material, we include tree visualizations, comparing gold trees with phrasal categories, trees generated by the original VG-NSL, and trees generated by our best performing, simplified $1, \mathrm{s_{MHI}}, \mathrm{c_{MX}}$ variant. We select the trees to highlight the difference between VG-NSL and our variant. First, we select all development trees where all five VG-NSL models agree to avoid results that are likely due to initialization differences. We do the same for our variant. Finally, we select all trees where the two sets, from VG-NSL and our variant, disagree. This process leaves us with 814 development examples, out of the original 5,000 examples. We display ten examples from this final set.

Model	NP	VP	PP	ADJP	Avg. F_1	Self-F_1
Shi2019	79.6 ± 0.4	26.2 ± 0.4	42.0 ± 0.6	22.0 ± 0.4	50.4 ± 0.3	87.1
Shi2019*	80.5 ± 1.5	26.9 ± 0.9	45.0 ± 2.9	21.3 ± 1.2	51.4 ± 1.1	87.3
$1, s_{WS}, c_{ME}$	77.2 ± 5.3	17.0 ± 5.2	53.4 ± 12.8	18.2 ± 1.0	49.7 ± 5.9	76.0
$2, s_{WS}, c_{ME}$	$\mathbf{80.8 \pm 1.1}$	19.1 ± 1.1	52.3 ± 3.5	17.1 ± 1.0	51.6 ± 0.6	88.1
+HI						
Shi2019	74.6 ± 0.5	32.5 ± 1.5	66.5 ± 1.2	21.7 ± 1.1	53.3 ± 0.2	90.2
Shi2019*	73.1 ± 0.3	33.9 ± 0.8	64.5 ± 0.2	22.5 ± 0.4	51.8 ± 0.3	91.6
$1, s_{WS}, c_{ME}$	74.0 ± 0.4	35.2 ± 2.0	62.0 ± 1.1	24.2 ± 0.9	51.8 ± 0.4	87.3
$2, s_{WS}, c_{ME}$	73.8 ± 0.3	30.2 ± 0.4	63.7 ± 0.3	21.9 ± 0.3	51.3 ± 0.1	93.3
+HI+FastText						
Shi2019	78.8 ± 0.5	24.4 ± 0.9	65.6 ± 0.1	22.0 ± 0.7	54.4 ± 0.3	89.8
Shi2019*	77.3 ± 0.1	23.9 ± 0.5	64.3 ± 0.3	21.9 ± 0.3	53.3 ± 0.1	92.2
$1, s_{WS}, c_{ME}$	76.6 ± 0.3	21.9 ± 2.3	68.7 ± 4.1	20.6 ± 0.9	53.5 ± 1.4	87.8
$2, s_{WS}, c_{ME}$	77.5 ± 0.2	22.8 ± 0.4	66.3 ± 0.6	19.3 ± 0.7	53.6 ± 0.2	93.6
+HI+FastText-IN						
Shi2019*	78.3 ± 0.2	26.6 ± 0.3	67.5 ± 0.5	22.1 ± 1.0	54.9 ± 0.1	92.6
$1, s_{M}, c_{MX}$	79.6 ± 0.2	29.0 ± 0.7	38.3 ± 0.3	23.5 ± 0.6	49.7 ± 0.2	95.5
$1, s_{MHI}, c_{MX}$	77.6 ± 0.2	$\mathbf{45.0 \pm 0.8}$	72.3 ± 0.2	$\mathbf{24.3 \pm 1.0}$	$\mathbf{57.5 \pm 0.1}$	93.4
$1, s_{M}, c_{ME}$	80.0 ± 0.2	26.9 ± 0.2	62.2 ± 0.4	23.2 ± 0.4	54.3 ± 0.2	95.7
$1, s_{MHI}, c_{ME}$	76.5 ± 0.1	20.5 ± 0.8	63.6 ± 0.6	22.7 ± 0.7	52.2 ± 0.3	94.7
$1, s_{WS}, c_{ME}$	77.7 ± 0.1	26.3 ± 0.4	$\mathbf{72.5} \pm 0.2$	22.0 ± 0.6	55.5 ± 0.1	95.5
$2, s_{WS}, c_{ME}$	78.5 ± 0.4	26.3 ± 0.6	69.5 ± 1.2	21.1 ± 0.5	55.2 ± 0.1	93.7

Table 5: Test results. We report the results from Shi et al. (2019) as Shi2019 and our reproduction as Shi2019*. We report mean F_1 and standard deviation for each system and mean recall and standard deviation for four phrasal categories. Our variants are specified using a representation embedding ($d \in \{1, 2\}$), a score function (s_M: mean, s_{MHI}: mean+HI, s_{WS}: weighted sum), and a combine function (c_{MX}: max, c_{ME}: mean).

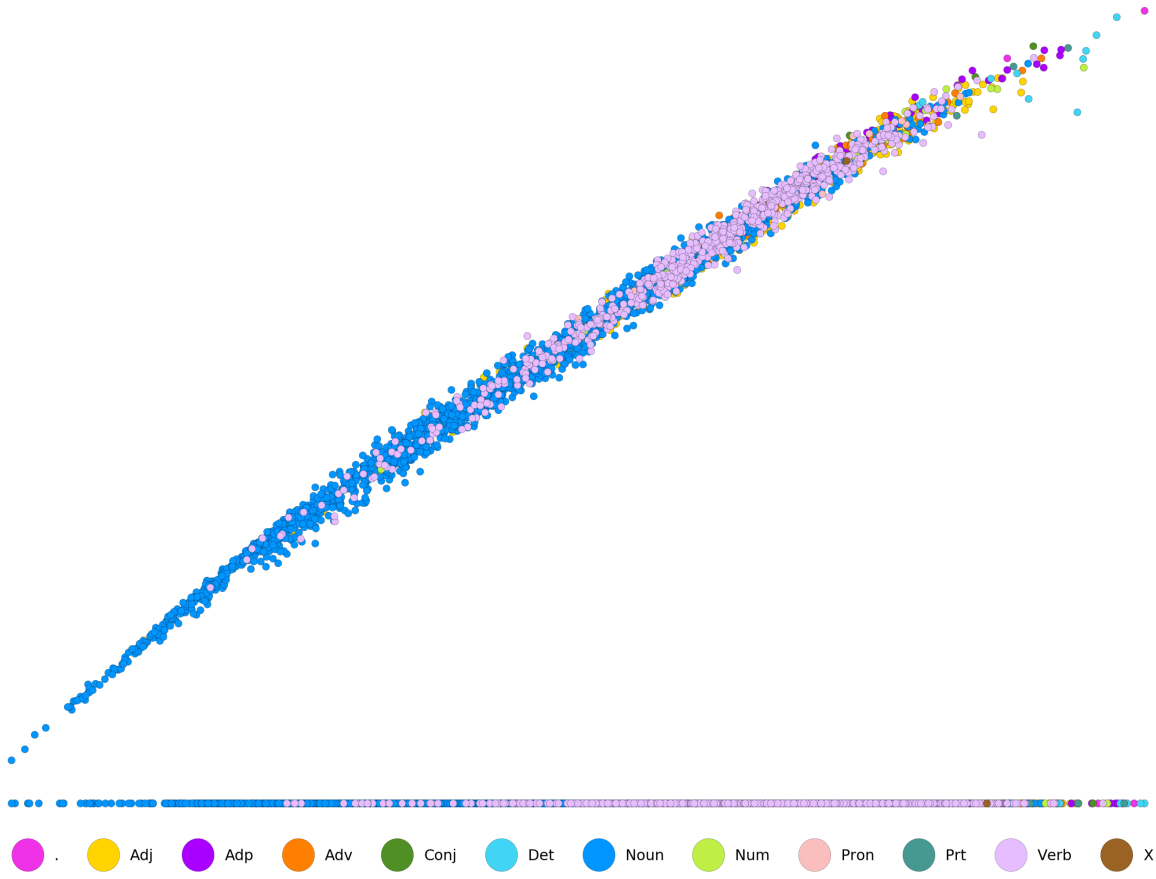
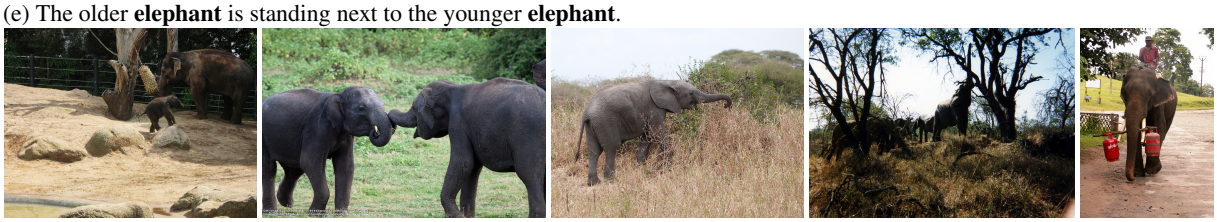


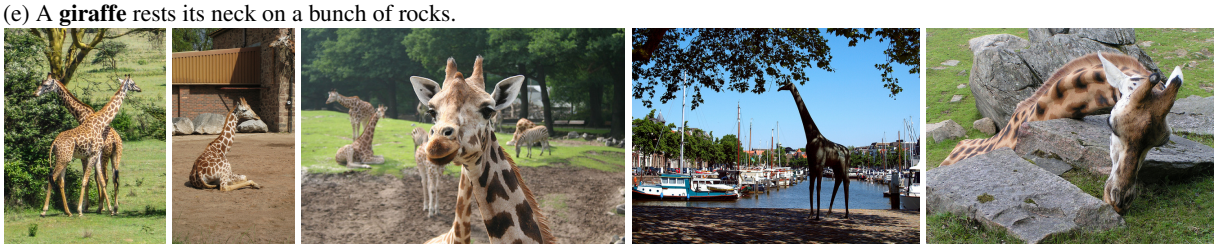
Figure 3: Token embedding visualization for $2, s_{WS}, c_{ME}$ (top) and $1, s_{WS}, c_{ME}$ (bottom) colored by universal POS tags (Petrov et al., 2012).

Elephant (4633 occurrences):
(a) A person riding an **elephant** and carrying gas cylinders.
(b) An **elephant** is in some brown grass and some trees.
(c) A captive **elephant** stands amid the branches of a tree in his park-like enclosure.
(d) Two baby gray **elephant** standing in front of each other.
(e) The older **elephant** is standing next to the younger **elephant**.



(a) (b) (c) (d) (e)

Giraffe (5546 occurrences):
(a) Two **giraffe** standing next to each other on a grassy field.
(b) A **giraffe** laying down on the dirt ground.
(c) A herd of **giraffe** standing next to each other on a field.
(d) A **giraffe** stands beneath a tree beside a marina.
(e) A **giraffe** rests its neck on a bunch of rocks.



(a) (b) (c) (d) (e)

Pizza (8340 occurrences):
(a) A woman holding a **pizza** up in the air.
(b) A slice of **pizza** sitting on top of a white plate.
(c) A **pizza** sitting on top of a plate covered in cheese and tomatoes.
(d) Three pieces of sliced **pizza** on a wooden surface.
(e) Some boxes of frozen **pizzas** are in the store.
(f) A **pizza** topped with cheese and pepperoni with veggies.
(g) A large **pizza** is in a cardboard box.

(a) (b) (c) (d) (e) (f) (g)

Snowboarder (922 occurrences):
(a) A **snowboarder** practicing his moves at a snow facility.
(b) A **snowboarder** is coming down a hill and some trees.
(c) A **snowboarder** rests in the snow on the snowboard.
(d) A **snowboarder** jumps off of a hill instead of just sliding down it.
(e) A **snowboarder** is jumping in the air with their board held to the side.
(f) The snowboard is almost as big as the **snowboarder**.

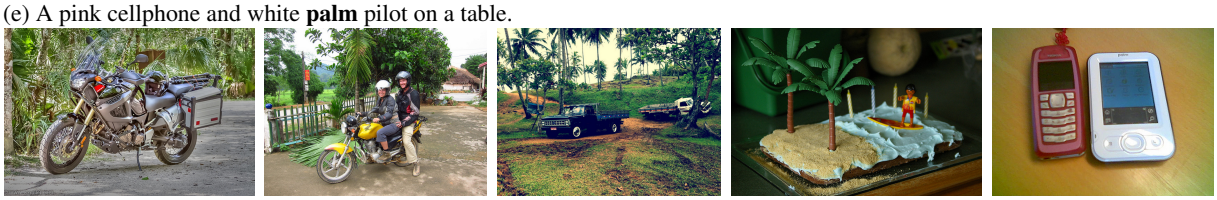
(a) (b) (c) (d) (e) (f)

Figure 4: Image-caption pairs corresponding to noun tokens estimated as most concrete (bottom 5%) in our $1, s_{WS}, c_{ME}$ variant. We also report the number of occurrences in the MSCOCO training set.

Metal (1630 occurrences):
(a) A pink piece of **metal** with a bolt and nut on top.
(b) Wilting roses and greenery in a **metal** vase.
(c) A couple of street signs sitting on top of a **metal** pole.
(d) Kitchen with wooden cabinets and a **metal** sink.
(e) A **metal** toilet and some tissue in a bathroom.

(a) (b) (c) (d) (e)

Palm (321 occurrences):
(a) A motorcycle sits parked in **palm** tree lined driveway.
(b) Two people in helmets on a parked motorcycle and a small **palm** tree to the side of them.
(c) Two flat bed work trucks among **palm** trees .
(d) A cake with **palm** trees, and a person on a surf board.
(e) A pink cellphone and white **palm** pilot on a table.



(a) (b) (c) (d) (e)

Picture (5932 occurrences):
(a) A blurry **picture** of a cat standing on a toilet.
(b) **Picture** of a church and its tall steeple.
(c) The street sign at the intersection of Broadway and 7th avenue is the star of this **picture**.
(d) A **picture** of some people playing with a frisbee.
(e) A little girl sitting in the middle of a restaurant and smiling for **picture**.

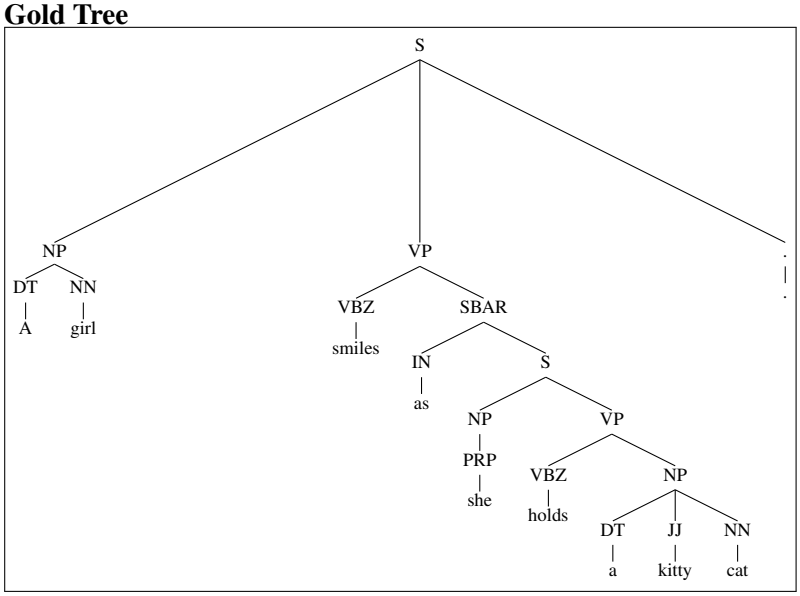
(a) (b) (c) (d) (e)

Time (1184 occurrences):
(a) A **time** lapse photo of a skier skiing down a hill.
(b) A skaterboarder getting major air over some stairs during a night **time** shoot.
(c) The man is trying to eat three hot dogs are the same **time**.
(d) A boy playing a WII game at Christmas **time**.
(e) A large display of a hand holding a cell phone to tell the **time**.

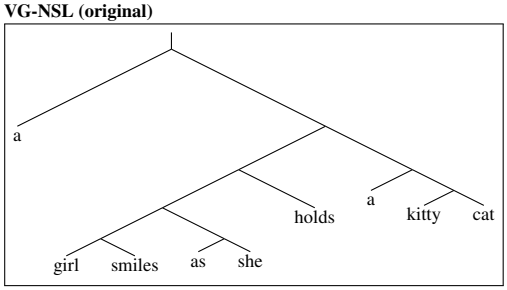
(a) (b) (c) (d) (e)

Figure 5: Image-caption pairs corresponding to noun tokens estimated as least concrete (bottom 5%) in our $1, \mathrm{s_{WS}}, \mathrm{c_{ME}}$ variant. We also report the number of occurrences in the MSCOCO training set.

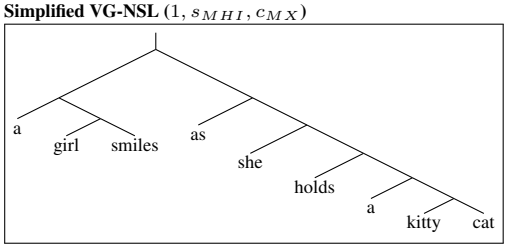
Gold Tree



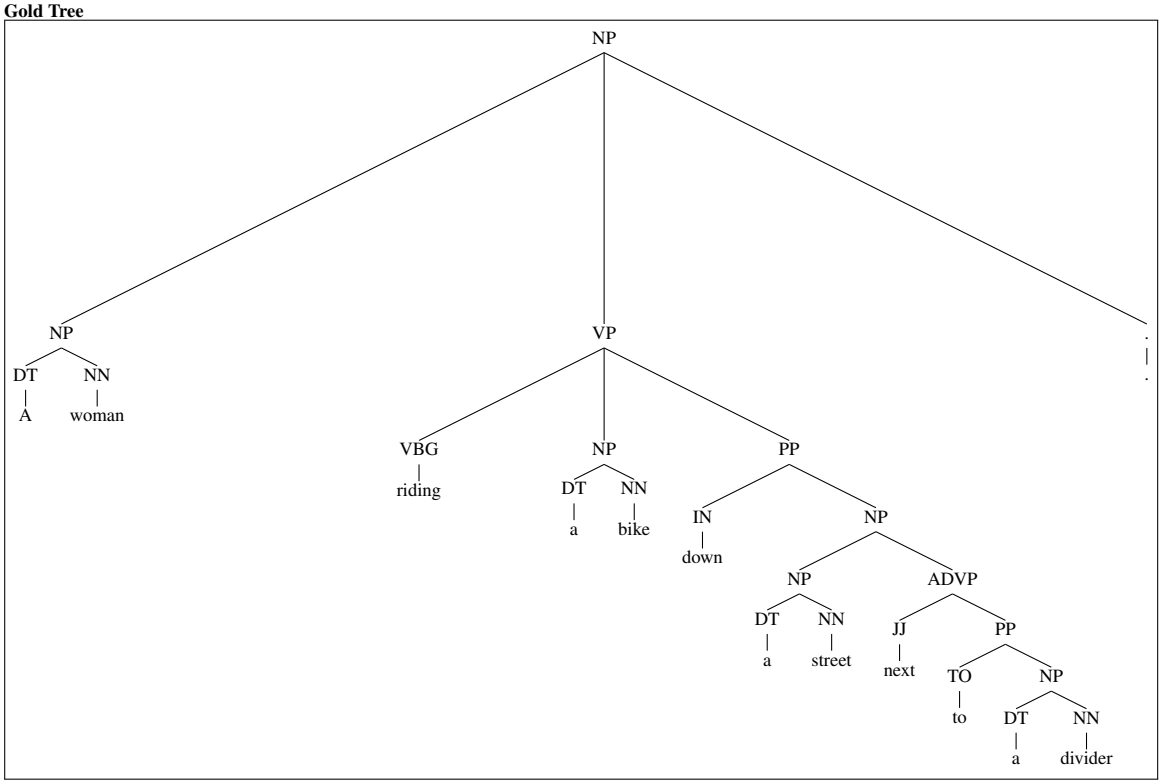
VG-NSL (original)



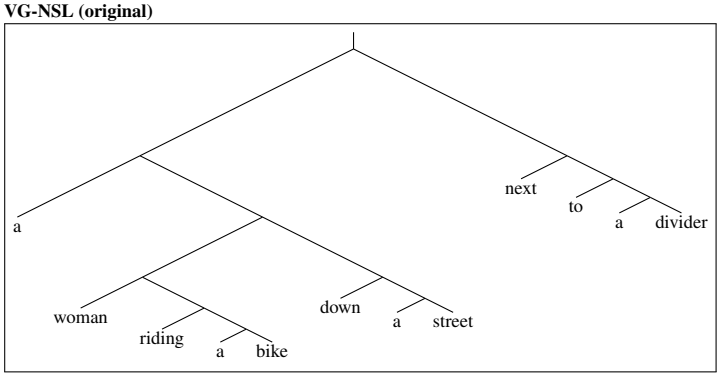
Simplified VG-NSL $(1, s_{MHI}, c_{MX})$



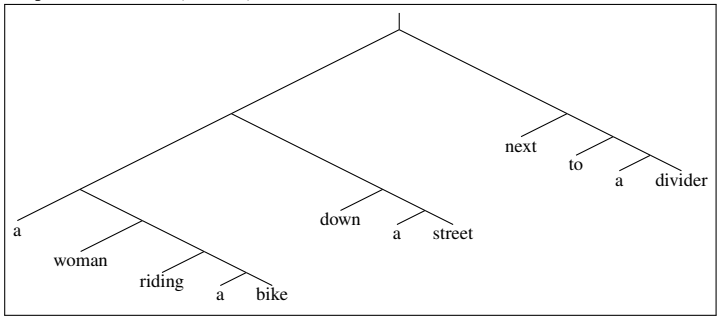
Gold Tree



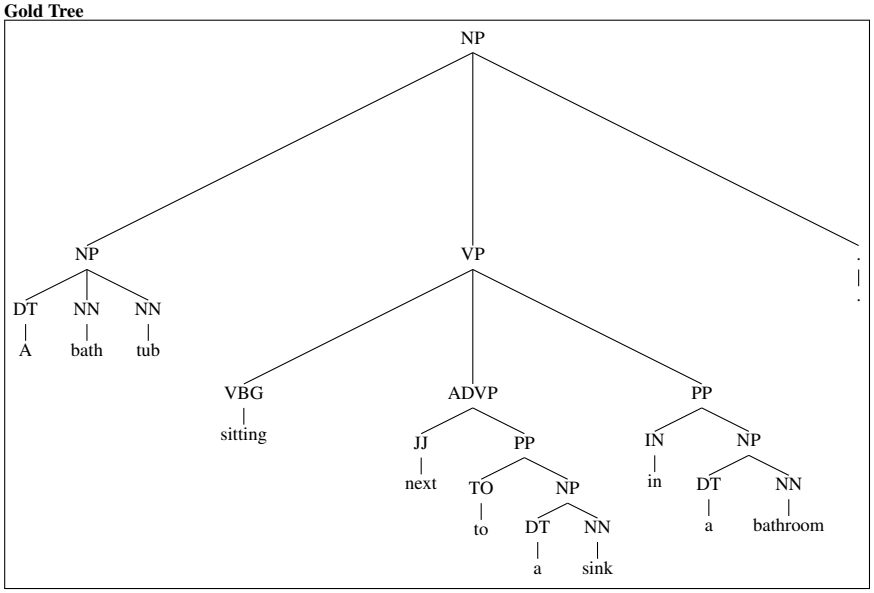
VG-NSL (original)



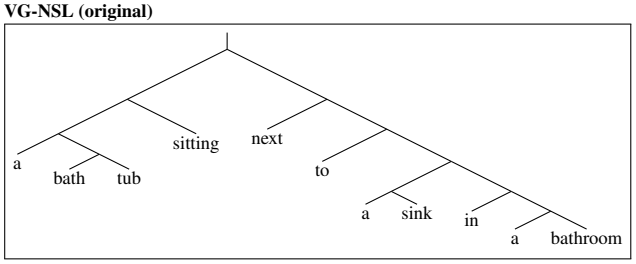
Simplified VG-NSL $(1, s_{MHI}, c_{MX})$



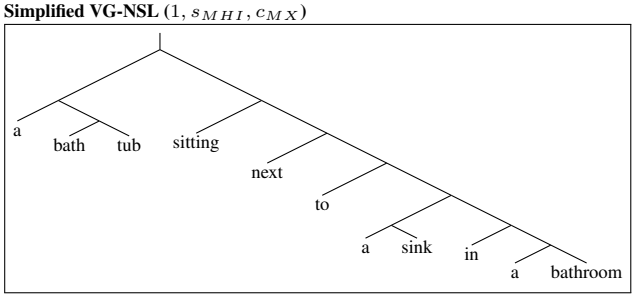
Gold Tree



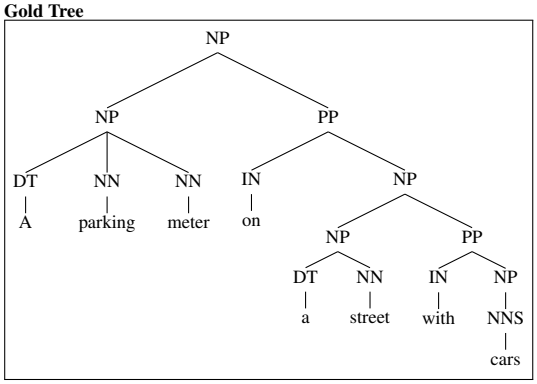
VG-NSL (original)



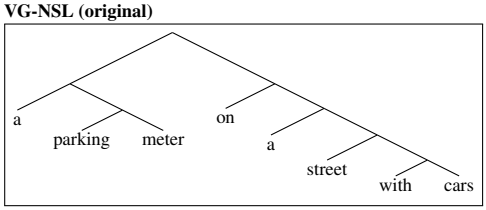
Simplified VG-NSL $(1, s_{MHI}, c_{MX})$



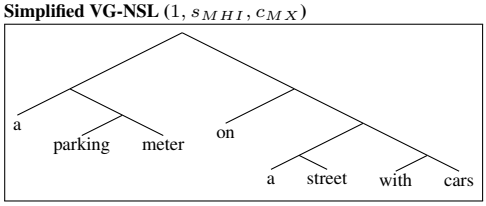
Gold Tree



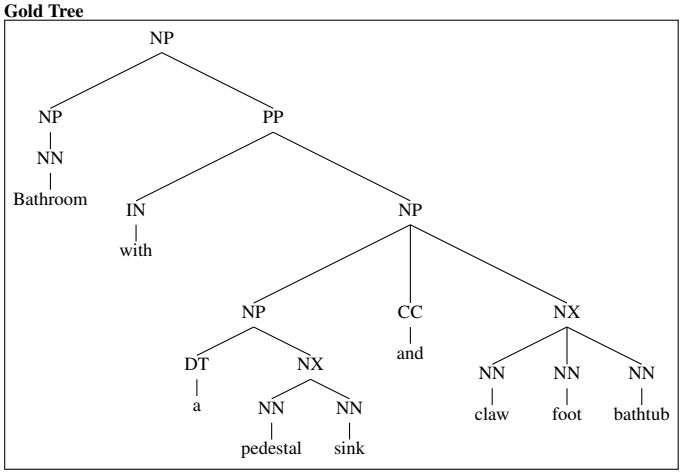
VG-NSL (original)



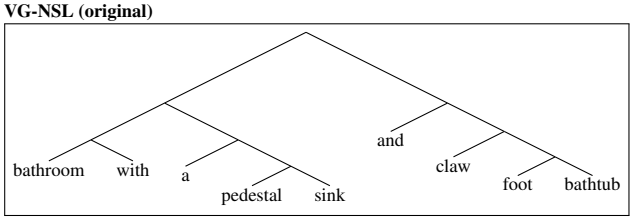
Simplified VG-NSL $(1, s_{MHI}, c_{MX})$



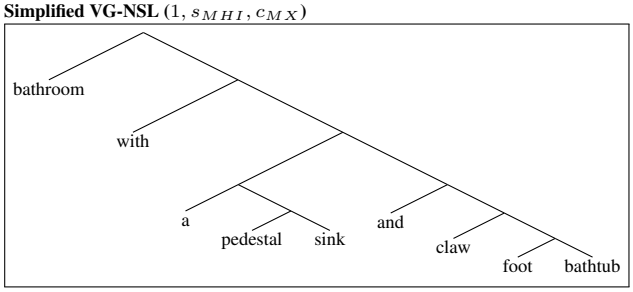
Gold Tree



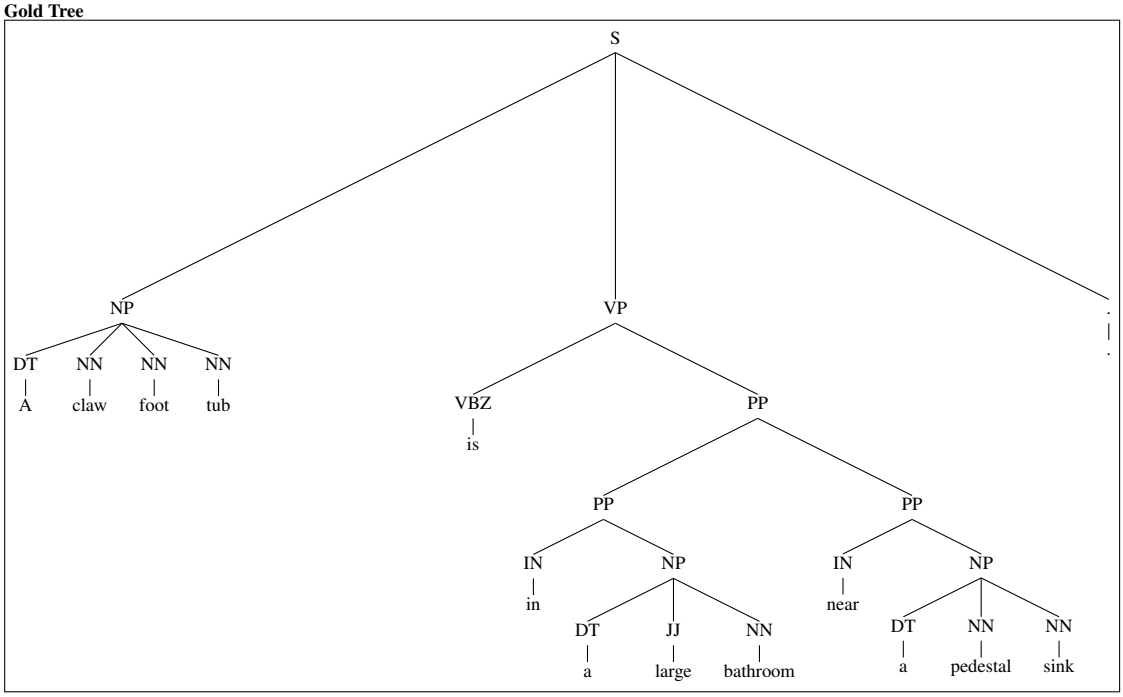
VG-NSL (original)



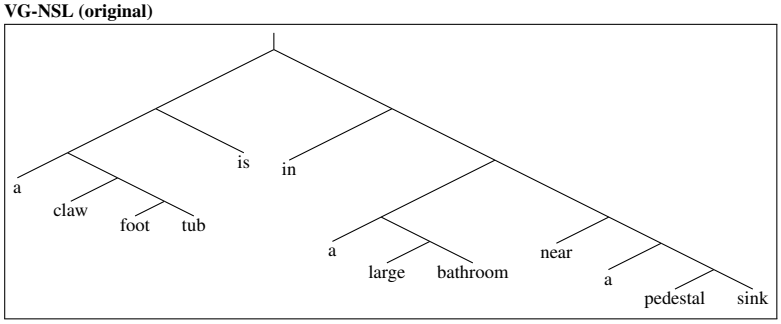
Simplified VG-NSL $(1, s_{MHI}, c_{MX})$



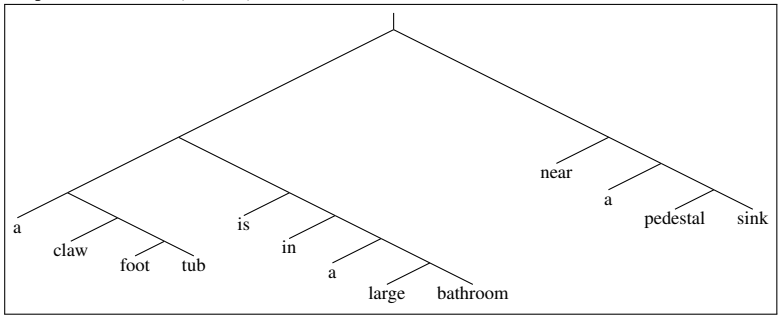
Gold Tree



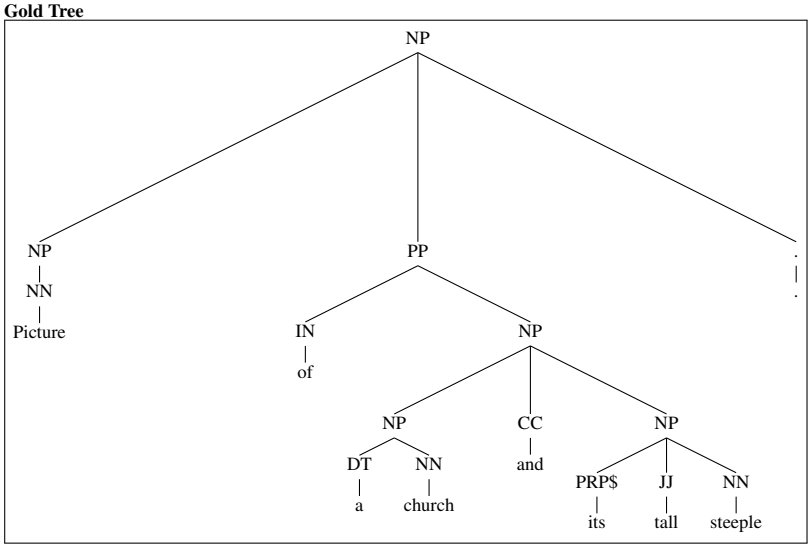
VG-NSL (original)



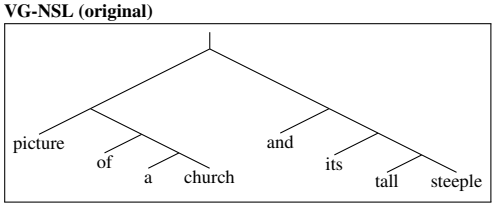
Simplified VG-NSL $(1, s_{MHI}, c_{MX})$



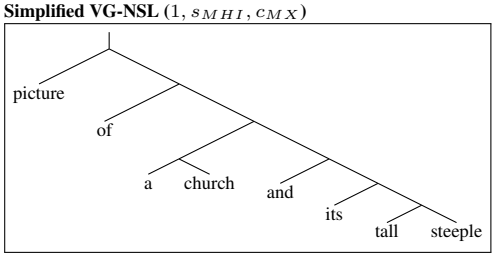
Gold Tree



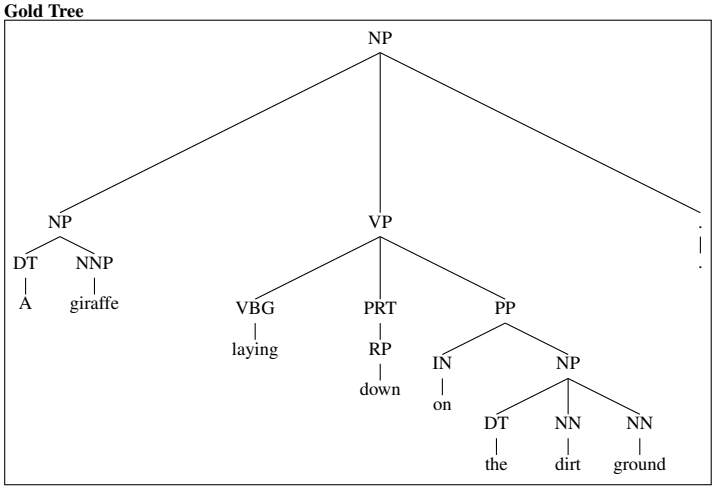
VG-NSL (original)



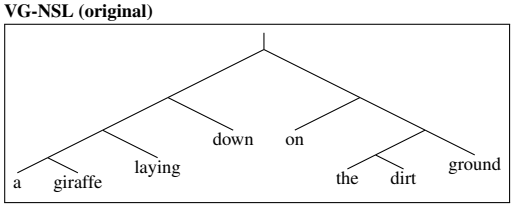
Simplified VG-NSL $(1, s_{MHI}, c_{MX})$



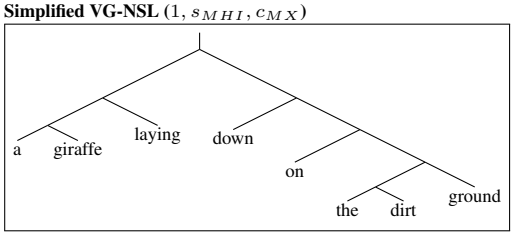
Gold Tree



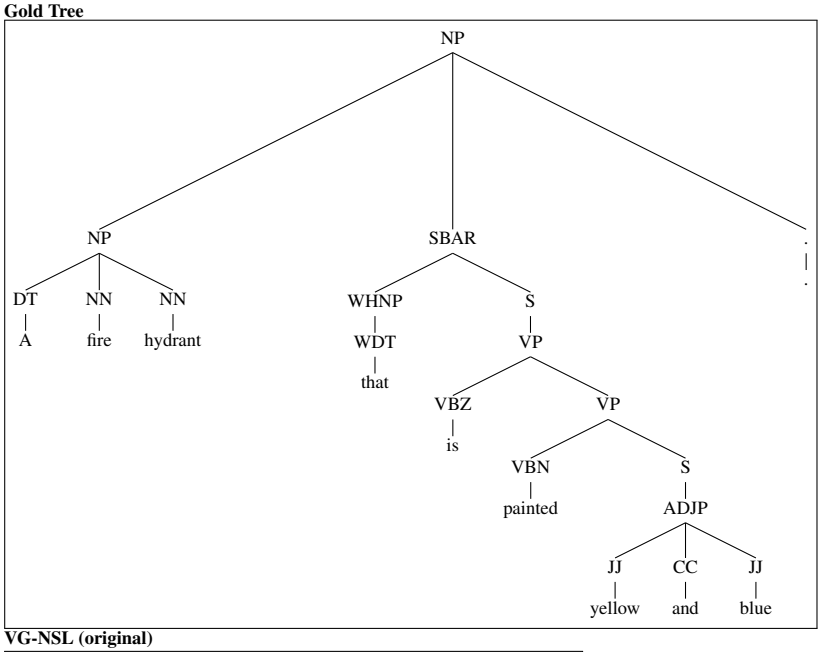
VG-NSL (original)



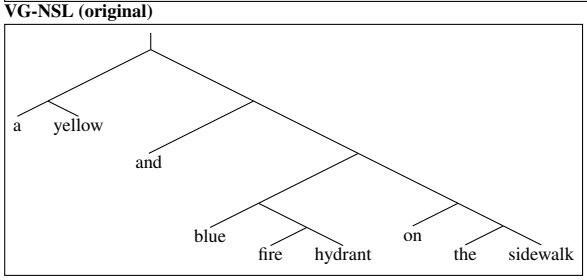
Simplified VG-NSL $(1, s_{MHI}, c_{MX})$



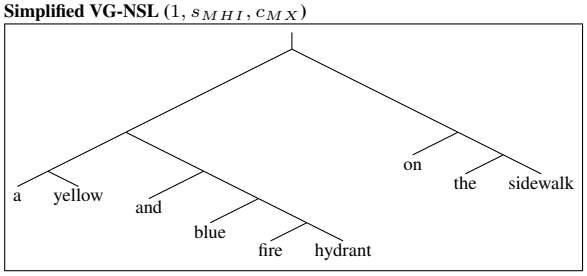
Gold Tree



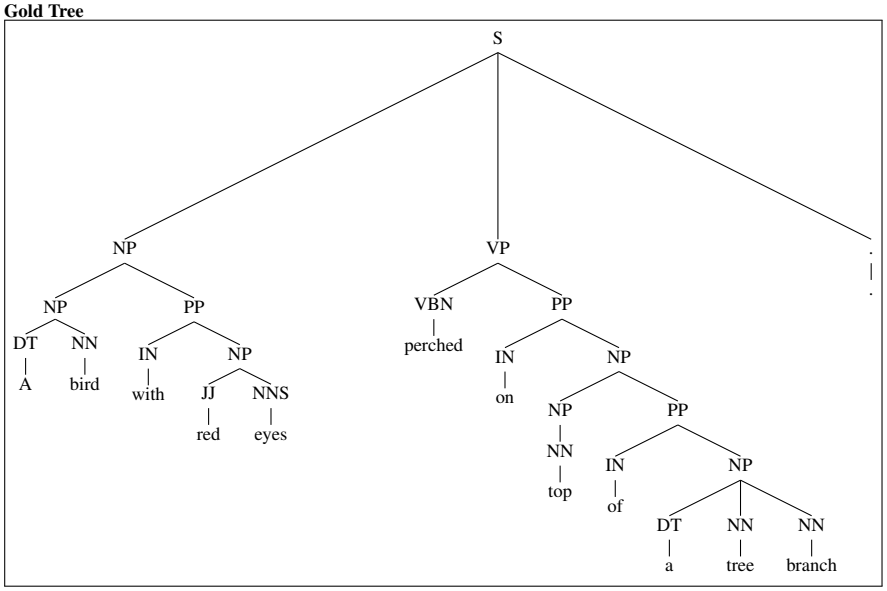
VG-NSL (original)



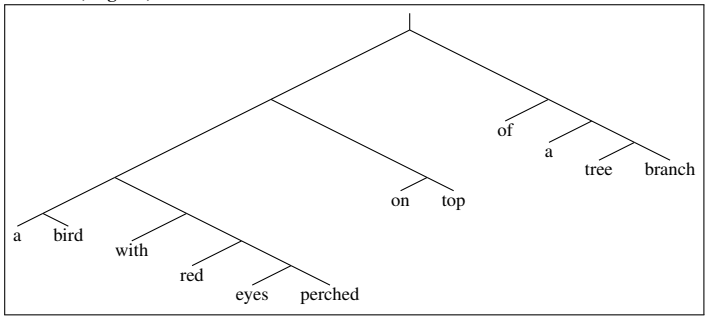
Simplified VG-NSL $(1, s_{MHI}, c_{MX})$



Gold Tree



VG-NSL (original)



Simplified VG-NSL $(1, s_{MHI}, c_{MX})$