Re-evaluating the Need for Multimodal Signals in Unsupervised Grammar Induction

Boyi Li[*, 1] **Rodolfo Corona**[*, 1] **Karttikeya Mangalam**[*, 1] **Catherine Chen**[*, 1]
Daniel Flaherty[1] **Serge Belongie**[2] **Kilian Q. Weinberger**[3]
Jitendra Malik[1] **Trevor Darrell**[1] **Dan Klein**[1]

[1]UC Berkeley [2]University of Copenhagen [3]Cornell University

Abstract

Are multimodal inputs necessary for grammar induction? Recent work has shown that multimodal training inputs can improve grammar induction. However, these improvements are based on comparisons to weak text-only baselines that were trained on relatively little textual data. To determine whether multimodal inputs are needed in regimes with large amounts of textual training data, we design a stronger text-only baseline, which we refer to as LC-PCFG. LC-PCFG is a C-PFCG that incorporates embeddings from text-only large language models (LLMs). We use a fixed grammar family to directly compare LC-PCFG to various multimodal grammar induction methods. We compare performance on four benchmark datasets. LC-PCFG provides an up to 17% relative improvement in Corpus-F1 compared to state-of-the-art multimodal grammar induction methods. LC-PCFG is also more computationally efficient, providing an up to 85% reduction in parameter count and 8.8× reduction in training time compared to multimodal approaches. These results suggest that multimodal inputs may not be necessary for grammar induction, and emphasize the importance of strong vision-free baselines for evaluating the benefit of multimodal approaches.

1 Introduction

Prior studies have shown that multimodal inputs can facilitate grammar induction. These studies paired text with inputs from images and videos, and found that models trained with paired multimodal inputs outperform text-only models on grammar induction (Shi et al., 2019; Zhao and Titov, 2020; Zhang et al., 2021, 2022a). These results suggest that multimodal inputs improve grammar induction by grounding textual inputs to the visual world. Indeed, a long line of work in human language learning suggests that paired multimodal inputs are crucial for language acquisition in humans (Gleitman, 1990; Pinker, 1984). While multimodal inputs can undoubtedly help with grammar induction, especially in regimes with low textual data, are multimodal inputs *necessary* to learn a grammar? To investigate this question, we test whether the benefits of multimodal inputs for grammar induction can be achieved by more textual data.

Prior studies of multimodal grammar induction compared multimodal methods to weak text-only baselines which were trained with relatively little data (Shi et al., 2019; Zhao and Titov, 2020; Zhang et al., 2021, 2022a). However, recent grammar induction approaches that incorporate representations from large language models (LLMs) produced large improvements in text-only grammar induction performance (e.g., Cao et al., 2020; Drozdov et al., 2019; Li and Lu, 2023). The performance of these LLM-based grammar induction methods suggest that exposure to larger quantities of textual training data can substantially improve grammar induction. However, prior studies used different settings to evaluate multimodal and text-only methods for grammar induction. Thus, it is unclear whether the performance of LLM-based grammar induction approaches can match the performance of multimodal approaches.

Here we compare multimodal methods for grammar induction to a strong text-only baseline. Our text-only baseline, which we refer to as LC-PCFG, is a C-PCFG that incorporates embeddings from text-only LLMs. We use and use a fixed grammar family (C-PCFGs) to directly compare LC-PCFG to multimodal methods, and perform comparisons with four multimodal grammar induction datasets. We find that compared to previous state-of-the-art multimodal methods, LC-PCFG achieves up to 17% relative improvement in Corpus-F1 score while requiring 8.8× less time to train. Moreover, the benefits of incorporating LLM embeddings does not straightforwardly stack with the benefits of multimodal training inputs: adding mul-

Findings of the Association for Computational Linguistics: NAACL 2024, pages 1113–1123
June 16-21, 2024 ©2024 Association for Computational Linguistics

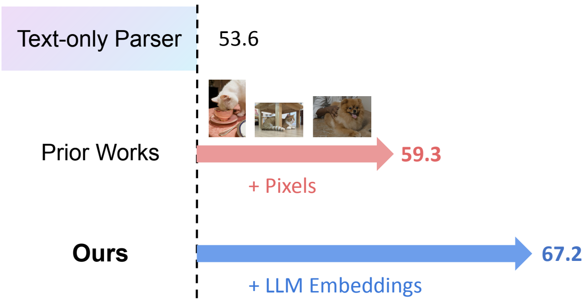

Figure 1: Comparison with prior multimodal methods on image-assisted grammar induction. Prior works showed that paired images can improve grammar induction (53.6 → 59.3 Corpus-level F1). We show that a strong text-only baselined line that incorporates embeddings from large language models (LLM) can match (and surpass) multimodal methods, suggesting that multimodal inputs may not be necessary for grammar induction (53.6 → 67.2).

timodal training inputs to LC-PCFG does not improve performance on grammar induction, suggesting that the benefits of multimodal inputs may be subsumed by training on large quantities of text. While multimodal training inputs may be useful in some settings, our results suggest that grammar induction may not require multimodal inputs. To facilitate further research we release our code at https://github.com/Boyiliee/Vision-free-Multimodal-Grammar-Induction.

2 Related Work

Grammar induction Grammar induction, the task of inducing syntactic structure without explicit supervision, has been extensively studied over the past few decades (e.g., Lari and Young, 1990; Carroll and Charniak, 1992; Clark, 2001; Klein and Manning, 2002; Smith and Eisner, 2005).

Many methods for grammar induction train on data from text alone (e.g., Lari and Young, 1990; Carroll and Charniak, 1992; Klein and Manning, 2002; Shen et al., 2018, 2019). However, based on the intuition that multimodal inputs capture information that is missing in text, recent studies have devised methods for grammar induction that incorporate information from images and videos (Shi et al., 2019; Zhao and Titov, 2020; Zhang et al., 2021, 2022a). These multimodal methods have been shown to outperform some text-only methods (Zhao and Titov, 2020; Shi et al., 2019; Zhang et al., 2022a, 2021).

LLM features for grammar induction. Recent advances in LLMs have enabled vast improvements on a wide range of downstream tasks, including both supervised syntactic parsing and grammar induction (e.g., Devlin et al., 2019; Radford et al., 2019; Kitaev et al., 2018; Cao et al., 2020; Drozdov et al., 2019; Li and Lu, 2023). However, prior work that evaluated the benefit of multimodal inputs for grammar induction used text-only baselines that incorporated much weaker word representations, such as random word embeddings or lexical word embeddings such as fastText. Thus, it is unclear whether stronger text-only methods for grammar induction can match the performance of multimodal approaches.

3 LC-PCFG: Grammar Induction with Large Language Models

The goal of grammar induction is to learn syntactic structure without explicit supervision. Methods for grammar induction assume a grammar formalism and then optimize grammar parameters to fit the data. We use Compound Probabilistic Context-Free Grammars (C-PCFGs) (Kim et al., 2019) as a grammar formalism. We construct a C-PCFG that incorporates LLM representations. We refer to this method as LC-PCFG. We compare LC-PCFG to prior methods that incorporate multimodal data (Zhao and Titov, 2020; Zhang et al., 2021, 2022a). Figure 2 provides an overview of our experiments.

Background. C-PCFGs extend the Probabilistic Context Free Grammar (PCFG) formalism, and are defined by a 5-tuple $\mathcal{G} = (S, \mathcal{N}, \mathcal{P}, \Sigma, \mathcal{R})$, consisting of a start symbol S, a set of non-terminals $\mathcal{N}$, a set of pre-terminals $\mathcal{P}$, a set of terminals Σ, and a set of derivation rules $\mathcal{R}$:

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

PCFGs define a probability distribution over transformation rules $\boldsymbol{\pi} = \{\pi_r\}_{r \in \mathcal{R}}$. Then the inside algorithm (Baker, 1979) can be used to efficiently perform inference over this probability distribution. In neural PCFGs, this distribution may be formu-

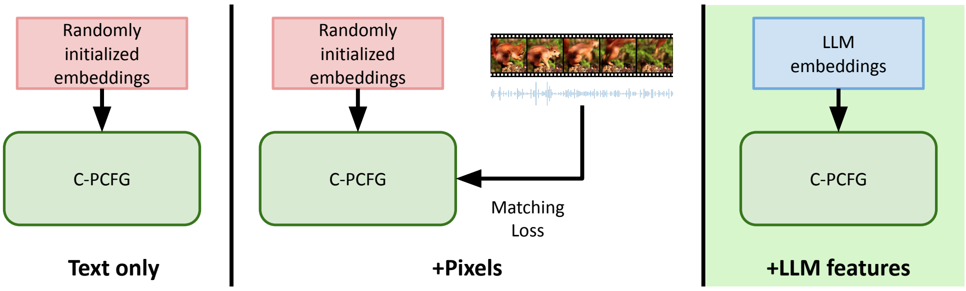

Figure 2: **Experimental Settings.** We explore using large language model features for unsupervised grammar induction. We use three experimental settings. (1) the standard setting in which word representations are learned from scratch (**Text Only**), (2) prior methods that incorporate a multimodal regularization loss (**+Pixels**), and (3) our method, which uses pre-trained text-only LLM features (**+LLM features**). We show that LLM features can obtain state-of-the-art performance, without requiring multimodal regularization.

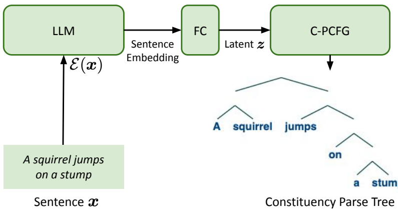

Figure 3: **LC-PCFG workflow.** A sentence $\boldsymbol{x}$ is fed to an LLM to obtain a sentence embedding $\mathcal{E}(\boldsymbol{x})$. $\mathcal{E}(\boldsymbol{x})$ is passed through a fully-connected layer (FC), producing the latent $\boldsymbol{z}$. $\boldsymbol{z}$ is fed to the C-PCFG to obtain a constituency parse tree. Note that unlike prior work, our approach does not require multimodal data.

lated as follows:

$$\pi_{S \to A} = \frac{\exp(\boldsymbol{u}_A^\top f_1(\boldsymbol{w}_S))}{\Sigma_{A' \in \mathcal{N}} \exp(\boldsymbol{u}_{A'}^\top f_1(\boldsymbol{w}_S))}$$

$$\pi_{A \to BC} = \frac{\exp(\boldsymbol{u}_{BC}^\top \boldsymbol{w}_A)}{\Sigma_{B'C' \in \mathcal{M}} \exp(\boldsymbol{u}_{B'C'}^\top \boldsymbol{w}_A)}$$

$$\pi_{T \to w} = \frac{\exp(\boldsymbol{u}_w^\top f_2(\boldsymbol{w}_T))}{\Sigma_{w' \in \Sigma} \exp(\boldsymbol{u}_{w'}^\top f_2(\boldsymbol{w}_T))}$$

where $\boldsymbol{u}$ are transformation vectors for each production rule, $\boldsymbol{w}$ are learnable parameter vectors for each symbol, and f_1 and f_2 are neural networks. The neural PCFG formulation preserves the benefits of fast inference while additionally incorporating distributional representations from neural networks.

Because PCFGs contain a strong context-free assumption, PCFGs cannot leverage global information that is useful for computing production probabilities during inference. C-PCFGs (Kim et al., 2019) extend PCFGs to incorporate global information. C-PCFGs formulate rule probabilities as a compound probability distribution (Robbins, 1956):

$$\boldsymbol{z} \sim p_\gamma(\boldsymbol{z}) \qquad \pi_{\boldsymbol{z}} = f_\lambda(\boldsymbol{z}, \boldsymbol{E}_\mathcal{G})$$

Where $\boldsymbol{z}$ is a latent variable generated by a prior distribution (generally assumed to be spherical Gaussian) and $\boldsymbol{E}_\mathcal{G} = \{\boldsymbol{w}_N | N \in \{S\} \cup \mathcal{N} \cup \mathcal{P}\}$ denotes the set of symbol embeddings. Rule probabilities $\boldsymbol{\pi_z}$ are additionally conditioned on this latent variable:

$$\pi_{\boldsymbol{z}, S \to A} \propto \exp(\boldsymbol{u}_A^\top f_1([\boldsymbol{w}_S; \boldsymbol{z}])),$$
$$\pi_{\boldsymbol{z}, A \to BC} \propto \exp(\boldsymbol{u}_{BC}^\top [\boldsymbol{w}_A; \boldsymbol{z}]),$$
$$\pi_{\boldsymbol{z}, T \to w} \propto \exp(\boldsymbol{u}_w^\top f_2([\boldsymbol{w}_T; \boldsymbol{z}]))$$

Importantly, the latent variable $\boldsymbol{z}$ allows global information to be shared across production decisions during, while maintaining the context-free assumption needed for efficient inference when $\boldsymbol{z}$ is fixed.

Becase the introduction of $\boldsymbol{z}$ makes inference intractable, variational methods are used to optimize C-PCFGs (Kingma and Welling, 2013) At inference time, given a sentence $\boldsymbol{x}$, the variational inference network q_ϕ is used to produce the latent $\boldsymbol{z} = \boldsymbol{\mu}_\phi(g(\mathcal{E}(\boldsymbol{x})))$. Here, g is a sentence encoder used to generate a vector representation given token embeddings $\mathcal{E}(\boldsymbol{x})$. For a more thorough treatment of C-PCFGs, please see Kim et al. (2019).

LLM-based C-PCFG for grammar induction. We design LC-PCFG, a simple but strong text-only baseline which incorporates pre-trained LLM representations into the C-PCFG inference network. Specifically, we formulate the inference network as:

$$\mathcal{E}(\boldsymbol{x}) = \text{LLM}(\boldsymbol{x}) \tag{1}$$

$$g(\boldsymbol{x}) = \text{FC}(m(\mathcal{E}(\boldsymbol{x}))) \tag{2}$$

Table 1: **Grammar induction with image and text**. Corpus-level F1 (C-F1) and sentence-level F1 (S-F1) scores on the MSCOCO 2014 caption dataset. We compare LC-PCFG against simple rule-based baselines (top, from (Zhao and Titov, 2020)), prior state-of-the-art methods that employ image data (middle), and methods, including ours, that use purely textual data (bottom). *RGB* indicates whether each method uses multimodal inputs. LC-PCFG outperforms all prior multimodal methods.

Method	RGB	LLM	Params (M)	C-F1	S-F1
Rule-based baselines					
Left Branching	No	No	-	15.1	15.7
Right Branching	No	No	-	51.0	51.8
Random Trees	No	No	-	24.2$_{\pm 0.3}$	24.6$_{\pm 0.2}$
Methods using extra-linguistic inputs					
VG-NSL (Shi et al., 2019)	Yes	No	-	50.4$_{\pm 0.3}$	-
VC-PCFG (Zhao and Titov, 2020)	Yes	No	41.5	59.3$_{\pm 8.2}$	59.4$_{\pm 8.3}$
VC-PCFG++	Yes	No	41.5	64.2$_{\pm 7.0}$	64.6$_{\pm 7.2}$
Methods using only textual inputs					
C-PCFG (Kim et al., 2019)	No	No	15.3	53.6$_{\pm 4.7}$	53.7$_{\pm 4.6}$
LC-PCFG (Ours)	No	Yes	**6.2**	**67.2**$_{\pm 1.1}$	**67.8**$_{\pm 1.2}$

where m represents a mean-pool operation. Here, an LLM is used to obtain text embeddings for each sentence $\boldsymbol{x}$, which are then fed to a fully connected (FC) layer as the C-PCFG inference network. Figure 3 provides an example of this method. A sentence $\boldsymbol{x}$ (“*A squirrel jumps on a stump*") is fed into an LLM to obtain an embedding of the sentence. Then the sentence embedding is passed into a fully-connected layer to obtain the latent variable $\boldsymbol{z}$. Finally, we feed $\boldsymbol{z}$ into the C-PCFG to obtain a constituency parse tree. Note that compared to prior multimodal CPFGs which used multimodal inputs for regularization, our approach does not use any multimodal data.

4 Experiments

4.1 Image-assisted Parsing

We compare LC-PCFG against VG-NSL (Shi et al., 2019) and VC-PCFG (Zhao and Titov, 2020), two state-of-the-art multimodal grammar induction methods that incorporate visual signals from paired image-caption data. In VG-NSL and VC-PCFG, a visual matching loss between representations of images and their captions serves as a regularizer during grammar induction.

Setup. We follow the experimental setup of Zhao and Titov (2020), evaluating on the same splits of the MSCOCO 2014 dataset (Lin et al., 2014). (Because MSCOCO does not provide captions for their test set, a portion of the validation set is used as a held-out test set.) Images in the MSCOCO dataset are each associated with 5 captions. The final dataset consists of 82,783 training, 1,000 validation, and 1,000 test images. During preprocessing, all sentences are converted to lowercase and numbers are replaced with the letter "N". For models using word embedding matrices, the most frequent 10,000 words (based on whitespace tokenization) are maintained with all other words mapped to a special UNK token. Captions greater than 45 words in length are removed. For LC-PCFG, we preprocess the dataset by extracting token-level embeddings for each caption from the last layer of an LLM.

Evaluation. Because the MSCOCO dataset does not have annotated ground truth parse trees, we follow prior work and use a supervised neural parser, Benepar (Kitaev and Klein, 2018), to generate parse trees for evaluation. Each unsupervised grammar induction method is evaluated by computing the F1 score between the predicted parse tree and the parse tree generated by Benepar. Due to instabilities observed during training, each method is trained with 10 random seeds and then the mean and standard deviation over the top 4 seeds (based on validation F1) are reported.

Implementation. For baseline models we use the implementation and hyperparameters provided by Zhao and Titov (2020).[1]

The original implementation of VC-PCFG uses a ResNet-152 network to embed images. However, there are now image embedding networks that are stronger than ResNet-152. To provide a fair comparison between our text-only model and multimodal approaches, we improve VC-PCFG by replacing ResNet-152 with ResNetV1.5

[1] https://github.com/zhaoyanpeng/vpcfg

Table 2: **Grammar induction with video and text**. Comparison across three video-text parsing benchmark datasets (DiDeMo, YouCook2 & MSRVTT). We show performance of simple rule-based baselines (top), prior state-of-the-art multimodal methods (middle) and text-only models including ours (LC-PCFG) (bottom). LC-PCFG outperforms all prior methods.

PCFG Method	LLM	RGB	DiDeMo		YouCook2		MSRVTT	
			C-F1	S-F1	C-F1	S-F1	C-F1	S-F1
Rule-based baselines								
Left Branching	No	No	16.2	18.5	6.8	5.9	14.4	16.8
Right Branching	No	No	53.6	57.5	35.0	41.6	54.2	58.6
Random	No	No	29.4$_{\pm0.3}$	32.7$_{\pm0.5}$	21.2$_{\pm0.2}$	24.0$_{\pm0.2}$	27.2$_{\pm0.1}$	30.5$_{\pm0.1}$
Methods using extra-linguistic inputs								
VC-PCFG (Zhao and Titov, 2020)	No	Yes	42.2$_{\pm12.3}$	43.2$_{\pm14.2}$	42.3$_{\pm5.7}$	47.0$_{\pm5.6}$	49.8$_{\pm4.1}$	54.2$_{\pm4.0}$
MMC-PCFG (Zhang et al., 2021)	No	Yes	55.0$_{\pm3.7}$	58.9$_{\pm3.4}$	44.7$_{\pm5.2}$	48.9$_{\pm5.7}$	56.0$_{\pm1.4}$	60.0$_{\pm1.2}$
Methods using only textual inputs								
C-PCFG (Kim et al., 2019)	No	No	38.2$_{\pm5.0}$	40.4$_{\pm4.1}$	37.8$_{\pm6.7}$	41.4$_{\pm6.6}$	50.7$_{\pm3.2}$	55.0$_{\pm3.2}$
LC-PCFG (Ours)	Yes	No	**57.1**$_{\pm4.7}$	**60.0**$_{\pm5.2}$	**52.4**$_{\pm0.1}$	**57.7**$_{\pm0.1}$	**56.1**$_{\pm3.6}$	**61.2**$_{\pm3.7}$

- 152 (ResNetV1.5, 2022). We also improve the optimization hyperparameters (learning rate and network dropout). We refer to the modernized version of VC-PCFG as VC-PCFG++. VC-PCFG++ outperforms VC-PCFG by about 3 points in both corpus and sentence-level F1 scores.

For LC-PCFG, we use an OPT-2.7B (Zhang et al., 2022b) model to extract token-level embeddings for each sentence. Sentence embeddings are then mean-pooled and passed through a single linear layer inference network. We use dropout of 0.5 on both the mean-pooled sentence embedding and the output latent vector from the inference network.

4.1.1 Results

Table 1 shows test F1 scores for each model. LC-PCFG achieves the highest overall corpus-level F1 (C-F1) and sentence-level F1 (S-F1) scores. Note that LC-PCFG does not use paired visual features, and contains 85% fewer parameters than the previous state-of-the-art approach (VC-PCFG).

4.2 Video-assisted Parsing

The results in Table 1 show that a text-only approach can outperform approaches that incorporate multimodal inputs from images. However, some have argued that images are a static snapshot of the world, and therefore may lack information needed to induce verb phrases (Zhang et al., 2021). Based on the intuition that video can provide better multimodal training signals, one study presented an approach for grammar induction (MultiModal Compound PCFG; MMC-PCFG) that incorporates both visual and auditory signals from videos (Zhang et al., 2021). MMC-PCFG aggregates multimodal features and achieved a substantial improvement over previous multimodal methods for grammar induction. To test whether a text-only baseline can achieve the same improvements as a video-enhanced method, we compare LC-PCFG to MMC-PCFG.

Setup. Following Zhang et al. (2021), we use three benchmarking video datasets for our experiments: Distinct Describable Moments (*DiDeMo*) (Anne Hendricks et al., 2017), Youtube Cooking (*YouCook2*) (Zhou et al., 2018) and MSRVideo to Text (*MSRVTT*) (Xu et al., 2016). DiDeMo consists of unedited, personal videos in diverse visual settings with pairs of localized video segments and referring expressions. It includes 32994, 4180 and 4021 video-sentence pairs in the training, validation, and test sets. YouCook2 contains 2000 videos that are nearly equally distributed over 89 recipes. Each video contains 3–16 procedure segments. It includes 8713, 969 and 3310 video-sentence pairs in the training, validation and test sets. MSRVTT is a large-scale benchmark for video understanding with 10K web video clips with 41.2 hours and 200K clip-sentence pairs in total. It includes 130260, 9940 and 59794 video-sentence pairs across all the data splits.

The extracted multimodal features (Zhang et al., 2021) include object features (SENet (Xie et al., 2017)), action features (I3D (Carreira and Zisserman, 2017)), scenes (Huang et al., 2017; Zhou et al., 2017), audio (Hershey et al., 2017), OCR (Deng et al., 2018; Liu et al., 2018), faces (Liu et al., 2016; He et al., 2016) and speech (Mikolov et al., 2013).

We run all experiments 4 times for 10 epochs each, with different random seeds. We report the mean and standard deviation of the C-F1 and S-F1

Table 3: **Transferring Learnt Grammar**. Models are trained on the 'Trainset' data and evaluated without additional training on the target benchmarks (DiDeMO, YouCook2 & MSRVTT) on the Sentence-level F1 (S-F1) and Corpus-level F1 (C-F1) metrics. All HowTo100M results are reported on 592k samples.

Method	Trainset	DiDeMo		YouCook2		MSRVTT	
		C-F1	S-F1	C-F1	S-F1	C-F1	S-F1
MMC-PCFG	DiDeMo	55.0$_{\pm 3.7}$	58.9$_{\pm 3.4}$	49.1$_{\pm 4.4}$	53.0$_{\pm 4.9}$	49.6$_{\pm 1.4}$	53.8$_{\pm 0.9}$
MMC-PCFG	YouCook2	40.1$_{\pm 4.4}$	44.2$_{\pm 4.4}$	44.7$_{\pm 5.2}$	48.9$_{\pm 5.7}$	34.0$_{\pm 6.4}$	37.5$_{\pm 6.8}$
MMC-PCFG	MSRVTT	59.4$_{\pm 2.9}$	62.7$_{\pm 3.3}$	49.6$_{\pm 3.9}$	54.2$_{\pm 4.1}$	56.0$_{\pm 1.4}$	60.0$_{\pm 1.2}$
MMC-PCFG	HowTo100M	58.5$_{\pm 7.3}$	62.4$_{\pm 7.9}$	53.9$_{\pm 6.6}$	58.0$_{\pm 7.1}$	55.1$_{\pm 7.0}$	60.2$_{\pm 8.0}$
PTC-PCFG	HowTo100M	**61.3**$_{\pm 3.9}$	**65.2**$_{\pm 5.3}$	58.9$_{\pm 2.5}$	63.2$_{\pm 2.3}$	57.4$_{\pm 4.6}$	62.8$_{\pm 5.7}$
LC-PCFG (Ours)	HowTo100M	60.6$_{\pm 5.2}$	61.5$_{\pm 6.1}$	**61.1**$_{\pm 2.1}$	**65.2**$_{\pm 1.4}$	**59.4**$_{\pm 5.0}$	**63.0**$_{\pm 5.8}$

scores.

4.2.1 Results

Table 2 compares grammar induction performance between C-PCFG, VC-PCFG (which incorporates visual signals), and MMC-PCFG (which incorporates signals from multiple extralinguistic modalities). LC-PCFG outperforms the video-regularized models for all three benchmark datasets.

4.3 Large-scale Video Pretraining

While MMC-PCFG incorporates multimodal inputs from small amounts of video data, other work has proposed to use larger scale video data for improve grammar induction (Zhang et al., 2022a). That work proposed Pre-Trained Compound PCFGs (PTC-PCFG), a multimodal method for grammar induction that obtains paired video and text inputs from captioned instructional YouTube videos in the *HowTo100M* dataset (Miech et al., 2019). Then a matching loss between these paired inputs is used as a regularizer during grammar induction. PTC-PCFG outperformed previous state-of-the-art multimodal grammar induction models.

To determine how PTC-PCFG compares to our text-only baseline, we train LC-PCFG with the captions of the *HowTo100M*dataset (Miech et al., 2019) without using any multimodal inputs. Following Zhang et al. (2022a), we induce a grammar from 592k samples of the HowTo100M train set and then evaluate on the three video-enhanced parsing benchmarks shown in Table 2 (DiDeMo, YouCook2, and MSRVTT).

Table 3 shows the test F1 scores for MMC-PCFG, PTC-PCFG, and LC-PCFG on the three video-enhanced parsing benchmarks. LC-PCFG outperforms MMC-PCFG, even in settings where LC-PCFG is trained on out-of-distribution HowTo100M dataset and MMC-PCFG is trained on in-distribution samples from each benchmark dataset. On the three benchmarks, LC-PCFG either outperforms or nearly matches PTC-PCFG.

Table 4: **Training Time Evaluation** for both image-based (top) and video-based (bottom) grammar induction methods. Run-time for both pre-extracting the embeddings ('Embedding') and model training ('Training') are reported. We pre-embed captions for LC-PCFG with two 24GB Titan RTX GPUs and pre-embed images/videos for models with a visual component. Training times for image and video results are benchmarked on a single 12G 2080 Ti and on 2× 32G V100s respectively.

PCFG Method	Embedding (hours)	Training (hours)
C-PCFG	-	7.6
VC-PCFG	0.25	13.3
LC-PCFG	2.0	8.0
C-PCFG	-	1.5
MMC-PCFG	>25	15
PTC-PCFG	>25	10
LC-PCFG (Ours)	2.5	1.7

4.4 Runtime Comparison

To compare the runtime of each method, we follow the setting of PTC-PCFG and calculate the time to extract embeddings and train each model. Table 4 shows the runtime for each model. LC-PCFG requires more time for embedding extraction than VC-PCFG, but LC-PCFG results 10× less time for embedding extraction time compared to video-enhanced models. LC-PCFG is 1.3 to 8.8 times faster to train than either image-enhanced or video-enhanced models.

5 Model Analysis

5.1 Perplexity-based Evaluation

To facilitate comparisons between methods, the results reported in Section 4 are based on the model selection procedure used in prior studies (Zhao and Titov, 2020; Zhang et al., 2021, 2022a). This model selection procedure trains models with different random seeds, and then uses validation C-F1 score to choose a subset of random seeds for test evaluation.

However, this model selection procedure assumes that gold parse trees are available during validation. To ensure that our results do not rely on having gold trees at validation time, we repeat our experiments but instead use perplexity (PPL) (Chen et al., 1998) to perform model selection:

$$PPL(X) = -\frac{1}{t}\sum_{i}^{t} log\ p(x_i|x_{<i})$$

where $X = (x_1, x_2, x_3, ..., x_t)$ is a tokenized sequence of words and $p(x_i|x_{<i})$ represents the log-likelihood of the ith token conditioned on the preceding tokens $x_{<i}$.

PPL allows us to perform model selection without relying on gold parse trees. We train models with 10 random seeds, and then use PPL to select the four best-performing seeds.

Table 5 shows test C-F1 performance on image-assisted parsing for experiments in which we used PPL to perform model selection. LC-PCFG consistently outperforms methods that use multimodal inputs. We observe the same results for video-assisted parsing (Table 6).

Table 5: **Unsupervised Run Selection Criterion for Unsupervised Grammar Induction**. Corpus-level F1 scores using validation set F1 ('Val-F1'), perplexity ('PPL'), and mean branching factor ('MBF', the average proportion between leaves in the right and left branches of nodes in each tree across the corpus). Unlike Validation-F1 based-selection, PPL and MBF do not require gold trees during validation.

PCFG Method	Run Selection Criteria		
	Val-F1	PPL	MBF
C-PCFG	60.1$_{\pm 4.6}$	52.0$_{\pm 7.5}$	56.8$_{\pm 9.3}$
VC-PCFG	61.3$_{\pm 2.6}$	55.3$_{\pm 10.2}$	51.0$_{\pm 13.4}$
LC-PCFG (Ours)	**67.2**$_{\pm 1.1}$	**67.2**$_{\pm 1.1}$	**65.3**$_{\pm 2.1}$

5.2 Branching Factor

We performed grammar induction over texts in English, which is a right-branching language. To investigate whether induced grammars capture the right-branching nature of English, we measure the branching factor of predicted parse trees. For each branch in each parse tree we measure the proportion of leaves under the right branch over those of the left branch. This proportion is then averaged across all nodes in the tree to produce an average score s. s is referred to as the branching factor of the tree ($s > 1.0$ means that the tree is overall right-branching, whereas $s < 1.0$ means that the tree is overall left-branching). Formally, for each parse tree t with $|t|$ nodes $n \in t$ we compute the mean over nodes' ratio of leaves in their right and left branches:

$$\text{MBF}(t) = \frac{1}{|t|}\sum_{n \in t}\frac{\text{CR}(n)}{\text{CL}(n)}$$

where CR and CL are the respective counts of leaves under the right and left branches of a node.

Table 7 shows the mean branching factor (MBF) for each model (computed over 10 seeds). We find that all models predict right-branching trees, and LC-PCFG has the lowest MBF (i.e., most right-branching trees).

To test whether MBF could be used as a run selection criteria, we used MBF instead of validation C-F1 score to select random seeds. For VC-PCFG and LC-PCFG, using MBF as a seed-selection method performs slightly worse than using PPL or validation C-F1 score as a seed selection method.

5.3 Model Ablations

To understand the effect of different model components on grammar induction performance, we perform a series of ablations on parsers trained on the MSCOCO dataset.

Table 6: **Unsupervised Run Selection Criterion for Unsupervised Grammar Induction**. Similar to Table 5, we report the results of run selection based on validation perplexity (PPL) for video benchmarks (Section 5.1).

PCFG Method	DiDeMo		YouCook2		MSRVTT	
	C-F1	S-F1	C-F1	S-F1	C-F1	S-F1
Compound (Kim et al., 2019)	40.4$_{\pm 10.1}$	42.1$_{\pm 9.1}$	38.6$_{\pm 7.2}$	42.8$_{\pm 7.7}$	49.2$_{\pm 3.8}$	53.1$_{\pm 4.0}$
Multi-modal (Zhang et al., 2021)	42.1$_{\pm 12.6}$	45.7$_{\pm 12.4}$	38.9$_{\pm 3.6}$	43.8$_{\pm 3.3}$	48.1$_{\pm 1.0}$	52.4$_{\pm 0.9}$
LC-PCFG (Ours)	**46.3**$_{\pm 6.9}$	**49.9**$_{\pm 7.3}$	**46.7**$_{\pm 1.1}$	**52.4**$_{\pm 0.8}$	**50.5**$_{\pm 4.0}$	**55.2**$_{\pm 4.4}$

Table 7: **MBF on image-assisted parsing.**

PCFG Method	MBF
C-PCFG	$3.4_{\pm 0.3}$
VC-PCFG	$3.4_{\pm 0.3}$
LC-PCFG	$2.5_{\pm 0.7}$

To understand the contribution of the latent variable z, we ablate z in both training and in evaluation.

First we perform inference-time ablations. During inference-time we zero out the latent variable z ('Zero-z'), or randomly shuffle z within an evaluation batch ('Random-z'). Next we perform training-time ablations. We train a C-PCFG model without latents ('Zero-Train', a vanilla neural PCFG model).

The performance of each ablated model is shown in Table 8. We find that inference-time ablations on the latent yield comparable performance to the default parsers, whereas omitting the latent during training yields reduced performance from the standard C-PCFG/LC-PCFG models. These results suggest that the latent variable may be largely ignored at inference time, but that it serves an important role in the learning process of the parser.

Lastly, we ablate the input sentences. We evaluate parsers when shuffling ('Shuffle') or zeroing out input caption embeddings ('Zero-C'), word embeddings for VC-PCFG or LLM embeddings for LC-PCFG). We find that ablating the input sentences substantially reduces test performance, suggesting that learned parsers do not merely degenerate to a learned prior at inference time.

Table 8: **Parser Ablations.** Corpus-level F1 scores for PCFG parsers under ablations. We compare the default formulations ('Default') to conditions zeroing out the latent z ('Zero-z'), randomly shuffling latents across a batch ('Random-z'), shuffling words in each caption ('Shuffle') or zeroing captions out ('Zero-C'), as well as zeroing out latents during training ('Zero-Train'). Note that C-PCFG and LC-PCFG are functionally equivalent in the Zero-Train condition because LLM features are only used in latent computation.

Ablation	Test Corpus F1		
	C-PCFG	VC-PCFG	LC-PCFG
Default	$60.1_{\pm 4.6}$	$61.3_{\pm 2.6}$	$67.2_{\pm 1.1}$
Zero-z	$60.3_{\pm 5.2}$	$60.6_{\pm 2.6}$	$67.2_{\pm 1.1}$
Random-z	$60.3_{\pm 5.2}$	$60.9_{\pm 2.5}$	$67.2_{\pm 1.1}$
Shuffle	$30.0_{\pm 0.7}$	$31.0_{\pm 0.9}$	$40.6_{\pm 1.0}$
Zero-C	$35.2_{\pm 16.1}$	$44.6_{\pm 7.6}$	$48.6_{\pm 7.5}$
Zero-Train	$57.1_{\pm 6.5}$	$58.8_{\pm 0.9}$	$57.1_{\pm 6.5}$

Method	Params (M)	C-F1	S-F1
Ours	**6.2**	**67.2**$_{\pm 1.1}$	**67.8**$_{\pm 1.2}$
Ours + ImgFeas	32.3	$59.2_{\pm 0.5}$	$59.4_{\pm 0.5}$

Table 9: **Adding visual features to LC-PCFG.** Incorporating visual features ("ImgFeas") into LC-PCFG degrades performance.

5.4 Re-adding Visual Signals to LC-PCFG

Section 4 showed that LC-PCFG outperforms previous multimodal approaches to grammar induction. But can re-adding visual signals to LC-PCFG further improve grammar induction? Such an improvement would suggest that multimodal signals contribute to grammar induction beyond what can be learned from text alone.

To test this possibility we re-trained LC-PCFG with the addition of paired visual features. Visual features were incorporated with the same multimodal regularization loss as used in prior work (Shi et al., 2019; Zhao and Titov, 2020). Table 9 and Table 10 show the effect of adding image and video signals to LC-PCFG.

Adding visual signals to LC-PCFG reduces performance compared to the text-only version of the model. We observe this degradation across all datasets, both for pixel-based and video-based visual features. We hypothesize that LC-PCFG may overfit to the added visual features and thereby obfuscate the signals in LLM embeddings.

6 Conclusion and Future Work

We propose LC-PCFG, a strong text-only baseline for grammar induction. LC-PCFG is a C-PCFG model that incorporates representations from LLMs trained on text alone. On four benchmarks for multimodal grammar induction, LC-PCFG outperforms several prior state-of-the-art multimodal approaches. Furthermore, adding visual inputs to LC-PCFG does not improve grammar induction. These experiments show that for grammar induction, the benefits of multimodal inputs can be achieved by more textual data. Our results challenge the notion that multimodal inputs are necessary for grammar induction.

Based on the result that LC-PCFG performs as well as methods trained on multimodal inputs, we speculate that representations from LLMs provide information that is redundant with information provided by multimodal inputs. Indeed, some work has shown that multimodal inputs improve grammar induction by providing signals of noun concreteness (Kojima et al., 2020). Other work has shown that LLMs acquire some knowledge of word concreteness (Ramakrishnan and Deniz, 2021). Thus, large amounts of textual training data may provide signals of word concreteness that obviate multimodal inputs for grammar induction.

7 Broader Impacts Statement

Our experiments show that a text-only baseline can outperform computationally intensive multimodal approaches for grammar induction. These results emphasize the promise of less computationally demanding methods, and we we hope they encourage the community to re-think the necessity of expensive multimodal approaches for certain tasks.

8 Limitations

Our results show that a strong LLM-based text-only baseline outperforms current state-of-the-art multi-modal grammar induction methods, and that adding visual features to this baseline does not further improve grammar induction. It is possible that future work will find better methods of combining visual features with LLMs, and that these methods will outperform any text-only approaches.

9 Acknowledgements

We would like to thank the anonymous reviewers and metareviewers for their constructive comments and suggestions. We also thank the Machine Common Sense project, ONR MURI award number N00014-21-1-2801, and Google's TPU Research Cloud (TRC) for providing cloud TPUs. RC was funded by the DARPA SemaFor program, BAIR Commons, and an NSF Graduate Research Fellowship. CC was funded by an IBM PhD fellowship.

	DiDeMo		YouCook2		MSRVTT	
PCFG Method	C-F1	S-F1	C-F1	S-F1	C-F1	S-F1
Ours	$57.1_{\pm 4.7}$	$60.0_{\pm 5.2}$	$52.4_{\pm 0.1}$	$57.7_{\pm 0.1}$	$56.1_{\pm 3.6}$	$61.2_{\pm 3.7}$
Ours + VideoFeas	$50.1_{\pm 3.7}$	$52.9_{\pm 3.7}$	$53.2_{\pm 1.1}$	$58.0_{\pm 0.8}$	$51.5_{\pm 0.5}$	$55.9_{\pm 1.2}$

Table 10: Incorporating video features (“VideoFeas") into LC-PCFG degrades performance.

References

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. 2017. Localizing moments in video with natural language. In *Proceedings of the IEEE international conference on computer vision*, pages 5803–5812.

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

Steven Cao, Nikita Kitaev, and Dan Klein. 2020. Unsupervised parsing via constituency tests. In *Conference on Empirical Methods in Natural Language Processing*.

Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308.

Glenn Carroll and Eugene Charniak. 1992. *Two experiments on learning probabilistic dependency grammars from corpora*. Department of Computer Science, Univ.

Stanley F Chen, Douglas Beeferman, and Roni Rosenfeld. 1998. Evaluation metrics for language models.

Alexander Clark. 2001. Unsupervised induction of stochastic context-free grammars using distributional clustering. *Proceedings of the 2001 workshop on Computational Natural Language Learning - ConLL '01*.

Dan Deng, Haifeng Liu, Xuelong Li, and Deng Cai. 2018. Pixellink: Detecting scene text via instance segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL*.

Andrew Drozdov, Pat Verga, Mohit Yadav, Mohit Iyyer, and Andrew McCallum. 2019. Unsupervised latent tree induction with deep inside-outside recursive autoencoders. In *NAACL*.

Lila Gleitman. 1990. The structural sources of verb meanings. *Language acquisition*, 1(1):3–55.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer.

Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. 2017. Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 131–135. IEEE.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

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.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Nikita Kitaev, Steven Cao, and Dan Klein. 2018. Multilingual constituency parsing with self-attention and pre-training. In *Annual Meeting of the 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)*, 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.

Noriyuki Kojima, Hadar Averbuch-Elor, Alexander M. Rush, and Yoav Artzi. 2020. What is learned in visually grounded neural syntax acquisition. *ArXiv*, abs/2005.01678.

Karim Lari and Steve J Young. 1990. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer speech & language*, 4(1):35–56.

Jiaxi Li and Wei Lu. 2023. Contextual distortion reveals constituency: Masked language models are implicit parsers.

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. Springer.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. 2016. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer.

Yang Liu, Zhaowen Wang, Hailin Jin, and Ian Wassell. 2018. Synthetically supervised feature learning for scene text recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 435–451.

Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. 2019. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2630–2640.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Steven Pinker. 1984. *Language learnability and language development*. Cambridge University Press.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Kalyan Ramakrishnan and Fatma Deniz. 2021. Non-complementarity of information in word-embedding and brain representations in distinguishing between concrete and abstract words. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 1–11, Online. Association for Computational Linguistics.

ResNetV1.5. 2022. ResNet Version 1.5. `https://pytorch.org/vision/main/models/generated/torchvision.models.resnet152.html`. [Online; accessed 19-January-2023].

Herbert Robbins. 1956. An Empirical Bayes Approach to Statistics. *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, pages 157–163.

Yikang Shen, Zhouhan Lin, Chin-Wei Huang, and Aaron C. Courville. 2018. Neural language modeling by jointly learning syntax and lexicon. *ICLR*.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. 2019. Ordered neurons: Integrating tree structures into recurrent neural networks. *ICLR*.

Haoyue Shi, Jiayuan Mao, Kevin Gimpel, and Karen Livescu. 2019. Visually grounded neural syntax acquisition. *arXiv preprint arXiv:1906.02890*.

Noah A Smith and Jason Eisner. 2005. Guiding unsupervised grammar induction using contrastive estimation. In *Proc. of IJCAI Workshop on Grammatical Inference Applications*, pages 73–82.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 2017. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296.

Songyang Zhang, Linfeng Song, Lifeng Jin, Haitao Mi, Kun Xu, Dong Yu, and Jiebo Luo. 2022a. Learning a grammar inducer by watching millions of instructional youtube videos. In *EMNLP*.

Songyang Zhang, Linfeng Song, Lifeng Jin, Kun Xu, Dong Yu, and Jiebo Luo. 2021. Video-aided unsupervised grammar induction. *NAACL*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022b. Opt: Open pre-trained transformer language models.

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.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464.

Luowei Zhou, Chenliang Xu, and Jason J Corso. 2018. Towards automatic learning of procedures from web instructional videos. In *Thirty-Second AAAI Conference on Artificial Intelligence*.