Phrase-aware Unsupervised Constituency Parsing

Xiaotao Gu[1] **Yikang Shen**[2,3] **Jiaming Shen**[4] **Jingbo Shang**[5] **Jiawei Han**[1]
[1]University of Illinois at Urbana-Champaign [5]University of California San Diego
[2]Mila / Université de Montréal [3]Pattern Recognition Center, WeChat AI, Tencent Inc
[4]Google Research
[1]{xiaotao2, hanj}@illinois.edu [5]jshang@ucsd.edu
[2,3]yikang.shn@gmail.com [4]jmshen@google.com

Abstract

Recent studies have achieved inspiring success in unsupervised grammar induction using masked language modeling (MLM) as the proxy task. Despite their high accuracy in identifying low-level structures, prior arts tend to struggle in capturing high-level structures like clauses, since the MLM task usually only requires information from local context. In this work, we revisit LM-based constituency parsing from a phrase-centered perspective. Inspired by the natural reading process of human readers, we propose to regularize the parser with phrases extracted by an unsupervised phrase tagger to help the LM model quickly manage low-level structures. For a better understanding of high-level structures, we propose a phrase-guided masking strategy for LM to emphasize more on reconstructing non-phrase words. We show that the initial phrase regularization serves as an effective bootstrap, and phrase-guided masking improves the identification of high-level structures. Experiments on the public benchmark with two different backbone models demonstrate the effectiveness and generality of our method.

1 Introduction

The hierarchical structure of natural language plays a key role in accurate language understanding, but can be unfortunately overlooked when text is treated as a plain sequence. To this end, considerable efforts have been made in integrating structural inductive bias into neural language models (LM) (Shen et al., 2018b; Wang et al., 2019; Shen et al., 2020). Despite different implementations, the general idea is to first apply a parsing module to induce the soft grammar tree of the input text, and then incorporate the induced tree into an encoding model (*e.g.*, Transformer (Vaswani et al., 2017)). The model is optimized in an unsupervised manner with masked language modeling (MLM) (Devlin et al., 2019) as a common proxy task.

These models have shown inspiring success in inducing meaningful parsing trees without human annotation, but still face two challenging problems. Firstly, the parsing module is randomly initialized at the beginning of the training process. Suboptimal initial parsing accuracy can lead to problematic structural constraints in the encoder model, and further influence the training process and final performance (Gimpel and Smith, 2012). Secondly, the token-level language modeling task encourages the model to focus on local structures, since the reconstruction of a masked word mainly relies on its local context. As a result, the learned model achieves high accuracy in local constituents, like noun phrases (NP), but significantly worse accuracy in high-level, long-distance structures, such as subordinate clauses (SBAR) and prepositional phrases (PP). On the PTB dataset, the most recent structured language model (Shen et al., 2020) still falls behind neural probabilistic context-free grammar models (*e.g.*, Kim et al. (2019b)) by over 4% in average SBAR and PP recall.

In this work, we revisit the LM-based unsupervised parsing models by providing a *phrase-centered perspective*. We model the reading process of a sentence in a stylized pipeline: when we try to parse a sentence, instead of handling each individual *word*, we first recognize the obvious *phrases*, for instance, names, concepts, slogans, etc. Some phrases are known beforehand, while some are learned from the current context. We then treat each phrase as a complete unit, and only need to figure out the high-level structures that connect these phrases. Following this intuition, we mimic the reading process with a three-stage learning framework. In the first stage, we identify the multigram phrases with the help of an unsupervised phrase tagging model. The extracted phrase set guides the parsing module to quickly manage the pattern of short constituents at the early training stage. The “warm-up” process does not require any external

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 6406 - 6415
May 22-27, 2022

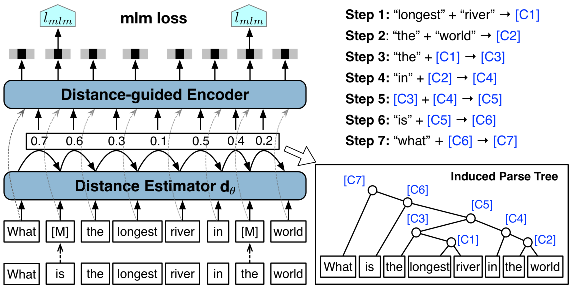

l_{mlm}
mlm loss
l_{mlm}
Step 1: "longest" + "river" → [C1]
Step 2: "the" + "world" → [C2]
Step 3: "the" + [C1] → [C3]
Step 4: "in" + [C2] → [C4]
Step 5: [C3] + [C4] → [C5]
Step 6: "is" + [C5] → [C6]
Step 7: "what" + [C6] → [C7]
Distance-guided Encoder
0.7 0.6 0.3 0.1 0.5 0.4 0.2
Distance Estimator d_θ
What [M] the longest river in [M] world
What is the longest river in the world
Induced Parse Tree
[C7]
[C6]
[C5]
[C3]
[C4]
[C1]
[C2]
What is the longest river in the world

Figure 1: Illustration of LM-based unsupervised constituency parsing. The parse tree is induced from a distance sequence generated by the distance estimator d_θ, which is jointly optimized with a distance-guided encoder from the masked language modeling task.

resource, and effectively improves and stabilizes the initial parsing accuracy. In the second stage, the model is optimized through the original MLM task. After this stage, the model is good at capturing local structures, as stated above. In the third stage, to push the model out of its comfort zone and force it to learn about high-level structures, we apply a simple and effective phrase-guided masked language modeling task. Specifically, we extract short phrases in the training sentences as the local constituents identified by the model, which are relatively "easy cases" for the model. We then sample a part of the phrases, and exclude them from the MLM task, so we are basically downsampling intra-phrase words in the reconstruction task, and emphasizing non-phrase words that connect phrases. The proposed method is general and can be applied to arbitrary LM-based parsers in a plug-and-play manner.

Contributions. The major contributions of this paper are summarized as follows: (1) We point out the major challenges faced by LM-based unsupervised constituency parsing, and revisit the problem with a phrase-centered perspective; (2) We propose a novel framework with phrase-regularized warm-up and phrase-guided mask language modeling, that can be applied to general LM-based parsers for improvement; (3) Experiments on the public benchmark with two different base models demonstrate the effectiveness of our method. Code and data will be published for further research study.

2 Preliminary

In this section, we present our problem formulation and briefly review the general framework of LM-based unsupervised constituency parsing, as illustrated in Figure 1.

Parsing as Distance Estimation. Constituency parsing aims to assign an undirected constituency tree to the input sentence, which illustrates how different parts are hierarchically combined in the sentence (Jurafsky, 2000). To enable end-to-end model learning, following prior works (Wang et al., 2019; Shen et al., 2020), the discrete parsing tree is represented as a *distance sequence* $d_\theta(\mathbf{s}) = \{d_1, d_2, ..., d_{n-1}\}$, where d_i is the distance score between adjacent words w_i and w_{i+1}, parameterized by model θ. Given the distance sequence, the tree structure can be induced in a greedy manner: starting from each single token as a leaf constituent, we recursively merge two constituents with the minimum distance score into a large constituent. The tree structure is hence uniquely determined by the relative order of the distance sequence. Figure 1 shows a concrete example of the parse tree induction process from an estimated distance sequence. Our goal is to learn a high-quality distance estimator d_θ from unlabeled text corpus that induces accurate parsing trees.

Distance-guided Model Learning. For model learning, the generated distance sequence is injected into an encoding model (*e.g.*, Transformer) as structural bias to control information exchange between words. Intuitively, two adjacent words with smaller distance score are more likely to belong to the same constituent, and will exchange more information to each other. The distance estimator d_θ is jointly optimized with the distance-guided encoder from the masked language modeling (MLM) task as a proxy. Formally, given a masking rate μ and a sentence $\mathbf{s} = \{w_1, w_2, ..., w_n\}$, a mask sequence is sampled from uniform Bernoulli sampling, where m_i is a binary variable with $p(m_i = 1) = \mu$. We then get the masked sentence $\hat{\mathbf{s}} = \{\hat{w}_1, ..., \hat{w}_n\}$ by replacing w_i with a mask token where $m_i = 1$. The MLM loss is computed as:

$$\ell_{mlm}(\mathbf{s}) = \frac{\sum_{w_i \in X_{mask}} \log p(w_i|\hat{\mathbf{s}})}{|X_{mask}|},$$

where X_{mask} is the set of masked tokens. The encoding model is trained to minimize ℓ_{mlm} based on the distance-constrained information aggregation. We will introduce more details about the distance-aware encoders in Section 4.

3 Framework Overview

In this work, we recognize and examine two major challenges of LM-based grammar induction:

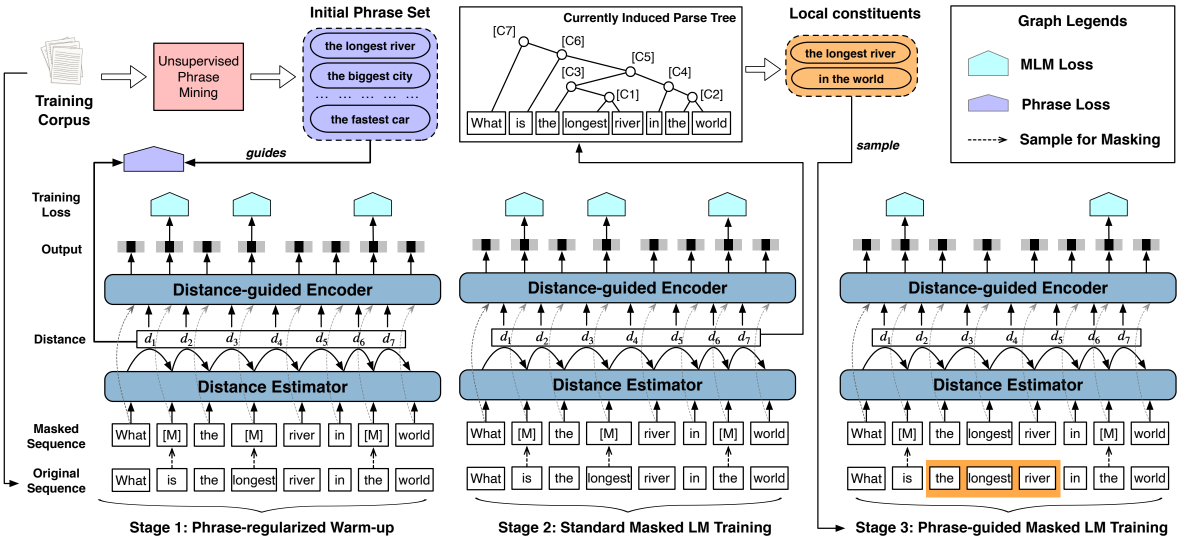


Figure 2: An overview of the proposed framework. Given the training corpus, the training process consists of three stages. **Stage 1**: phrase-regularized warm-up using the initial phrase set extracted by an off-the-shelf unsupervised phrase mining module (Section 5); **Stage 2**: standard masked language model learning; **Stage 3**: extract a new phrase set with the local constituents identified by the model itself, and apply phrase-guided masked language model learning (Section 6).

(1) the randomly initialized distance estimator can yield a suboptimal information exchange network in the encoder in the cold start phase, which may further lead to suboptimal parsing accuracy due to error accumulation. (2) the token reconstruction task mainly relies on the aggregation of local information, thus can hardly guide the model to manage high-level structures across long distances.

To tackle the challenges, we revisit LM-based unsupervised constituency parsing from a *phrase-centered perspective*. We propose a three-stage training framework, as shown in Figure 2. In the first stage, we extract an initial phrase set using an off-the-shelf unsupervised phrase tagger. The extracted phrases serve as effective guidance to help warm up the distance estimator to boost its initial accuracy in the cold start phase. The model then gradually gets rid of the help from the initial phrase set and learns about local structures from the original MLM task in the second stage. In the third stage, we try to push the model out of its comfort zone by moving the focus from local structures to high-level structures. We extract a new phrase set from the local constituents identified by the model itself, which consists of "easy cases" for the model. We then downsample the intra-phrase words for the reconstruction task, and emphasize more on the relatively harder reconstruction of non-phrase words, which connect local constituents into high-level structures. In following sections, we first introduce the base encoding models we experiment with, and then present more details of the proposed framework.

4 Distance-guided Encoders

Our method can be applied to any encoder with a distance estimator and distance-constrained information aggregation. In this work, we examine our method on two recently developed models, TreeTransformer (Wang et al., 2019) and StructFormer (Shen et al., 2020), as our base models. Both models extend the original Transformer encoder (Vaswani et al., 2017) by adding a structure-aware attention term. Specifically, the original Transformer computes the attention matrix A as

$$A = \mathrm{softmax}(\frac{QK^\top}{\sqrt{d_{head}}}),$$

where $a_{ij} \in A$ is the attention score between word w_i and word w_j, Q is the query matrix, K is the key matrix, and d_{head} is the attention head size. The extended attention score in a structure-constrained encoder is written as $a'_{ij} = q_{ij} \cdot a_{ij}$, where q_{ij} is the structure-based attention score determined by the distance sequence.

The two base encoders differ in their ways to parameterize the distance function d_θ and to define the structure-based attention score q_{ij}.

TreeTransformer parameterizes the distance sequence with an additional attention module. The structure-based attention score q_{ij} represents the probability that two words belong to the same constituent, and is defined as

$$q_{ij} = \prod_{k=i}^{j-1}(1 - d_k).$$

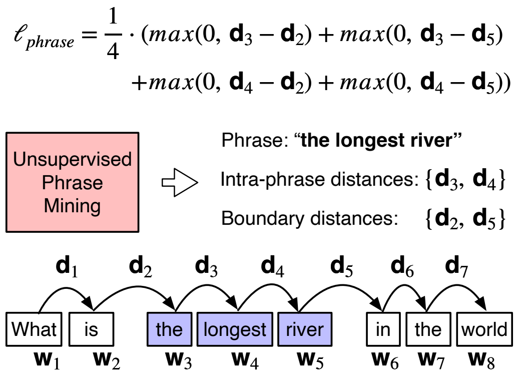
Intuitively, words within a closer distance have more information exchange in TreeTransformer.

Structformer parameterizes the distance sequence with a Convolutional Neural Network. StructFormer uses a more complicated structure constraint: *each constituent has a head word, and information can only be exchanged between the head word and remaining child words in the constituent.* The structure-based attention score q_{ij} stands for the probability that w_i and w_j can exchange information, which means w_i is the head word of any constituent containing w_j, or vice versa. q_{ij} is jointly determined by the distance sequence and a syntacic height sequence. Ideally, the height of each child word in a constituent should not exceed the boundary distances. More details can be found in the original paper (Shen et al., 2020).

To summarize, the distance estimator d_θ determines the attention matrix in the encoder. Through the MLM task, the model learns to optimize d_θ for more effective information aggregation. We then induce the parse tree from the distance sequence generated by d_θ in the parsing process. In following sections, we introduce details about the proposed phrase-regularized warm-up and phrase-guided masked language modeling, which jointly help train a better d_θ.

5 Phrase-regularized Warm-up

Given a target sentence, we first extract spans that are likely to be phrases. By definition, we seek word sequences that consistently occur "consecutively in the text, forming a complete semantic unit in certain contexts" (Finch, 2016). The extracted phrases are used as additional guidance for the distance estimator at the very beginning of the training process. Specifically, we encourage the distance estimator to assign smaller intra-phrase distances than phrase boundary distances to draw a clear gap on the phrase boundaries. Figure 3 shows a concrete example of intra-phrase and phrase boundary distances. Here we introduce more details about the unsupervised phrase extraction process and phrase regularization for warm-up.



$$\ell_{phrase} = \frac{1}{4} \cdot (max(0, \mathbf{d}_3 - \mathbf{d}_2) + max(0, \mathbf{d}_3 - \mathbf{d}_5) + max(0, \mathbf{d}_4 - \mathbf{d}_2) + max(0, \mathbf{d}_4 - \mathbf{d}_5))$$

Unsupervised Phrase Mining
Phrase: "the longest river"
Intra-phrase distances: {d3, d4}
Boundary distances: {d2, d5}
d1 d2 d3 d4 d5 d6 d7
What is the longest river in the world
w1 w2 w3 w4 w5 w6 w7 w8

Figure 3: An example of phrase-regularized warm-up. Given the example sentence with the tagged initial phrase "the longest river", we try to encourage the average intra-phrase distance to be smaller than the average phrase boundary distance through a margin loss.

Phrase Extraction. Without introducing any exogenous resource, we apply the core phrase mining module of the UCPhrase model (Gu et al., 2021), which does not require any complicated model training. Specifically, within each document $\mathcal{D}$, its core phrase $\mathcal{P}_\mathcal{D}$ is defined as the set of max frequent n-grams in $\mathcal{D}$. For each phrase $\mathbf{w}_{i:j} = \{w_i, ..., w_j\} \in \mathcal{P}_\mathcal{D}$, "frequent" means it has to occur in the document for at least τ times. "max" means there does not exist any "super phrase" $\mathbf{w}' \supseteq \mathbf{w}_{i:j}$ in the same document. Such document-level max frequent n-grams are shown to have reasonably high quality and preserve contextual completeness. Uninformative sequences are filtered by a corpus-oriented stopword list generated by TF-IDF ranking. The extracted phrase set serves as effective regularization for the randomly initialized parsing model in early training steps. Note that the phrase extraction module can be replaced by any phrase tagger. Here we show that even phrases extracted by this simple heuristic tagger can bring clear improvement.

Phrase Regularization. Given the target sentence $\mathbf{s} = \{w_1, w_2, ..., w_n\}$ and its initial phrase set $\mathcal{P}_\mathbf{s}$, we encourage the parser to generate smaller distance scores between intra-phrase words than the distance scores on the phrase boundaries. Formally, we compute the phrase distance loss for each phrase $\mathbf{w}_{i:j} = \{w_i, ..., w_j\} \in \mathcal{P}_\mathbf{s}$ as the average margin loss between intra-phrase distance scores and phrase boundary distance scores:

$$\ell_{phrase}(\mathbf{w}_{i:j}) = \frac{1}{|\mathbf{w}_{i:j}|}\sum_{k=i}^{j-1}\frac{\max(0, d_k - d_{i-1}) + \max(0, d_k - d_j)}{2}.$$

The phrase distance loss for the entire sentence is

$$\ell_{phrase}(\mathbf{s}) = \frac{1}{|\mathcal{P}_\mathbf{s}|} \sum_{\mathbf{w}_{i:j} \in \mathcal{P}_\mathbf{s}} \ell_{phrase}(\mathbf{w}_{i:j}).$$

For StructFormer, we replace the intra-phrase distances into the intra-phrase heights to satisfy its structure constraint as introduced in Section 4.

The overall loss function at training step t is formed as:

$$\ell(\mathbf{s}) = \ell_{mlm}(\mathbf{s}) + \lambda_t \cdot \ell_{phrase}(\mathbf{s}),$$

which is basically the original masked language modeling loss ℓ_{mlm} regularized by the phrase distance loss ℓ_{phrase} with coefficient λ_t. For smooth transition, we apply a step-wise linear coefficient decay. At training step t, we have $\lambda_t = \lambda_0 \cdot (1 - t/T_1)$, so that we apply full regularization at the very beginning, and then gradually remove the regularization until the model learns completely from the MLM task. In experiments, we set T_1 to the number of steps in one training epoch by default.

6 Phrase-guided Masked Language Modeling

The masked language modeling task mainly relies on the aggregation of local context information around the masked word. For instance, in the example sentence presented in Figure 2, the prediction of "*longest*" mainly depends on its neighbor "*river*". Hence, the parser can quickly manage the structure of short phrases as they are closely related to the optimization proxy. High-level long constituents, however, can hardly be captured in this process. From this perspective, the sentence parsing task can then be divided into two parts: parsing the structures of short phrases, and capturing high-level long structures that connect short phrases. The former can be learned from the intra-phrase word reconstruction task, and the latter depends on the modeling of other non-phrase words.

Following this intuition, we propose simple and effective phrase-guided masked language modeling to emphasize the reconstruction of words outside of local constituents. Specifically, we parse the training sentences with the learned model, and treat all local constituents (*e.g.*, with fewer than 4 tokens) from the generated parsing trees. Given a sentence with tagged local phrases, we first apply uniform Bernoulli sampling on the phrases with probability μ_p. The sampled phrases are excluded from the MLM task: words inside of the sampled phrases will not be masked. All rest words are sampled for masking with the original masking rate μ. Formally, given a sentence $\mathbf{s}$ with the tagged phrase set $\mathcal{P}_\mathbf{s}$, the probability of word w_i being masked in the MLM task is computed as:

$$P(m_i = 1) = \begin{cases} (1-\mu_p)\cdot\mu, & w_i \in \cup\mathcal{P}_\mathbf{s} \\ \mu, & otherwise. \end{cases}$$

By doing so, we try to push the model out of its comfort zone of local structure learning, and encourage it to focus more on how the local constituents are connected.

Discussion. Another natural idea to achieve similar intuition is to apply phrase-level reconstruction through whole-phrase masking. Namely, we mask the entire phrase so that the model cannot make prediction merely based on information aggregated through local structures, but can only rely on cross-phrase structures to gather information. We test this intuition in two ways: (1) replace each token in the phrase with a mask token, and apply standard MLM; (2) replace the entire phrase with one mask token, and apply autoregressive phrase reconstruction with a decoder similar to Raffel et al. (2020). Interestingly, results from both implementations show that whole-phrase masking can hurt the accuracy of unsupervised parsing. A possible reason is that reconstructing the entire masked phrase relies on deep semantic knowledge rather than just syntactic structures. We list this finding here and leave it as a potential research problem.

7 Experiments

Dataset and Evaluation. Following prior studies (Shen et al., 2018b; Wang et al., 2019; Shen et al., 2020), we train all models on the plain text of the PTB corpus (Mikolov et al., 2010) and evaluate them on the WSJ test set (Taylor et al., 2003), in which punctuations are removed.

We follow the standard evaluation for unsupervised parsing: given a predicted parsing tree, we fetch all of its subtrees (nested constituents), and compare with those from the gold tree to compute the F1 score. We also report recall scores of the typed constituents in gold trees, including noun (NP), verb (VP), prepositional (PP), adjective (ADJ), adverb (ADV) phrases and subordinate clauses (SBAR). The precision score for each type is not available in the unsupervised setting since the predicted constituents do not have types.

Methods	F1 (%)
PRPN (Shen et al., 2018a)	37.4
ON-LSTM (Shen et al., 2018b)	47.7
URNNG (Kim et al., 2019c)	52.4
C-PCFG (Kim et al., 2019b)	55.2
Neural L-PCFGs (Zhu et al., 2020)	55.3
TreeTransformer (Wang et al., 2019)	47.9
+ PMLM	48.7
+ PRW	49.0
+ PRW + PMLM	49.3
StructFormer (Shen et al., 2020)	54.0
+ PMLM	54.1
+ PRW	55.3
+ PRW + PMLM	55.7

Table 1: Unlabeled F1 score (%) for unsupervised constituency parsing on WSJ test set.

Method	NP	VP	ADJ	ADV	SBA	PP
PRPN	59.2	46.7	44.3	32.8	50.0	57.2
ON-LSTM	64.5	41.0	38.1	31.6	52.5	54.4
C-PCFG	74.7	41.7	40.4	52.5	56.1	68.8
TreeTransformer	63.7	37.1	32.3	56.8	37.0	49.7
+ PMLM	63.5	37.9	31.7	56.8	38.0	50.4
+ PRW	64.2	36.3	27.9	53.8	36.2	53.0
+ PRW + PMLM	64.2	37.2	29.6	53.7	35.9	53.3
StructFormer	73.7	43.2	53.4	70.5	51.8	64.5
+ PMLM	73.6	43.7	53.4	69.3	51.9	64.6
+ PRW	74.0	44.9	52.9	69.9	52.7	69.4
+ PRW + PMLM	74.2	45.1	53.2	69.3	53.9	70.1

Table 2: Recall scores (%) of typed gold constituents.

Compared Models. Our baseline methods include three major types of unsupervised parsing method. PRPN (Shen et al., 2018a), ON-LSTM (Shen et al., 2018b) and URNNG (Kim et al., 2019c) are recurrent neural network based methods. They are trained by recurrent language modeling loss, where the model is asked to predict the next token given the previous context. C-PCFG (Kim et al., 2019b) and Neural L-PCFGs (Zhu et al., 2020) are neural network augmented methods based on the traditional probabilistic context-free grammar framework, where a set of weighted linguistic rules are learned for tree generation. TreeTransformer (Wang et al., 2019) and StructFormer (Shen et al., 2020) are the backbone models we apply in our study, as introduced in Section 4. For our method, we report performances of three variants based on each base model: the performance with phrase-regularized warm-up (**+PRW**), the performance with the phrase-guided masked language modeling (**+PMLM**), and the performance with both (**+PRW+PMLM**).

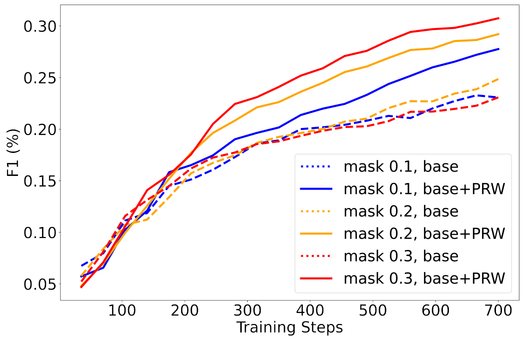


Figure 4: Illustration of how the F1 score grows with more training steps in the first epoch. We present the curves of the original TreeTransformer (*base*, dashed lines) and the curves with phrase-regularized warm-up (*base+PRW*, solid lines) under different masking rates.

Reproduction Details. We use the published StructFormer and TreeTransformer implementations with their default hyperparameters and optimizers as our backbone models. The learning rate is controlled with a linear scheduler for both models, which starts from the original learning rate, and applies a linear learning rate decay until it reaches 0.0 at the last training step. The initial coefficient λ_0 for PRW is set to 0.02 for both models. The phrase masking rate μ_p for PMLM is set to 0.9. The total number of training steps is fixed, and PMLM is included after 80% of training steps. Training and evaluation are conducted on NVIDIA RTX A6000 GPUs. We report average results from four random seeds $(1, 11, 111, 1111)$. Results from both backbone models are reproduced in the same machine as variants with our methods for fair comparison. Results from other baseline models are taken from Shen et al. (2020).

7.1 Performance Comparison

Table 1 shows average F1 scores for the compared methods on the WSJ test set. Both PRW and PMLM bring improvements in the F1 score. Specifically, PRW increases the F1 score by $+1.1\%$ and $+1.3\%$ on TreeTransformer and StructFormer respectively; PMLM increases the F1 score by $+0.8\%$ and $+0.1\%$ respectively; When applied together, PRW and PMLM bring improvement on F1 score by 1.4% and 1.7% respectively. Compared with other parsing models, the enhanced models have very competitive performances. The proposed method helps StructFormer achieve at least comparable F1 score with the state-of-the-art model based on neural linguistic rule learning (C-PCFG).

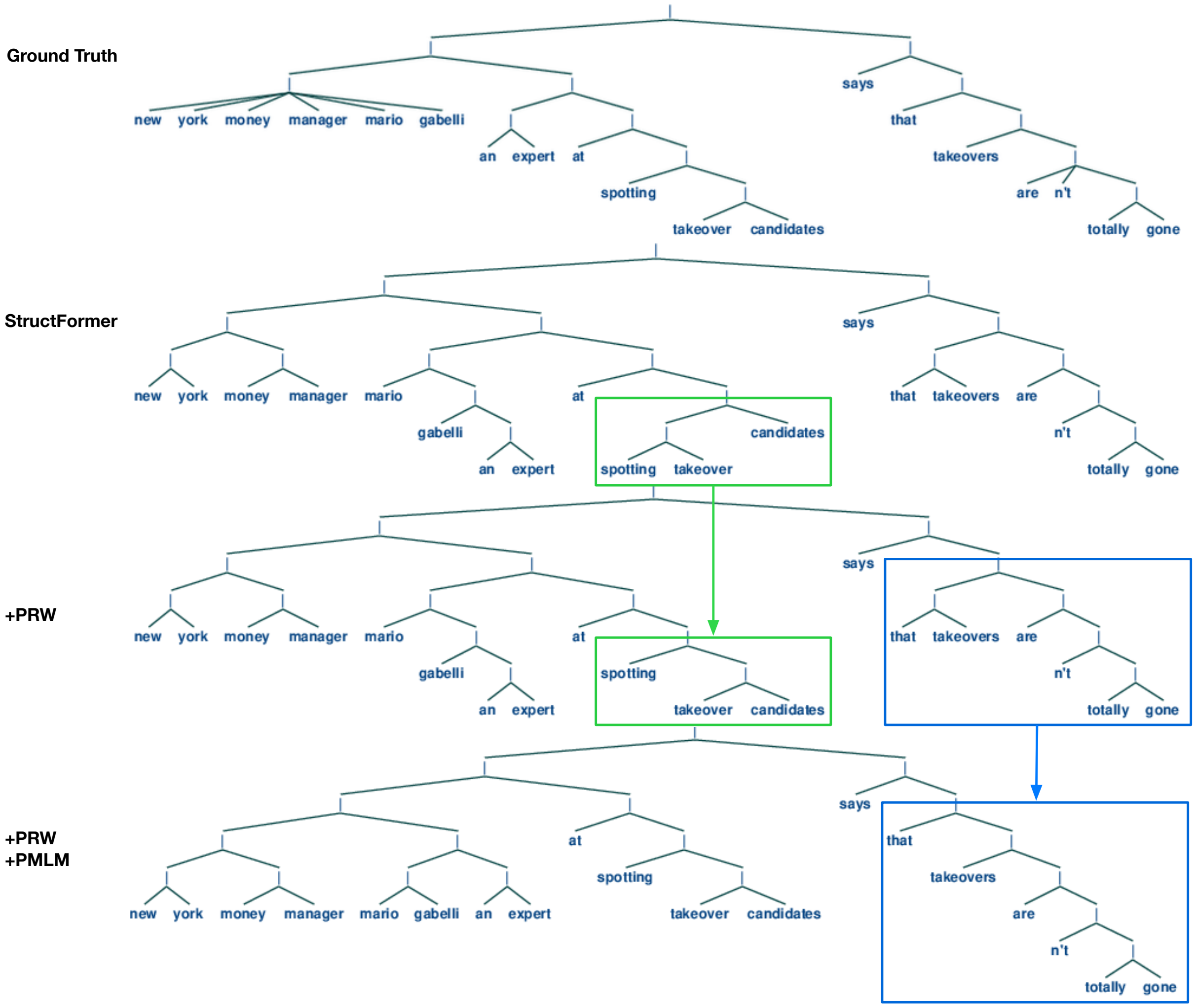

Ground Truth
StructFormer
+PRW
+PRW
+PMLM
new york money manager mario gabelli an expert at spotting takeover candidates says that takeovers are n't totally gone

Figure 5: Comparison between the parsing trees generated by different models on the same input sentence.

Table 2 provides a more in-depth view of the performance change of each type of constituents. Consistent with our intuition, PRW improves the recall of local constituents like NP, and PMLM improves the recall of compositional constituents like VP, SBA and PP. To our surprise, PRW also brings strong improvement in PP, which means the better accuracy in local structure parsing may have a positive impact on high-level structures as well. StructFormer achieves state-of-the-art PP recall with the help of PRW and PMLM.

7.2 How does phrase-regularized warm-up help initialization?

PRM brings strong performance gain, and we are curious about whether the strength of such enhancement, if any, starts from the initial training steps as our design, and how the strength changes with different masking rates. Intuitively, a larger masking rate may make the initial parsing task even harder, since there is less information available. Figure 4 shows the F1 curves of the base TreeTransformer model and the enhanced variant with PRW under different masking rates. We observe that, PRW always brings significant improvement in the initial parsing performance. Different masking rates do not bring very clear differences in the initial performance of the base model. However, the strength of enhancement from PRW becomes more significant as the masking rate gets higher, which verifies our intuition, that the guidance from the initial phrase set may be more valuable with less information available to the initial parser.

7.3 Case Study

To better understand the effectiveness of PRW and PMLM, we conduct case study of the generated parsing trees, as shown in Figure 5. Consider the subtree in the green square. The real noun phrase in the ground truth is “*takeover candidates*”, while

StructFormer mistakenly merges "*spotting*" and "*takeover*" first. The model with PRW identifies the correct noun phrase. The improved initialization with phrase regularization does enhance the parser in its ability to identify short phrases.

The subtree in the blue square shows an example of high-level constituent structure, where "*takeovers aren't totally gone*" forms a clause together with "*that*". StructFormer merges "*that*" with "*takeovers*" and breaks the clause. The original MLM task mainly focuses on local structures, and may prioritize potential local constituents ("*that takeovers*" can form a noun phrase from a local view). PRW cannot fix this issue, but PMLM helps make the right decision. This verifies our intuition, that PMLM encourages the model to learn about the structure of non-phrase words, and to capture better high-level structures.

Limitations. Note that in Figure 5, all models cannot resolve the structure ambiguity between "*Mario Gabelli an expert*" and "*an expert at ...*". It indicates that the current unsupervised methods may have little understanding of semantic and commonsense knowledge. Both structures make sense to the model. Weakly-supervised, or knowledge-enhanced learning may alleviate the problem.

8 Related Work

The study of unsupervised constituency parsing can be traced back to 50 years ago (Booth, 1969; Salomaa, 1969). We highlight some recent progresses that are closely related to our work:

1) Adding syntactic inductive bias into modern neural network models. ON-LSTM (Shen et al., 2018b) allows hidden neurons to learn long-term or short-term information by a novel gating mechanism and activation function. In URNNG (Kim et al., 2019c), amortized variational inference was applied between a recurrent neural network grammar (RNNG) (Dyer et al., 2016) decoder and a tree structure inference network, which encourages the decoder to generate reasonable tree structures. TreeTransformer (Wang et al., 2019) adds extra locality constraints to the Transformer encoder's self-attention to encourage the attention heads to follow a tree structure such that each token can only attend on nearby neighbors in lower layers and gradually extend the attention field to further tokens when climbing to higher layers. StructFormer (Shen et al., 2020) propose a joint dependency and constituency parser, then uses the dependency adjacency matrix to constraint the self-attention heads in transformer models.

2) Using neural network to parameterize linguistic models. The compound PCFG (Kim et al., 2019b) achieves grammar induction by maximizing the marginal likelihood of the sentences which are generated by a probabilistic context-free grammar (PCFG). Neural L-PCFG (Zhu et al., 2020) demonstrated that PCFG can benefit from modeling lexical dependencies. NBL-PCFG (Yang et al., 2021) took a step further by directly modeling bilexical dependencies and reducing both learning and representation complexities of LPCFGs. DIORA (Drozdov et al., 2019) proposed using inside-outside dynamic programming to compose latent representations from all possible binary trees. The representations of inside and outside passes from the same sentences are optimized to be close to each other.

3) Extracting syntactic structure from pretrained language models. Kim et al. (2019a) extract trees from pretrained transformers. Using the model's representations for each word in the sentence, they score fenceposts (positions between words) by computing distance between the two adjacent words. They parse by recursively splitting the tree at the fencepost with the largest distance.

4) Leveraging statistic features to identify constituents. Cao et al. (2020) use constituency tests, that specify a set of transformations and use an unsupervised neural acceptability model to make grammaticality decisions. Clark (2001) proposed to identify constituents based on their span statistics, e.g. mutual information between left and right contexts of the span.

9 Conclusion

In this work, we study the role of phrases in language model-based unsupervised constituency parsing. We propose a phrase-centered framework with novel phrase-regularized warm-up and phrase-aware masked language modeling. Experiments with two different base models demonstrate the effectiveness of the proposed methods. Comprehensive case study is conducted for straightforward understanding of the advantages of our model. Although this work mainly focuses on the task of unsupervised parsing, the presented idea and observation can be valuable in more general context. We plan to follow this line of work and further incorporate our method in long-range structured language model learning in the future.

Acknowledgments

Research was supported in part by US DARPA KAIROS Program No. FA8750-19-2-1004, SocialSim Program No. W911NF-17-C-0099, and INCAS Program No. HR001121C0165, National Science Foundation IIS-19-56151, IIS-17-41317, and IIS 17-04532, IBM-Illinois Discovery Accelerator Institute, and the Molecule Maker Lab Institute: An AI Research Institutes program supported by NSF under Award No. 2019897. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and do not necessarily represent the views, either expressed or implied, of DARPA or the U.S. Government. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

Taylor L Booth. 1969. Probabilistic representation of formal languages. In *10th annual symposium on switching and Automata Theory (swat 1969)*, pages 74–81. IEEE.

Steven Cao, Nikita Kitaev, and Dan Klein. 2020. Unsupervised parsing via constituency tests. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4798–4808.

Alexander Clark. 2001. Unsupervised induction of stochastic context-free grammars using distributional clustering. In *Proceedings of the ACL 2001 Workshop on Computational Natural Language Learning (ConLL)*.

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

Andrew Drozdov, Patrick Verga, Mohit Yadav, Mohit Iyyer, and Andrew McCallum. 2019. Unsupervised latent tree induction with deep inside-outside recursive auto-encoders. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1129–1141.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A Smith. 2016. Recurrent neural network grammars. In *Proceedings of NAACL-HLT*, pages 199–209.

Geoffrey Finch. 2016. *Linguistic terms and concepts*. Macmillan International Higher Education.

Kevin Gimpel and Noah A Smith. 2012. Concavity and initialization for unsupervised dependency parsing. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 577–581.

Xiaotao Gu, Zihan Wang, Zhenyu Bi, Yu Meng, Liyuan Liu, Jiawei Han, and Jingbo Shang. 2021. Ucphrase: Unsupervised context-aware quality phrase tagging. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*.

Dan Jurafsky. 2000. *Speech & language processing*. Pearson Education India.

Taeuk Kim, Jihun Choi, Daniel Edmiston, and Sanggoo Lee. 2019a. Are pre-trained language models aware of phrases? simple but strong baselines for grammar induction. In *International Conference on Learning Representations*.

Yoon Kim, Chris Dyer, and Alexander M Rush. 2019b. Compound probabilistic context-free grammars for grammar induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2369–2385.

Yoon Kim, Alexander M Rush, Lei Yu, Adhiguna Kuncoro, Chris Dyer, and Gábor Melis. 2019c. Unsupervised recurrent neural network grammars. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1105–1117.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*, volume 2, pages 1045–1048. Makuhari.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Arto Salomaa. 1969. Probabilistic and weighted grammars. *Information and Control*, 15(6):529–544.

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

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. 2018b. Ordered neurons: Integrating tree structures into recurrent neural networks. In *International Conference on Learning Representations*.

Yikang Shen, Yi Tay, Che Zheng, Dara Bahri, Donald Metzler, and Aaron Courville. 2020. Structformer: Joint unsupervised induction of dependency and constituency structure from masked language modeling. *arXiv preprint arXiv:2012.00857*.

Ann Taylor, Mitchell Marcus, and Beatrice Santorini. 2003. The penn treebank: an overview. *Treebanks*, pages 5–22.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Yaushian Wang, Hung-Yi Lee, and Yun-Nung Chen. 2019. Tree transformer: Integrating tree structures into self-attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1061–1070.

Songlin Yang, Yanpeng Zhao, and Kewei Tu. 2021. Neural bi-lexicalized pcfg induction. *arXiv preprint arXiv:2105.15021*.

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