Dynamic Programming in Rank Space

Scaling Structured Inference with Low-Rank HMMs and PCFGs

Songlin Yang, Wei Liu, Kewei Tu

June 9, 2022

SIST, ShanghaiTech University

Structured models

We focus on structured models associated with latent discrete structures.

- Discrete structrues are ubiquitous in NLP.
- Representative models: **Hidden Markov Models** (HMMs, chain structure), **Probabilistic Context-Free Grammars** (PCFGs, tree structure).
  - Combinatorially large number of possible structures.
  - Polynomial time dynamic programming algorithms exist for efficient inference.

Structured models

- We aim to **scale** the hidden state size.
  - **Over-parameterization** has been shown beneficial in unsupervised learning of latent-variable models[1].
  - Superior performance in **HMM language modeling**[2] and **PCFG unsupervised parsing**[3].
- **Bottlenecked** by the inference time and space complexities.
  - Quadratic/cubic in the hidden state size for forward/inside algorithm, respectively.

---

[1] Buhai et al., "Empirical Study of the Benefits of Overparameterization in Learning Latent Variable Models".

[2] Chiu and Rush, "Scaling Hidden Markov Language Models".

[3] Yang, Zhao, and Tu, "PCFGs Can Do Better: Inducing Probabilistic Context-Free Grammars with Many Symbols".

Method overview

- Perform tensor rank decomposition (aka. CPD) to decrease complexities.
- Leverage factor graph grammars (FGGs) for intuitive presentation and generalization beyond HMMs and PCFGs.
- Change the order of variable elimination to further decrease complexities in the low-rank large-state setting.

FGGs

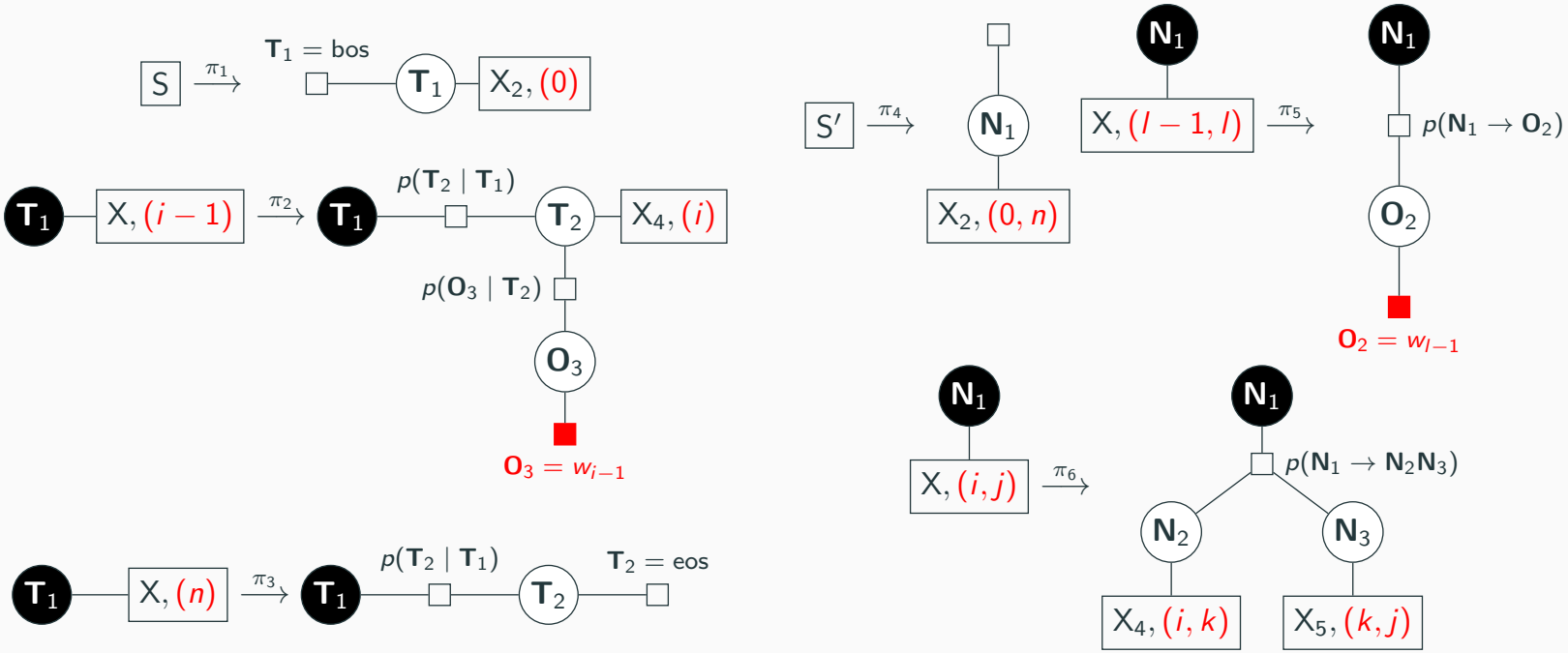


Figure 1: FGG representations of HMMs (left) and PCFGs (right)

Inference with FGGs

- Adaption of **variable elimination** to deal with **hypergraph** structures, similar to hypergraph-based parsing.
- **Bottlenecked** by the size of factors.
- **Key idea:** decomposition of large factors into several smaller factors will **decrease** the inference complexity!

Tensor rank decomposition

We can represent a factor $F(e)$ as an order-k tensor in $\mathbb{R}^{N_1 \times \cdots \times N_k}$. Apply CPD on $F(e)$:

$$F(e) = \sum_{q=1}^{r} \mathbf{w}_{e_1}^q \otimes \mathbf{w}_{e_2}^q \otimes \cdots \otimes \mathbf{w}_{e_k}^q$$

Let $\mathbf{W}_{e_j} = [\mathbf{w}_{e_j}^1, \cdots, \mathbf{w}_{e_j}^r]^T \in \mathbb{R}^{r \times N_j}$, we can depict CPD as follows, where $F(e_i)$ corresponds to $\mathbf{W}_{e_i}$:

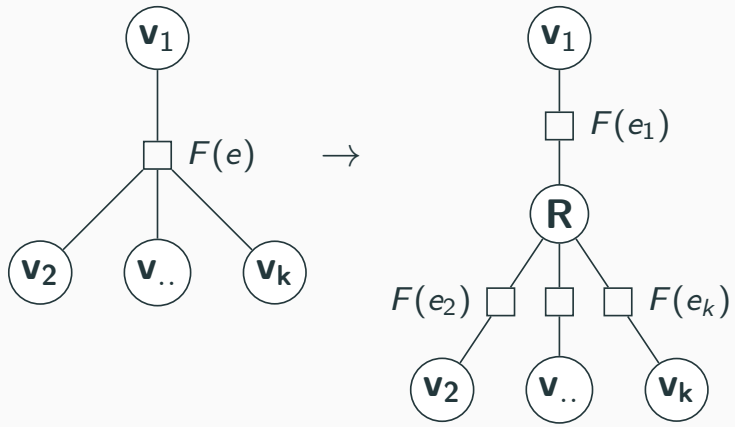
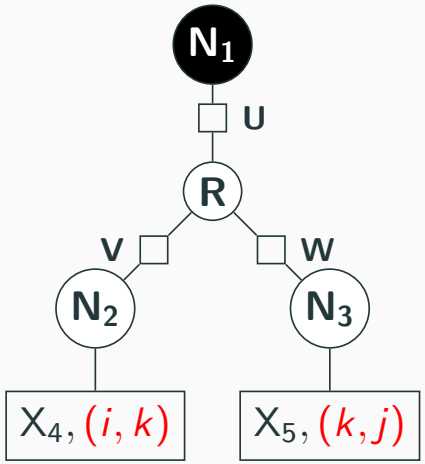


Figure 2: Using CPD to decompose a factor can be seen as adding a new node.

Unifying previous low-rank PCFG models

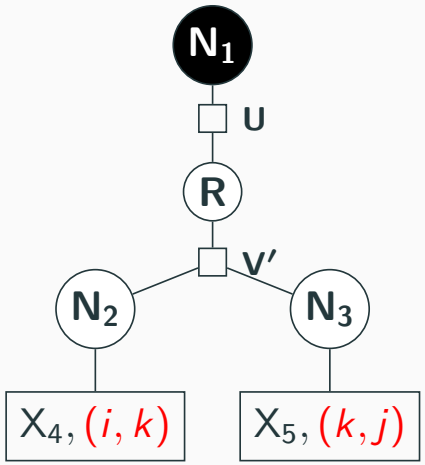


(a)

Accelerated inside computation:

$$\alpha_{i,j} = \sum_{k=i+1}^{j-1} \mathbf{U}^T \left((\mathbf{V}\alpha_{i,k}) \odot (\mathbf{W}\alpha_{k,j}) \right)$$

$$= \mathbf{U}^T \sum_{k=i+1}^{j-1} \left((\mathbf{V}\alpha_{i,k}) \odot (\mathbf{W}\alpha_{k,j}) \right)$$



(b)

$$\alpha_{i,j} = \sum_{k=i+1}^{j-1} \mathbf{U}^T \left(\mathbf{V}' \cdot \alpha_{k,j} \cdot \alpha_{i,k} \right)$$

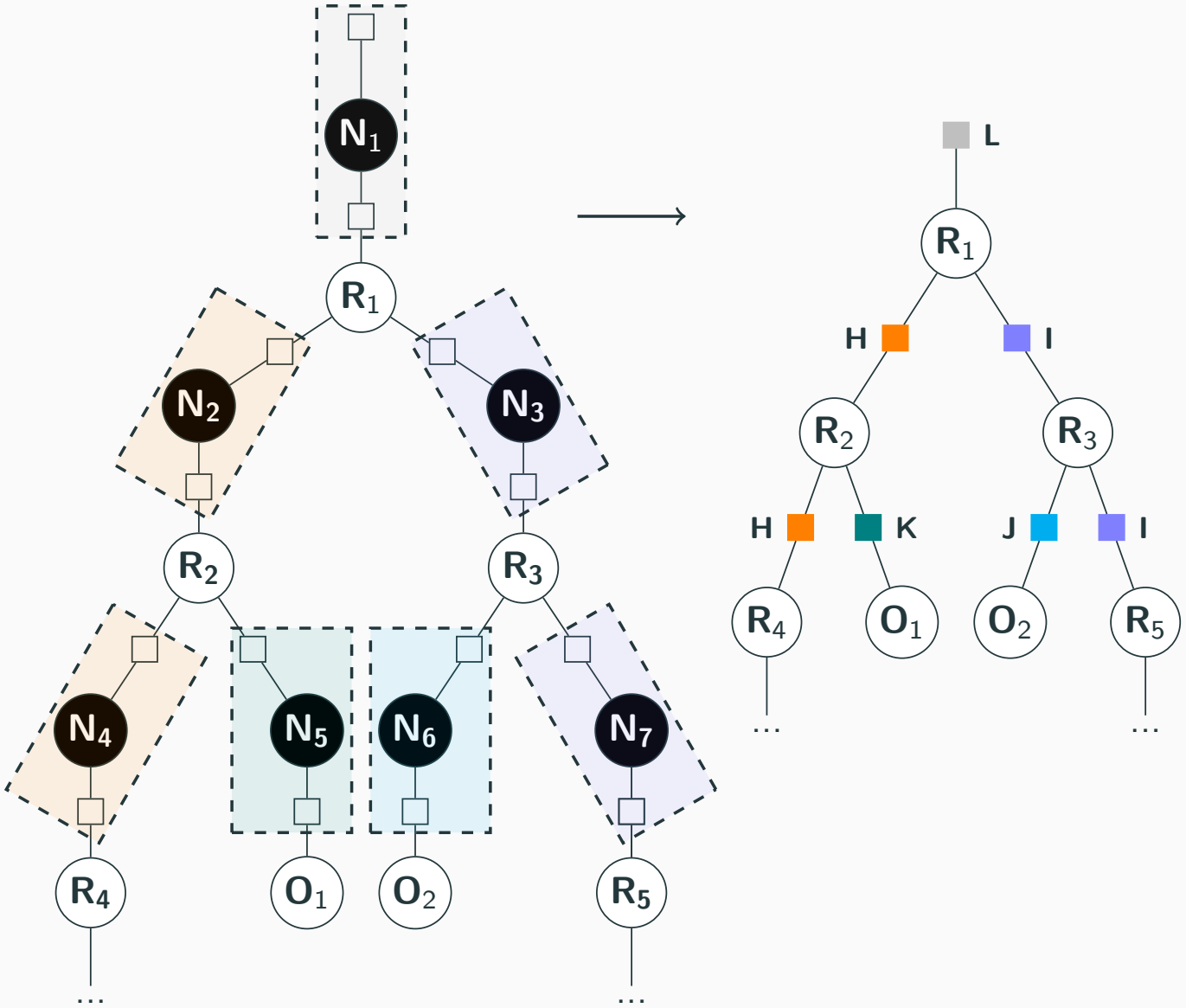
$$= \mathbf{U}^T \sum_{k=i+1}^{j-1} \left(\mathbf{V}' \cdot \alpha_{k,j} \cdot \alpha_{i,k} \right)$$

Figure 3: (a): TD-PCFG (Cohen et al, 2013). (b): LPCFG (Chiu et al, 2021).

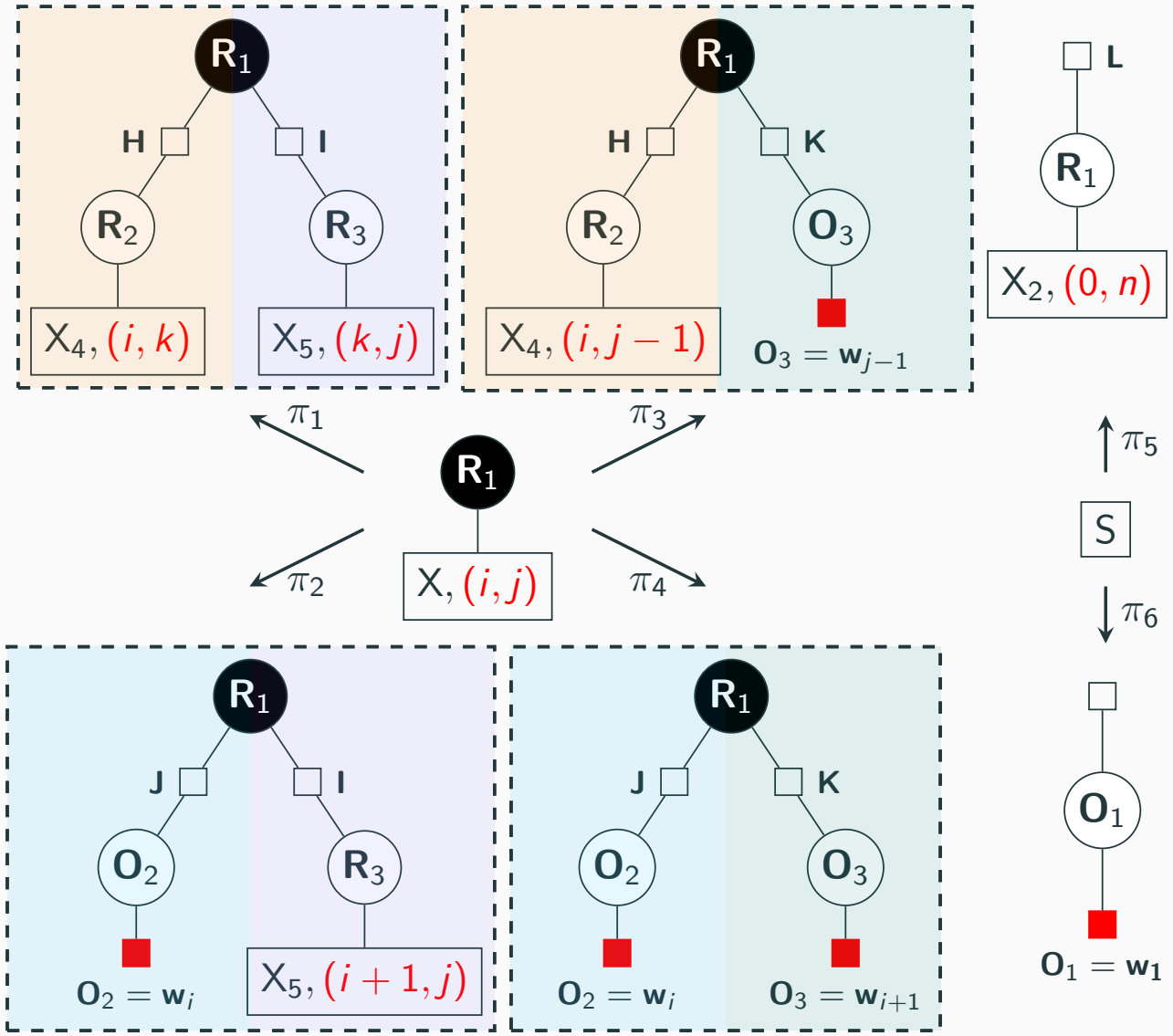
Rank-space inference

- Focus on **B-FGGs** defined analogously to B-graph, i.e., there is only one external node in each fragment.
- Merge factors within a single fragment into a large factor, then apply CPD on it, introducing rank nodes.
- Find repeated substructures that take **rank nodes** as external nodes. **Marginalize all state nodes** to derive new rules defined **in the rank space**.
- Design inference algorithms based on derived new rules.

Rank-space inside algorithm



Rank-space inside algorithm

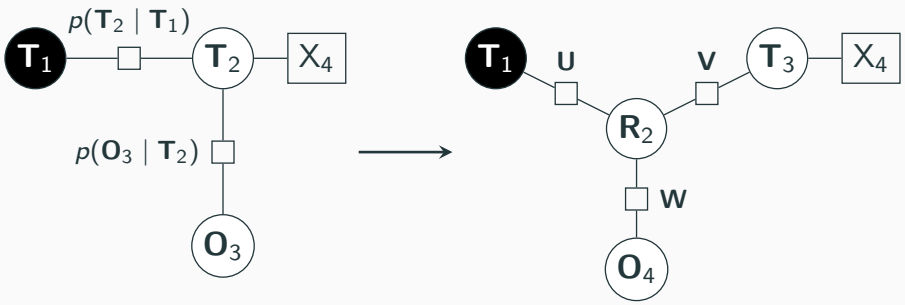


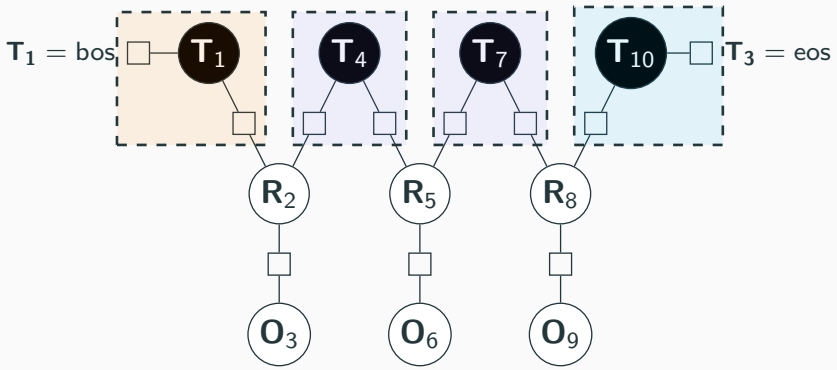
Rank-space inside algorithm

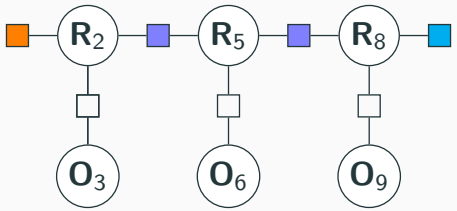
$$\boldsymbol{\alpha}_{i,j} = \overbrace{\sum_{i+1<k<j-1} (\mathbf{H}\boldsymbol{\alpha}_{i,k} \odot \mathbf{I}\boldsymbol{\alpha}_{k,j})}^{\text{from } \pi_1} + \underbrace{\mathbf{J}_{:,w_i} \odot \mathbf{I}\boldsymbol{\alpha}_{i+1,j}}_{\text{from } \pi_2} + \underbrace{\mathbf{H}\boldsymbol{\alpha}_{i,j-1} \odot \mathbf{K}_{:,w_{j-1}}}_{\text{from } \pi_3}$$

- $\mathbf{H}, \mathbf{I}, \mathbf{J}, \mathbf{K}, \mathbf{L}$ are computed only *once* and then reused multiple times during inference.
- We can further optimize the complexity by using **unfold-refold transformation!** (see paper for more details).
- Inference complexity: $O(n^3 r + n^2 r^2)$. Smaller than $O(n^3 r + n^2 rm)$ of TD-PCFG when $r < m$.

Rank-space forward algorithm







Neural parameterization

- Use neural networks to produce probabilities.
- Prior works show that neural parameterization **benefits learning** and **unsupervised induction of syntactic structures**.

$$\mathbf{s} = \frac{\exp(\mathbf{u}_S^T h_1(\mathbf{w}_A)}{\sum_{A' \in \mathcal{N}} \exp(\mathbf{u}_S^T h_1(\mathbf{w}_{A'}))}$$

$$\mathbf{E} = \frac{\exp(\mathbf{u}_E^T h_2(\mathbf{w}_t)}{\sum_{E' \in \Sigma} \exp(\mathbf{u}_{E'}^T h_2(\mathbf{w}_t))}$$

$$\mathbf{U} = \frac{\exp(\mathbf{u}_H^T f_1(\mathbf{w}_n)}{\sum_{n' \in \mathcal{N}} \exp(\mathbf{u}_H^T f_1(\mathbf{w}_{n'}))}$$

$$\mathbf{V} = \frac{\exp(\mathbf{u}_H^T f_2(\mathbf{w}_l)}{\sum_{H' \in \mathcal{H}} \exp(\mathbf{u}_{H'}^T f_2(\mathbf{w}_l))}$$

$$\mathbf{W} = \frac{\exp(\mathbf{u}_H^T f_3(\mathbf{w}_l)}{\sum_{H' \in \mathcal{H}} \exp(\mathbf{u}_{H'}^T f_3(\mathbf{w}_l))}$$

$$h_i(\mathbf{x}) = g_{i,1}(g_{i,2}(\tilde{\mathbf{W}}_i \mathbf{x}))$$

$$g_{i,j}(\mathbf{y}) = ReLU(\tilde{\mathbf{V}}_{i,j} ReLU(\tilde{\mathbf{U}}_{i,j} \mathbf{y})) + \mathbf{y}$$

Experiments on PCFG induction

Model	**S-F1**
N-PCFG (Kim et al, 2019)	50.8
C-PCFG (Kim et al, 2019)	55.2
NL-PCFG (Zhu et al, 2020)	55.3
TN-PCFG (Yang et al, 2021)	57.7
NBL-PCFG (Yang et al, 2021)	60.4
Ours with 9000 PTs and 4500 NTs	**64.1**
For reference	
Constituency (Cao et al, 2020)	62.8
S-DIORA (Drozdov et al, 2020)	57.6
StructFormer (Shen et al, 2021)	54.0
DIORA+span constraint (Xu et al, 2021)	61.2

Table 1: Results on PTB. S-F1: sentence-level F1. PTs: preterminals. NTs: nonterminals.

Experiments on PCFG induction

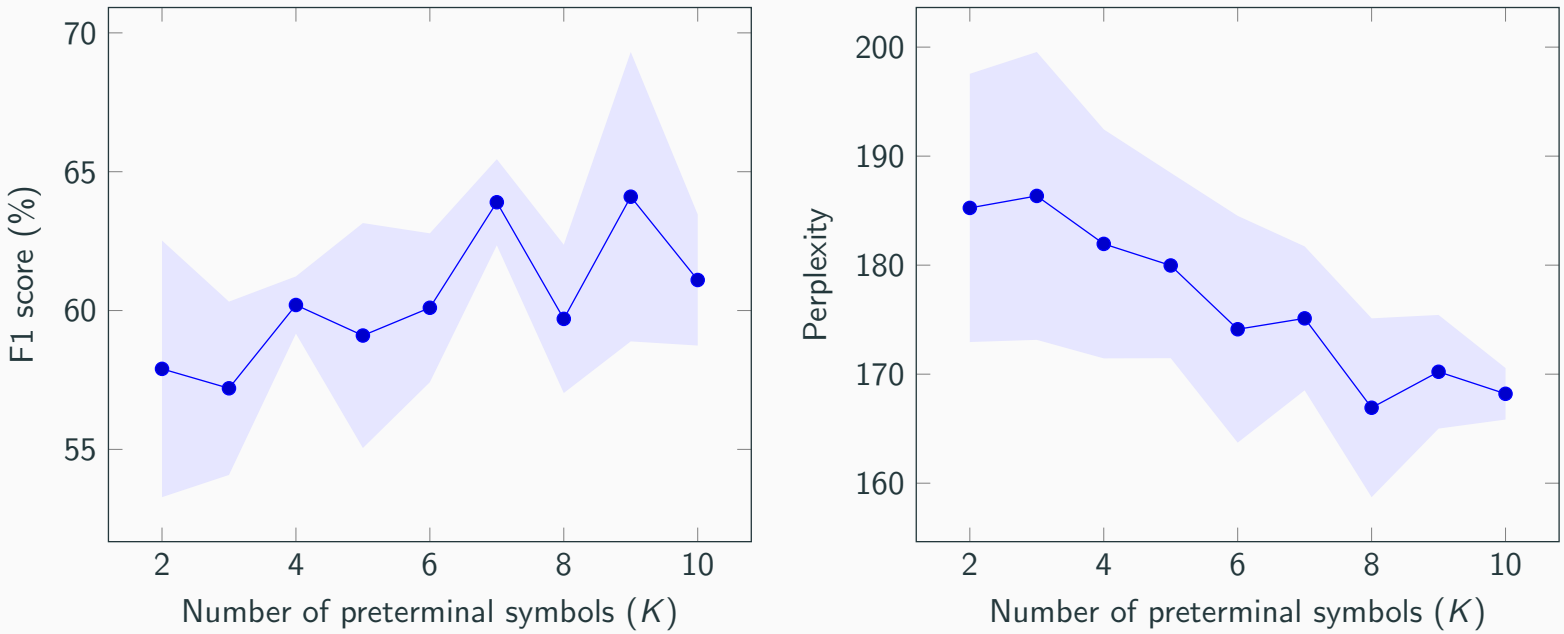


Figure 4: The change of F1 scores and perplexities with the change of number of perterminal symbols.

Experiments on HMM language modeling

Model	Val	Test
VL-HMM (2^{15} states, Brown)	125.0	116.0
VL-HMM (2^{14} states, Brown)†	136	-
VL-HMM (2^{14} states, Uniform)†	146	-
LHMM (2^{14} states)	141.4	131.8
Ours (2^{14} states)	135.6	127.0
Ours (2^{15} states)	137.0	126.4
For reference		
HMM+RNN (Buys et al, 2018)	142.3	-
AWD-LSTM (Merity et al, 2018)	60.0	57.3

Table 2: Resulting perplexity on PTB validate set and test set. VL-HMM: (Chiu and Rush, 2020). LHMM: (Chiu et al, 2021). † denotes results reported by ablation study of (Chiu and Rush, 2020).

Experiments on HMM language modeling

#States	Val	Test
2^{12}	149.8	139.1
2^{13}	143.8	133.4
2^{14}	149.5	137.4
2^{15}	141.1	131.1

Table 3: Perplexity with varying numbers of states. Following (Chiu et al, 2021), we fix the rank to 2048 for faster ablation studies.

Thank you for listening!