IMPROVING UNSUPERVISED CONSTITUENCY PARSING VIA MAXIMIZING SEMANTIC INFORMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Unsupervised constituency parsers organize phrases within a sentence into a tree-shaped syntactic constituent structure that reflects the organization of sentence semantics. However, the traditional objective of maximizing sentence log-likelihood (LL) does not explicitly account for the close relationship between the constituent structure and the semantics, resulting in a weak correlation between LL values and parsing accuracy. In this paper, we introduce a novel objective for training unsupervised parsers: maximizing the information between constituent structures and sentence semantics (SemInfo). We introduce a bag-of-substrings model to represent the semantics and apply the probability-weighted information metric to estimate the SemInfo. Additionally, we develop a Tree Conditional Random Field (TreeCRF)-based model to apply the SemInfo maximization objective to Probabilistic Context-Free Grammar (PCFG) induction, the state-of-the-art non-ensemble method for unsupervised constituency parsing. Experiments demonstrate that SemInfo correlates more strongly with parsing accuracy than LL. Our algorithm significantly enhances parsing accuracy by an average of 7.85 points across five PCFG variants and in four languages, achieving state-of-the-art level results in three of the four languages.

1 INTRODUCTION

Unsupervised constituency parsing is a syntactic task of organizing phrases of a sentence into a tree-shaped and unlabelled constituent structure without relying on any linguistic annotations (Klein & Manning, 2002). The constituent structure is a fundamental tool in analyzing sentence semantics (i.e., the meaning) (Carnie, 2007; Steedman, 2000) and can significantly improve performance for downstream Natural Language Processing systems, such as natural language inference (He et al., 2020), machine translation (Xie & Xing, 2017) and semantic role labeling (Chen et al., 2022). Each constituent in the structure corresponds to a meaningful substring in the sentence, which guides us to progressively construct the sentence semantics. Figure 1 illustrates the progressive semantic construction of the sentence "John has been working on a theory until late night". This example demonstrates that *constituent substrings in the sentence carry significant semantic information*, illustrating an alignment between syntax and semantics.

Maximizing sentence log-likelihood has traditionally been the primary training objective for unsupervised constituency parsers (Eisner, 2016; Kim et al., 2019a). However, Log-Likelihood (LL), the objective function, does not explicitly factor in the syntax-semantics alignment. This leads to a poor correlation between the LL value and the parsing accuracy, which we will discuss further in Section 5.3. As pointed out in previous research, training a Probabilistic Context-Free Grammar (PCFG) parser that outperforms trivial baselines with the LL maximization objective is challenging (Carroll & Charniak, 1992; Kim et al., 2019a). Successful training commonly involves altering the LL maximization objective, such as imposing sparsity constraints (Cohen et al., 2008; Johnson et al., 2007) or heuristically estimating the LL value (Spitkovsky et al., 2010). Evidence suggests that the LL function might not provide robust information to distinguish between constituents and non-constituents, rendering LL an insufficient objective function for unsupervised parsing.

In this paper, we propose a novel objective for unsupervised constituency parsing: maximizing the semantic information that constituent structures carry. We propose a novel method for measuring *SemInfo*, the information between constituent structures and sentence semantics. Specifically,

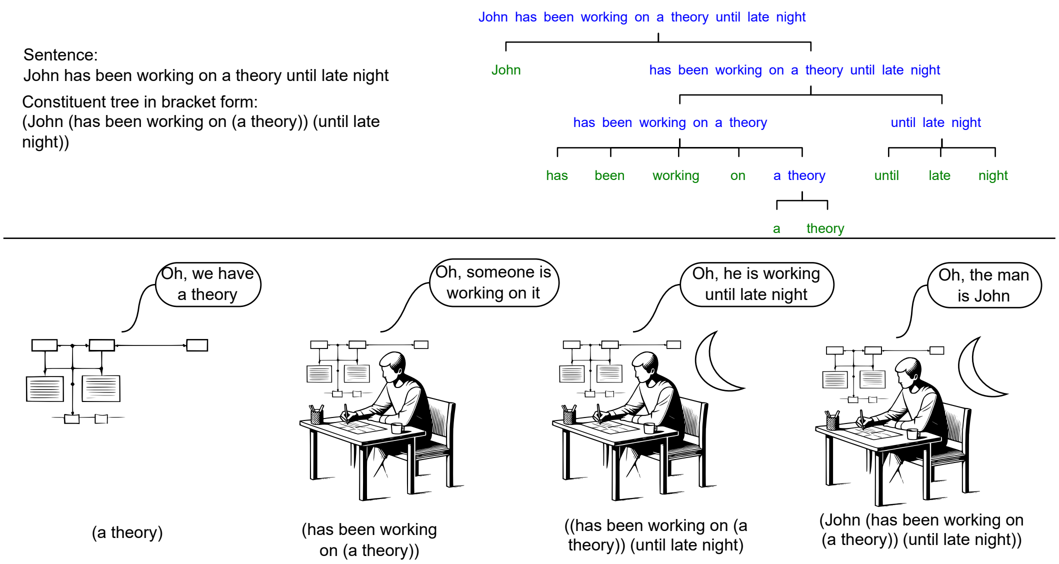

Figure 1: An illustration of the progressive semantics build-up in accordance with the constituent structure. The tree structure in the top-right shows the simplified constituent structure for illustration purposes. Constituent strings are highlighted in blue.

we introduce a bag-of-substrings model to represent the sentence semantics and quantify semantic information encoded in each substring using the probability-weighted information metric developed for bag-of-words models (Aizawa, 2003). We calculate the SemInfo value by aggregating the substring-semantic information associated with the constituent structure. Experiments show a much stronger correlation between SemInfo and the parsing accuracy than the correlation between LL and the parsing accuracy, suggesting that SemInfo is an effective objective function for unsupervised constituency parsing. In addition, we develop a Tree Conditional Random Field (TreeCRF)-based model to apply the SemInfo maximization objective to PCFG induction, the state-of-the-art non-ensemble method for unsupervised constituency parsing. Experiments demonstrate that the SemInfo maximization objective improves the PCFG's parsing accuracy by 7.85 points across five latest PCFG variants and in four languages.

Our main contributions are: (1) Proposing a novel method for estimating SemInfo, the information between constituent structures and sentence semantics. (2) Demonstrating a strong correlation between SemInfo values and parsing accuracy. (3) Developing a TreeCRF model to apply the SemInfo maximization objective to PCFG induction, significantly improving parsing accuracy and achieving state-of-the-art level results as a non-ensemble model.

2 Background

The idea that constituent structures reflect the organization of sentence semantics is central to modern linguistic studies (Steedman, 2000; Pollard & Sag, 1987). A constituent is a substring s in a sentence x that can function independently (Carnie, 2007) and carries individual meaning (Heim & Kratzer, 1998). A collection of constituents forms a tree-shaped structure t, which we can represent as a collection of its constituent strings $t = \{s_1, s_2, ...\}$. For example, the constituent structure in the top right of Figure 1 can be represented as {"a theory", "until late night",...}. Previous research evaluate unsupervised constituency parsers using corpus-level sentence-F1 scores $\mathrm{SF1}^c$ (the average of instance-level sentence-F1 scores $\mathrm{SF1}^i$ across the corpus) (Shen et al., 2017) . The $\mathrm{SF1}^i$ score is computed as the F1 score of two string collections, one representing the predicted structure and the other representing the gold structure.

In this paper, we will utilize the Probability-Weighted Information (PWI) (Aizawa, 2003) developed for Bag-of-Words (BoW) models to measure SemInfo in our bag-of-substrings model. PWI provides an information-theoretic interpretation of the term frequency-inverse document frequency (tf-idf) statistics calculated in BoW models. Let $\mathcal{D}$ denote a document corpus, d_i the i-th document in the corpus, and w_{ij} the j-th word in d_i. The BoW model represents the document d_i as an unordered collection of words occurring in the document (i.e., $d_i = \{w_{i1}, w_{i2}, ...\}$). Tf-idf, as shown in Equation 1, is the product of the term frequency $F(w_{ij}, d_i)$ (i.e. the frequency of w_{ij} occurring in d_i) and the inverse document frequency (i.e. the inverse log-frequency of documents containing w_{ij}). Tf-idf is an important feature in finding keywords in documents (Li et al., 2007) or in efficiently locating documents based on the given keyword (Mishra & Vishwakarma, 2015). PWI interprets the term frequency as the word generation probability and the inverse document frequency as the piecewise word-document information (Equation 2). Intuitively, a word with a high tf-idf value suggests that it carries a significant amount of information about the document (i.e., a keyword of the document). Aizawa (2003) further interpret scaling or smoothing of the term frequency as variations in estimating $P(w_{ij}|d_i)$.

$$\text{tf-idf}(w_{ij}, d_i) = \underbrace{F(w_{ij}, d_i)}_{\text{term frequency}} \times \underbrace{\log \frac{|\mathcal{D}|}{|d' : d' \in \mathcal{D} \wedge w_{ij} \in d'|}}_{\text{inverse document frequency}} \tag{1}$$

$$\approx \underbrace{P(w_{ij}|d_i)}_{\text{word generation probability}} \times \underbrace{\log \frac{P(d_i|w_{ij})}{P(d_i)}}_{\text{piecewise word-document information}} \tag{2}$$

$$= PWI(w_{ij}, d_i)$$

Our method is developed upon the finding of Chen et al. (2024): constituent structures can be predicted by searching frequet substrings among semantically similar paraphrases. We extend their finidngs, interpreting the substring frequency statistic as a term in the substring-semantics information metric and apply the information metric to improve PCFG induction. As we will see in Section 5.2, our method significantly outperforms theirs in three out of the four languages tested.

PCFG induction is currently the state-of-the-art non-ensemble method for training unsupervised constituency parsers (Liu et al., 2023; Yang et al., 2021a). This method trains a binary PCFG parser over a text corpus by maximizing the average LL of the corpus. A PCFG is a generative model defined by a tuple (NT, T, R, S, π), where NT is the set of non-terminal symbols, T is the set of terminal symbols, R is the set of production rules, S is the start symbol, and π is the probability distribution over the rules. The generation process starts with the start symbol S and iteratively applies non-terminal expansion rules ($A \to BC : A, B, C \in NT$) or terminal rewriting rules ($A \to w : A \in NT, w \in T$) until it produces a complete sentence x. We can represent the generation process with a tree-shaped structure t. The PCFG assigns a probability for each distinct way of generating x, defining a distribution $P(x, t)$. The Inside-Outside algorithm (Baker, 1979) provides an efficient solution for computing the total sentence probability $P(x) = \sum_t P(x, t)$. The algorithm constructs a $\beta(s, A)$ table that records the total probability of generating a substring s of x from the non-terminal A. The sentence probability can be calculated as $P(x) = \beta(x, S)$, the probability of x being generated from the start symbol S. The $\beta(x, S)$ quantity is commonly referred to as $Z(X)$ (Eisner, 2016). Besides the total sentence probability, the β table can also be used to calculate the span-posterior probability of s being a constituent (Eisner, 2016) (Equation 3).[1]

$$P(s \text{ is a constituent}|x) = \sum_{A \in NT} \frac{\partial \log Z(x)}{\partial \log \beta(s, A)} \tag{3}$$

Span-based TreeCRF models are widely adopted in constituency parsers (Kim et al., 2019b; Stern et al., 2017) to model the parser distribution $P(t|x)$, the probability of constituent structure t given x. The span-based TreeCRF model determines the probability of t by evaluating whether all substrings involved in the structure are constituents. The TreeCRF model will assign a high score to a substring s in its potential function $\phi(s, x)$ if s is likely a constituent. Subsequently, the TreeCRF model can represent the parser distribution as $P(t|x) \propto \prod_{s \in t} \phi(s, x)$. Using the TreeCRF framework, we can interpret the Tree Minimum Bayesian Risk (TreeMBR) decoding of PCFG (Yang et al., 2021b) as a Viterbi decoding of a TreeCRF model. In this interpretation, the TreeCRF model uses the substring's span-posterior probability as its potential value (i.e., $\phi(s, x) = \exp(P(s \text{ is a constituent}|x))$). Intuitively, the more likely a substring is a constituent, the more likely the TreeCRF model will generate

[1] We explain the derivation in more detail in Section A.1.

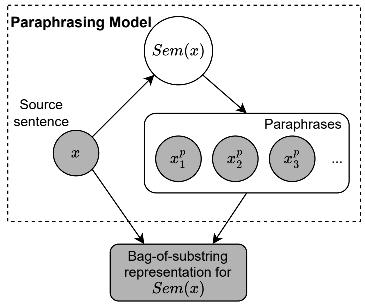

Figure 2: Bag-of-Substrings model derived from a paraphrasing model

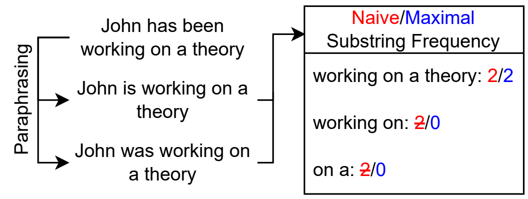

Figure 3: An example for naive substring frequency among paraphrases failing to estimate $P(s|Sem(x))$.

a structure including the substring as a constituent. Consequently, the parser distribution derived from the TreeMBR decoding algorithm is defined as follows.

$$P(t|x) \propto \exp\left(\sum_{s\in t}\sum_{A\in NT}\frac{\partial \log Z(x)}{\partial \log \beta(s, A)}\right) \tag{4}$$

3 SemInfo: A Metric of Information between Constituent Structures and Semantics

In this section, we introduce our estimation method of SemInfo, the information between constituent structures and sentence semantics. We first introduce a bag-of-substrings model, representing the semantics of a sentence by examining whether substrings of the sentence would be *regenerated* during a paraphrasing process. We assume the paraphrasing process is fully capable of generating *natural language paraphrases* (i.e., the paraphrases should both be acceptable as natural language sentences and have similar semantics with the original sentence). We use instruction-following large langauge models (LLMs) for the paraphrasing model, utilizing the their multi-lingual capability to enable multi-lingual parsing in the supported languages. We then apply the PWI (Aizawa, 2003) metric to measure the substring-semantics information. For a given constituent structure, we estimate its SemInfo value by aggregating the substring-semantics information associated with the structure.

Our bag-of-substrings model is based on the paraphrasing model $P(x^p|Sem(x))$ shown in Figure 2. The paraphrasing model takes a source sentence x as input, internally analyzes its semantics $Sem(x)$, and generates a paraphrase x^p. We can repeatedly sample from the process, collecting a paraphrase set $\mathbb{X}^p = \{x_1^p, x_2^p, ...\}$. To estimate the semantic information for substrings of x, we examine how often these substrings are regenerated by the paraphrasing model. Specifically, if a substring s from x is regenerated in any paraphrase x_i^p, we consider s to be generated by $Sem(x)$. We then define a bag-of-substrings representation of $Sem(x)$ by aggregating the regeneration observations across $\mathbb{X}^p$.

We apply the PWI metric to quantify the substring-semantics information $I(s, Sem(x))$ between s and $Sem(x)$, leveraging the similarity between our bag-of-substrings model and the traditional bag-of-words model. To facilitate the PWI calculation (Equation 5), we define two components: $P(s|Sem(x))$, the substring generation probability, and $\log\frac{P(Sem(x)|s)}{P(Sem(x))}$, the piecewise mutual information between s and $Sem(x)$. Following standard practices in bag-of-words models (Aizawa, 2003; Blei et al., 2003), we assume that sentences in the corpus $\mathcal{D}$ are equally likely and each represents distinct semantics. The two assumptions yield an empirical distribution $P(Sem(x)) = \frac{1}{\mathcal{D}}$, where $|\mathcal{D}|$ is the size of the corpus.

$$I(s, Sem(x)) = \frac{1}{|\mathcal{D}|}P(s|Sem(x))\log\frac{P(Sem(x)|s)}{P(Sem(x))} \tag{5}$$

3.1 Defining $P(s|Sem(x))$ and $\log\frac{P(Sem(x)|s)}{P(Sem(x))}$ with Maximal Substring Frequency

Frequency is a simple but effective empirical estimator of distributions. However, naively measuring substring frequency among paraphrases $\mathbb{X}^p$ will yield a misleading estimate of $P(s|Sem(x))$, the probability of s being generated to carry information of $Sem(x)$. The reason is that one substring can be nested in another substring. If a substring s is generated to convey semantic information, we will observe an occurrence of s along with an occurrence of all its substrings. Hence, the naive substring frequency will wrongly count substring occurrences caused by the generation of larger substrings as occurrences caused by $P(s|Sem(x))$. Let's consider the example illustrated in Figure 3. All three substrings in the example have a frequency of 2, yet only the first substring carries significant semantic information. This is because the occurrence of the first substring causes the occurrence of the second and third substrings. The true frequency of the second and third substrings should be 0 instead of 2.

We introduce the notion of maximal substring to counter this problem. Given a source sentence x and a paraphrase x_i^p, the maximal substring between the two is defined in Equation 6. Intuitively, a maximal substring is the largest substring that occurs in both x and x_i^p. Formally, we denote the partial order relationship of string α being a substring in string β by $\alpha \leq \beta$, and denote the set of maximal substrings by $MS(x, x_i^p)$. Using maximal substrings, we can avoid over-counting substring occurrences caused by the generation of larger substrings.

$$MS(x, x_i^p) := \{\alpha : \alpha \leq x \wedge \alpha \leq x_i^p \wedge \forall \alpha'(\alpha < \alpha' \implies \neg\alpha' \leq x \vee \neg\alpha' \leq x_i^p)\} \tag{6}$$

We are now ready to define $P(s|Sem(x))$ using the paraphrasing distribution $P(x^p|Sem(x))$ and the notion of maximal substrings. We define $P(s|Sem(x))$ to be proportional to s's probability of being generated as a maximal substring in paraphrases (Eqaution 7). The probability can then be approximated using the maximal substring frequency $F(s, \mathbb{X}^p)$, as shown in Equation 8.

$$P(s|Sem(x)) \propto \mathop{\mathbb{E}}_{x_i^p \sim P(x^p|Sem(x))} \mathbf{1}(s \in MS(x_i^p, x)) \tag{7}$$

$$\approx \frac{1}{C} F(s, \mathbb{X}^p) \tag{8}$$

Similarly, we define the inverse document frequency for maximal substrings (Equation 9). The inverse document frequency can serve as an estimate of the piecewise substring-semantics information, quantifying how useful a substring is to convey semantic information. A high inverse document frequency implies that we can easily identify the target semantics $Sem(x)$ in a corpus with the maximal substring (i.e., the substring carries high information about $Sem(x)$).

$$\log \frac{P(Sem(x)|s)}{P(Sem(x))} \approx \log \frac{|\mathcal{D}|}{|\{x' : x' \in \mathcal{D} \wedge s \in MS(x, x')\}|} \tag{9}$$

3.2 SemInfo Metric

A constituent structure t can be represented as a set of constituent substrings. We define SemInfo, the information between the structure t and the semantics $Sem(x)$, as the cumulative substring-semantics information associated with the structure (Equation 10). We estimate the substring-semantics information with the maximal substring frequency-inverse document frequency developed in the above sections.

$$I(t, Sem(x)) = \sum_{s \in t} I(s, Sem(x)) \tag{10}$$

$$\propto \sum_{s \in t} \underbrace{F(s, \mathbb{X}^p) \log \frac{|\mathcal{D}|}{|\{x' : x' \in \mathcal{D} \wedge s \in MS(x, x')\}|}}_{\text{maximal substring frequency-inverse document frequency}} \tag{11}$$

4 SemInfo Maximization via TreeCRF Model

We apply the SemInfo maximization objective to PCFG induction models by maximizing Equation 12 and using the pipeline shown in Figure 4. We consider the PCFG parser a one-step reinforcement learning policy $P(t|x)$ and consider the SemInfo value $I(t, Sem(x))$ the reward function. As shown in Figure 4, we first compute the $\log(Z(x))$ by applying the inside-algorithm on the PCFG model. Then, we perform back-propagation, calculating the parser distribution $P(t|x)$ in accordance with Equation 3 and Equation 4. We choose the TreeCRF model because we can efficiently sample from the tree distribution and calculate the entropy. We apply the REINFORCE algorithm with average baseline (Williams, 1992) and maximum entropy regularization (Ziebart et al., 2008)

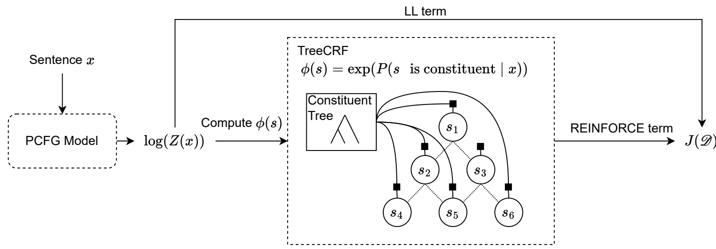

Figure 4: Pipeline of our SemInfo maximization training

to maximize the expected SemInfo. We also include the traditional LL term $\log Z(x)$ in Equation 12 because we found the inclusion significantly stabilizes the training process. The stabilization might be related to the correlation between LL values and parsing accuracy at the early training stage, as we will discuss in Section 5.3.2.

$$\mathcal{J}(\mathcal{D}) = \mathop{\mathbb{E}}_{x\sim\mathcal{D}}[\log Z(x) + \mathop{\mathbb{E}}_{t\sim P(t|x)}[\log P(t|x)(I(t, Sem(x)) - \mathop{\mathbb{E}}_{t\sim P(t|x)} I(t, Sem(x)) + \beta H(P(t|x)))]] \tag{12}$$

5 Experiment

5.1 Experiment Setup

We evaluate the effect of the SemInfo maximization objective on five latest PCFG variants: Neural-PCFG (NPCFG), Compound-PCFG (CPCFG) (Kim et al., 2019a), TNPCFG (Yang et al., 2021b), Simple-NPCFG (SNPCFG), and Simple-CPCFG (SCPCFG) (Liu et al., 2023).[2] SNPCFG and SCPCFG represent the current state-of-the-art for non-ensemble unsupervised constituency parsing. We use 60 NTs for NPCFG and CPCFG, and 1024 NTs for TNPCFG, SNPCFG, and SCPCFG in our experiment. We conduct the evaluations in three datasets and four languages, namely Penn TreeBank (PTB) (Marcus et al., 1999) for English, Chinese Treebank 5.1 (CTB) (Palmer et al., 2005) for the Chinese, and SPMRL (Seddah et al., 2013) for the German and French. We adopt the standard data split for the PTB dataset, using Sections 02-21 for training, Section 22 for validation, and Section 23 for testing (Kim et al., 2019a). We adopt the official data split for the CTB and SPMRL datasets.

Following Shen et al. (2017), we train the PCFG induction model on raw text and evaluate its parsing performance using the SF1^c scores. When aggregating for the SF1^c score, we include only sentences longer than two words and drop punctuation and trivial spans (i.e., sentence-level spans and spans with only one word). We also use the SF1^i score to evaluate the correlation between the SemInfo value and parsing accuracy.

Following Chen et al. (2024), we use the `gpt-4o-mini-2024-07-18` model as our paraphrasing model and apply normalization to the source sentence and the paraphrases. We use eight semantic-preserving prompts for the paraphrasing model.[3] We apply the snowball stemmer (Bird & Loper, 2004) to normalize the source sentence and its paraphrases before calculating the maximal substring frequency and the inverse document frequency. We apply the log-normalization (Sparck Jones, 1972) to the maximal substring frequency to avoid some high-frequency substrings dominating the SemInfo value. We found that the log-normalization variant performs marginally but consistently better than the unnormalized variant in our preliminary experiment. The application of log-normalization is aligned with the theoretical framework of Aizawa (2003).

5.2 SemInfo Maximization Significantly Improves Parsing Accuracy

Table 1 compares SemInfo-trained PCFGs and LL-trained PCFGs for five PCFG variants and in four languages. For each variant, we independently train three PCFG models on the SemInfo and LL objectives and report the mean and standard deviation of the model's performance. We can observe that the SemInfo-trained PCFGs achieve significantly higher parsing accuracy than the LL-trained PCFGs. The average improvements are 13.09, 6.02, 7.31, and 4.92 SF1^c scores in English, Chinese,

[2] Our implementation is based on the source code released by Yang et al. (2021b) and Liu et al. (2023)
[3] Detailed prompts are listed in Section A.7

	English		Chinese		French		German	
	SemInfo (Ours)	LL	SemInfo	LL	SemInfo	LL	SemInfo	LL
CPCFG	**65.74** $_{\pm 0.81}$	53.75 $_{\pm 0.81}$	50.39 $_{\pm 0.87}$	51.45 $_{\pm 0.49}$	**52.15** $_{\pm 0.75}$	47.50 $_{\pm 0.41}$	**49.80** $_{\pm 0.31}$	45.64 $_{\pm 0.73}$
NPCFG	**64.45** $_{\pm 1.13}$	50.96 $_{\pm 1.82}$	**53.30** $_{\pm 0.42}$	42.12 $_{\pm 3.07}$	**52.36** $_{\pm 0.62}$	47.95 $_{\pm 0.09}$	**50.74** $_{\pm 0.28}$	45.85 $_{\pm 0.63}$
SCPCFG	**67.27** $_{\pm 1.08}$	49.42 $_{\pm 2.42}$	51.76 $_{\pm 0.54}$	46.20 $_{\pm 3.65}$	**52.79** $_{\pm 0.80}$	45.03 $_{\pm 0.42}$	**47.97** $_{\pm 0.76}$	45.50 $_{\pm 0.71}$
SNPCFG	**67.15** $_{\pm 0.62}$	58.19 $_{\pm 1.13}$	**51.55** $_{\pm 0.82}$	43.79 $_{\pm 0.39}$	**55.21** $_{\pm 0.47}$	49.64 $_{\pm 0.91}$	**49.65** $_{\pm 0.29}$	40.51 $_{\pm 1.26}$
TNPCFG	**66.55** $_{\pm 0.96}$	53.37 $_{\pm 4.28}$	51.79 $_{\pm 0.83}$	45.14 $_{\pm 3.05}$	**54.11** $_{\pm 0.66}$	39.97 $_{\pm 4.10}$	**49.26** $_{\pm 0.64}$	44.94 $_{\pm 1.34}$
Average Δ	+13.09		+6.02		+7.31		+4.92	
MaxTreeDecoding	58.28		49.03		52.03		50.82	
GPT4o-mini	36.16		11.82		30.01		33.56	

Table 1: SF1^c scores of five PCFG variants trained with SemInfo and LL. Each cell in the upper section reports the mean SF1^{c}score and the standard deviation across three *identical and independently trained* PCFG models. Average Δ indicates average improvements in the SF1^{c}score when training with SemInfo compared to LL. Improvements that are statistically significant ($p < 0.05$) are highlighted in bold.

French, and German, respectively. Two-tailed t-tests indicate the improvement to be statistically significant (p<0.05) in 17 out of 20 combinations. Two of the three insignificant results are due to the high score variance of the LL-trained PCFGs. The significant improvement demonstrates the benefit of the SemInfo maximization objective in the unsupervised constituency parsing task. The result also confirms the importance of semantic factors in identifying the syntactic constituent structure.

Table 1 also compares the SemInfo trained PCFG with two baseline parsers: the Maximum Tree Decoding (MTD) parser that predicts the constituent structure by finding the structure with maximum SemInfo value, and the GPT4o-mini parser that asks the GPT4o-mini model to directly predict the constituent structure. Among the two baselines, we see that the MTD parser has significantly higher SF1^{c}scores than the GPT4o-mini parser across the four languages. The accuracy gap indicates that SemInfo is discovering non-trivial information about the constituent structure. Comparing the SemInfo-trained PCFG and the MTD parser, we see that all SemInfo-trained PCFG variants outperform the MTD parser in English, Chinese, and French. The accuracy improvement indicates that the constituent information provided by the SemInfo value is noisy, and the grammar learns to mitigate the noises. In German, SemInfo-trained PCFGs perform worse than the MTD parser. One possible reason is that the German validation/testing set has a significantly different word vocabulary compared to the training set, unlike the datasets in the other three languages. The out-of-vocabulary rate in German dataset is 14%, while the rate is 5%, 6%, and 7% in the English, Chinese, and French dataset. This shift in word distribution might be a significant factor in German PCFGs' poor parsing accuracy.

5.3 SemInfo Strongly Correlates with Parsing Accuracy

In this section, we evaluate two aspects of the SemInfo and LL functions: (1) Whether the function can accurately evaluate the model's prediction. (2) Whether the function can approximately rank models in accordance with their performance. We evaluate the two aspects using Spearman correlation (Spearman, 1904), examining whether the objective function can rank good predictions/models higher than the bad ones. We evaluate the prediction ranking capability using a sentence-level correlation and the model ranking capability using a corpus-level correlation.

5.3.1 SemInfo is an accurate metric of parsing accuracy

The sentence-level correlation assesses the objective function's capability to evaluate the model prediction accurately. We independently train eight *identical* PCFG models using the LL maximization objective. Each model is trained with a unique random seed for 30k steps. For every sentence, the eight models produce eight (SF1^i, SemInfo, LL) tuples, which we use to calculate the sentence-level Spearman correlation coefficient.

Figure 5 illustrates the SemInfo-SF1^i correlation and the LL-SF1^i correlation using a random sentence in the English validation set. We observed a strong positive correlation for the (SemInfo, SF1^i) pairs but observed no positive correlation for the (LL, SF1^i) pairs. Table 2 demonstrates the correlation in a more statistically reliable way. We perform mean-aggregation on the sentence-level correlation coefficient using Fisher's Z transformation (Fisher, 1915). Fisher's Z transformation converts the correlation coefficient to a uni-variance space, reducing the negative impact caused by

	SemInfo-$SF1^i$	LL-$SF1^i$	SemInfo-LL
CPCFG	0.6518	0.0223	0.0196
NPCFG	0.6347	-0.0074	-0.0045
SCPCFG	0.6431	-0.0013	0.0505
SNPCFG	0.9289	0.0102	0.0182
TNPCFG	0.6449	0.1077	0.1426

Table 2: Spearman correlation coefficient among (SemInfo, LL, $SF1^i$), and LL over the English validation set. Correlations are aggregated at the corpus-level.

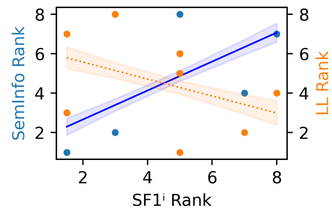

Figure 5: Spearman rank analysis of (SemInfo, LL, $SF1^i$) pairs obtained from eight independently trained NPCFG models. The values are measured on a sentence from the English dataset. Please refer to Figure 8 for more examples.

the skewness in the coefficient's distribution(Silver & Dunlap, 1987). The aggregated coefficients for the SemInfo-$SF1^i$ correlation range from 0.6-0.9, whereas the aggregated coefficients for the LL-$SF1^i$ correlation center around 0. We can consistently observe the correlation coefficient gap in different stages of training, as discussed in Appendix A.2. This sentence-level correlation analysis demonstrates a strong positive correlation between SemInfo and $SF1^i$, while identifying no apparent correlation between LL and $SF1^i$. The high SemInfo-$SF1^i$ coefficient indicates that SemInfo can serve as an accurate metric of parsing accuracy. The gap in correlation coefficients suggests that SemInfo is a better objective function for unsupervised constituency parsing than LL.

5.3.2 SemInfo ranks PCFG models better than LL

The corpus-level correlation evaluates the objective functions' capability to rank models in accordance with their performance. We examine the correlation using model checkpoints collected over different training stages of the above eight PCFG models. Each stage is represented by a window over the amount of training steps. For example, a stage [1k, 10k] contains checkpoints from 1k to 10k steps. These checkpoints produce $SF1^c$ scores, average SemInfo values, and average LL values, which we use to calculate the corpus-level coefficient at that training stage.

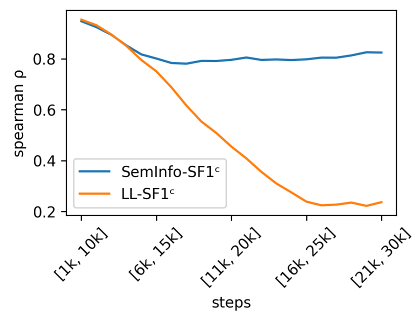

Figure 6: Spearman ρ with $SF1^c$ in different training stages of NPCFG.

Figure 6 illustrates the SemInfo-$SF1^c$ and LL-$SF1^c$ correlation curves for NPCFG.[4] We can observe that LL does have a strong corpus-level correlation with $SF1^c$ at the early stage of training despite having a near-non-existent sentence-level correlation. However, the strength of the corpus-level correlation diminishes quickly, dropping below 0.4 at the late training stage. This result indicates that LL is able to identify a reasonable PCFG model among a set of poorly performing models in the early training stage, but such ability quickly degrades as the training goes on. In comparison, SemInfo maintains a strong correlation across the whole training process, which indicates SemInfo's superior capability in ranking PCFG models by their performance.

5.4 Comparing with State-Of-The-Arts

Table 3 compares three SemInfo-trained PCFG variants with the state-of-the-art non-ensemble methods for unsupervised constituency parsing. The SemInfo-trained PCFGs achieved state-of-the-art level parsing accuracy in English, Chinese, and French, outperforming the second-best algorithm by 1.82, 11.02, and 4.47 $SF1^c$scores, respectively. The SemInfo-trained PCFGs, while using less than half the parameters, perform on par or significantly better than the larger SCPCFG and SNPCFG reported by Liu et al. (2023). The comparison showcases the strong parsing accuracy of the SemInfo-trained PCFGs, confirming the usefulness of semantic information in discovering the constituent structure.

[4]We include the correlation curve for the other four PCFG variants in Appendix A.3.

	English	Chinese	French	German
NPCFG (60NT)	$63.62_{\pm 1.07}$	$\mathbf{53.92}_{\pm 0.48}$	$51.88_{\pm 0.73}$	$47.77_{\pm 0.26}$
SCPCFG (1024NT)	$\mathbf{66.92}_{\pm 0.76}$	$52.26_{\pm 0.41}$	$52.29_{\pm 0.53}$	$45.32_{\pm 0.67}$
SNPCFG (1024NT)	$66.84_{\pm 0.53}$	$52.04_{\pm 0.93}$	$\mathbf{54.37}_{\pm 0.10}$	$47.27_{\pm 0.16}$
Spanoverlap (Chen et al., 2024)	52.9	48.7	48.5	**49.5**
SCPCFG (2048NT) (Liu et al., 2023)	60.6	42.9	49.9	49.1
SNPCFG (4096NT) (Liu et al., 2023)	65.1	39.9	38	46.7
URNNG (Kim et al., 2019b)	40.7	29.1	-	-
NBL-PCFG (Yang et al., 2021a)	60.4	-	-	-
S-DIORA (Xu et al., 2021)	57.6	-	-	-
Constituency Test (Cao et al., 2020)	62.8	-	-	-

Table 3: $\mathrm{SF1}^c$on English, Chinese, French, and German test sets. The top section shows the score for SemInfo-trained PCFGs while the bottom section shows the result from previous work.

6 RELATED WORKS

Parsing with PCFG PCFG induction is a long-established (Klein & Manning, 2002) and state-of-the-art (Liu et al., 2023) approach for non-ensemble unsupervised constituency parsing. Much research has dedicated to improving PCFG induction from the model perspective, such as scaling up the PCFG model (Yang et al., 2021b; Liu et al., 2023), integrating lexical information (Yang et al., 2021a), and allowing PCFG rule probabilities to condition on sentence embeddings through variational inferences (Kim et al., 2019a). Our improvement is from the model optimization perspective and can be combined with the above efforts. Our experiments validate the effectiveness of the SemInfo maximization objective in improving unlexicalized PCFGs. The SemInfo maximization objective is also applicable to lexicalized PCFGs, which we leave to future work.

Parsing with Semantics Zhao & Titov (2020) and Zhang et al. (2021) have sought to improve PCFG induction by learning to identify visual features, maximizing the association between constituent structures and these visual features. If we consider the visual features as semantic representations, their approach is effectively maximizing the semantic information of the constituent structure. In comparison, our method shares the same underlying principle but represents the semantics with textual features. Our method leverages large language models as semantic processors, utilizing their outstanding semantic processing capabilities (Minaee et al., 2024). We believe that combining both textual and visual semantic representations presents a significant research direction for unsupervised parsing tasks.

Improving Parsing with Ensemble Models Ensembling unsupervised parsers (Shayegh et al., 2024) significantly improves accuracy for unsupervised parsing by aggregating predictions from various base parsers. They show that those base parsers predict the constituent structure differently and utilize the difference to obtain a more accurate parsing result. Our method can be combined with the ensemble method for better parsing accuracy. We conduct a parser agreement analysis in Appendix A.5 to show the potential. The agreement analysis shows an agreement score of 80 among our SemInfo-trained PCFG parsers using various paraphrasing models. The agreement score is similar to that of homogeneous parsers reported in Shayegh et al. (2024). The analysis also shows that our parsers have an agreement score of 50 with other base parsers, similar to the score between heterogeneous parsers. The similarity in agreement score suggests that our parsers should be able to serve as a useful component.

7 CONCLUSION

In this paper, we proposed and validated a novel objective for unsupervised constituency parsing: maximizing the information between constituent structures and sentence semantics (SemInfo). We developed a bag-of-substrings model to represent the semantics and applied the probability-weighted information metric to estimate the SemInfo. We applied the SemInfo maximization objective by maximizing the expected SemInfo for a PCFG-based TreeCRF parser. Experiments showed that SemInfo has a strong sentence-level correlation with parsing accuracy and that SemInfo maintains a consistent corpus-level correlation throughout the PCFG training process. The result indicates that SemInfo can serve as an accurate metric of parsing accuracy as well as a reliable training objective for unsupervised parsing. As a result, SemInfo-trained PCFGs significantly outperformed LL-trained PCFGs across four languages, achieving state-of-the-art level performance in three of them. Our findings underscore the effectiveness of leveraging semantic information in unsupervised constituency parsing, paving the way for semantically-informed unsupervised parsing methods.

8 REPRODUCIBILITY

We provide a detailed description of our method in Sections 3 and 4. Implementation details, including data source, model architecture, and hyperparameter settings, are included in Section 5.1. We will release our source code publicly after the review process.

REFERENCES

Akiko Aizawa. An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65, January 2003. ISSN 03064573. doi: 10.1016/S0306-4573(02)00021-3. URL https://linkinghub.elsevier.com/retrieve/pii/S0306457302000213.

J. K. Baker. Trainable grammars for speech recognition. *The Journal of the Acoustical Society of America*, 65(S1):S132–S132, June 1979. ISSN 0001-4966, 1520-8524. doi: 10.1121/1.2017061. URL https://pubs.aip.org/jasa/article/65/S1/S132/739840/Trainable-grammars-for-speech-recognition.

Steven Bird and Edward Loper. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pp. 214–217, Barcelona, Spain, July 2004. URL P04-3031.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):993–1022, March 2003. ISSN 1532-4435.

Steven Cao, Nikita Kitaev, and Dan Klein. Unsupervised parsing via constituency tests. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4798–4808, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.389. URL https://aclanthology.org/2020.emnlp-main.389.

Andrew Carnie. *Syntax: a generative introduction*. Introducing linguistics. Blackwell Pub, Malden, MA, 2nd ed edition, 2007. ISBN 9781405133845.

Glenn Carroll and Eugene Charniak. Two experiments on learning probabilistic dependency grammars from corpora. Technical report, Brown University, USA, 1992.

Junjie Chen, Xiangheng He, and Yusuke Miyao. Modeling syntactic-semantic dependency correlations in semantic role labeling using mixture models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7959–7969, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.548. URL https://aclanthology.org/2022.acl-long.548.

Junjie Chen, Xiangheng He, Danushka Bollegala, and Yusuke Miyao. Unsupervised parsing by searching for frequent word sequences among sentences with equivalent predicate-argument structures. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics ACL 2024*, pp. 3760–3772, Bangkok, Thailand and virtual meeting, August 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.findings-acl.225.

Shay Cohen, Kevin Gimpel, and Noah A Smith. Logistic normal priors for unsupervised probabilistic grammar induction. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou (eds.), *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008. URL https://proceedings.neurips.cc/paper_files/paper/2008/file/f11bec1411101c743f64df596773d0b2-Paper.pdf.

Jason Eisner. Inside-outside and forward-backward algorithms are just backprop (tutorial paper). In Kai-Wei Chang, Ming-Wei Chang, Alexander Rush, and Vivek Srikumar (eds.), *Proceedings of the Workshop on Structured Prediction for NLP*, pp. 1–17, Austin, TX, November 2016. doi: 10.18653/v1/W16-5901. URL W16-5901.

R. A. Fisher. Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population. *Biometrika*, 10(4):507, May 1915. ISSN 00063444. doi: 10.2307/2331838. URL https://www.jstor.org/stable/2331838?origin=crossref.

Qi He, Han Wang, and Yue Zhang. Enhancing generalization in natural language inference by syntax. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4973–4978, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.447. URL https://aclanthology.org/2020.findings-emnlp.447.

Irene Heim and Angelika Kratzer. *Semantics in generative grammar*. Blackwell textbooks in linguistics. Blackwell, Malden, MA, 1998. ISBN 9780631197126.

Mark Johnson, Thomas Griffiths, and Sharon Goldwater. Bayesian inference for PCFGs via Markov chain Monte Carlo. In Candace Sidner, Tanja Schultz, Matthew Stone, and ChengXiang Zhai (eds.), *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pp. 139–146, Rochester, New York, April 2007. URL N07-1018.

Yoon Kim, Chris Dyer, and Alexander Rush. Compound probabilistic context-free grammars for grammar induction. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2369–2385, Florence, Italy, July 2019a. doi: 10.18653/v1/P19-1228. URL P19-1228.

Yoon Kim, Alexander Rush, Lei Yu, Adhiguna Kuncoro, Chris Dyer, and Gábor Melis. Unsupervised recurrent neural network grammars. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *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)*, pp. 1105–1117, Minneapolis, Minnesota, June 2019b. doi: 10.18653/v1/N19-1114. URL N19-1114.

Dan Klein and Christopher D. Manning. A generative constituent-context model for improved grammar induction. In Pierre Isabelle, Eugene Charniak, and Dekang Lin (eds.), *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 128–135, Philadelphia, Pennsylvania, USA, July 2002. doi: 10.3115/1073083.1073106. URL P02-1017.

Juanzi Li, Qi'na Fan, and Kuo Zhang. Keyword extraction based on tf/idf for chinese news document. *Wuhan University Journal of Natural Sciences*, 12(5):917–921, September 2007. ISSN 1007-1202, 1993-4998. doi: 10.1007/s11859-007-0038-4. URL http://link.springer.com/10.1007/s11859-007-0038-4.

Wei Liu, Songlin Yang, Yoon Kim, and Kewei Tu. Simple hardware-efficient PCFGs with independent left and right productions. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 1662–1669, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.113. URL https://aclanthology.org/2023.findings-emnlp.113.

Mitchell P. Marcus, Beatrice Santorini, Mary Ann Marcinkiewicz, and Ann Taylor. Treebank-3, 1999. URL https://catalog.ldc.upenn.edu/LDC99T42.

Shervin Minaee, Tom Mikolov, Narjes Nikzad, Meysam Asgari Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey. *ArXiv*, abs/2402.06196, 2024. URL https://api.semanticscholar.org/CorpusID:267617032.

Apra Mishra and Santosh Vishwakarma. Analysis of tf-idf model and its variant for document retrieval. In *2015 international conference on computational intelligence and communication networks (cicn)*, pp. 772–776. IEEE, 2015.

Martha Palmer, Fu-Dong Chiou, Nianwen Xue, and Tsan-Kuang Lee. Chinese treebank 5.0, January 2005. URL https://catalog.ldc.upenn.edu/LDC2005T01.

Carl Jesse Pollard and Ivan A. Sag. *Information-based syntax and semantics*. CSLI lecture notes. Center for the Study of Language and Information, Stanford, CA, 1987. ISBN 9780937073230.

Djamé Seddah, Reut Tsarfaty, Sandra Kübler, Marie Candito, Jinho D. Choi, Richárd Farkas, Jennifer Foster, Iakes Goenaga, Koldo Gojenola Galletebeitia, Yoav Goldberg, Spence Green, Nizar Habash, Marco Kuhlmann, Wolfgang Maier, Joakim Nivre, Adam Przepiórkowski, Ryan Roth, Wolfgang Seeker, Yannick Versley, Veronika Vincze, Marcin Woliński, Alina Wróblewska, and Eric Villemonte de la Clergerie. Overview of the SPMRL 2013 shared task: A cross-framework evaluation of parsing morphologically rich languages. In Yoav Goldberg, Yuval Marton, Ines Rehbein, and Yannick Versley (eds.), *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pp. 146–182, Seattle, Washington, USA, October 2013. URL `W13-4917`.

Behzad Shayegh, Yanshuai Cao, Xiaodan Zhu, Jackie C. K. Cheung, and Lili Mou. Ensemble distillation for unsupervised constituency parsing. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL `https://openreview.net/forum?id=RR8y0WKrFv`.

Yikang Shen, Zhouhan Lin, Chin-Wei Huang, and Aaron C. Courville. Neural language modeling by jointly learning syntax and lexicon. *ArXiv*, abs/1711.02013, 2017. URL `https://api.semanticscholar.org/CorpusID:3347806`.

N. Clayton Silver and William P. Dunlap. Averaging correlation coefficients: Should fisher's z transformation be used? *Journal of Applied Psychology*, 72(1):146–148, February 1987. ISSN 1939-1854, 0021-9010. doi: 10.1037/0021-9010.72.1.146. URL `http://doi.apa.org/getdoi.cfm?doi=10.1037/0021-9010.72.1.146`.

Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, January 1972. ISSN 0022-0418. doi: 10.1108/eb026526. URL `https://www.emerald.com/insight/content/doi/10.1108/eb026526/full/html`.

C. Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72, January 1904. ISSN 00029556. doi: 10.2307/1412159. URL `https://www.jstor.org/stable/1412159?origin=crossref`.

Valentin I. Spitkovsky, Hiyan Alshawi, Daniel Jurafsky, and Christopher D. Manning. Viterbi training improves unsupervised dependency parsing. In Mirella Lapata and Anoop Sarkar (eds.), *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pp. 9–17, Uppsala, Sweden, July 2010. URL `W10-2902`.

Mark Steedman. *The syntactic process*. Language, speech, and communication. MIT Press, Cambridge, Mass, 2000. ISBN 9780262194204.

Mitchell Stern, Jacob Andreas, and Dan Klein. A minimal span-based neural constituency parser. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 818–827, Vancouver, Canada, July 2017. doi: 10.18653/v1/P17-1076. URL `P17-1076`.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, May 1992. ISSN 0885-6125, 1573-0565. doi: 10.1007/BF00992696. URL `http://link.springer.com/10.1007/BF00992696`.

Pengtao Xie and Eric Xing. A constituent-centric neural architecture for reading comprehension. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1405–1414, Vancouver, Canada, July 2017. doi: 10.18653/v1/P17-1129. URL `P17-1129`.

Zhiyang Xu, Andrew Drozdov, Jay Yoon Lee, Tim O'Gorman, Subendhu Rongali, Dylan Finkbeiner, Shilpa Suresh, Mohit Iyyer, and Andrew McCallum. Improved latent tree induction with distant supervision via span constraints. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4818–4831, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.395. URL `https://aclanthology.org/2021.emnlp-main.395`.

Songlin Yang, Yanpeng Zhao, and Kewei Tu. Neural bi-lexicalized PCFG induction. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 2688–2699, Online, August 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.209. URL `https://aclanthology.org/2021.acl-long.209`.

Songlin Yang, Yanpeng Zhao, and Kewei Tu. PCFGs can do better: Inducing probabilistic context-free grammars with many symbols. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1487–1498, Online, June 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.117. URL `https://aclanthology.org/2021.naacl-main.117`.

Songyang Zhang, Linfeng Song, Lifeng Jin, Kun Xu, Dong Yu, and Jiebo Luo. Video-aided unsupervised grammar induction. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1513–1524, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.119. URL `https://aclanthology.org/2021.naacl-main.119`.

Yanpeng Zhao and Ivan Titov. Visually grounded compound PCFGs. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4369–4379, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.354. URL `https://aclanthology.org/2020.emnlp-main.354`.

Brian D. Ziebart, Andrew L. Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In *AAAI Conference on Artificial Intelligence*, 2008. URL `https://api.semanticscholar.org/CorpusID:336219`.

A Appendix

A.1 Computing Span-posterior probability via back-propagation

In Section 2, we claimed that the span-posterior probability $P(s \text{ is a constituent}|x)$ can be computed using back-propagation.

$$P(s \text{ is a constituent}|x) = \sum_{A \in NT} \frac{\partial \log Z(x)}{\partial \log \beta(s, A)} \tag{13}$$

Proof. Firstly, we define the span-posterior probability as Equation 14. Here s is a substring of x, spanning from the i-th word to the j-th word (i.e., $s := (x_i, ..., x_j)$). Intuitively, s is a constituent if there exists a non-terminal A that is expanded into s.

$$P(s \text{ is a constituent}|x) = \frac{\sum_{A \in NT} P(S \to x \wedge A \to s_{i,j})}{P(x)} \tag{14}$$

We split $P(S \to x \wedge A \to s_{i,j})$ into two parts in Equation 15: $P(S \to x_1, ..., x_{i-1}, A, x_{j+1}, ...)$, the probability of generating words *outside* s, and $P(A \to s)$, the probability generating words *inside* s. The outside probability can be computed using back-propagation (Eisner, 2016). The inside probability is already computed by the β table. Exploiting algebraic transformations shown

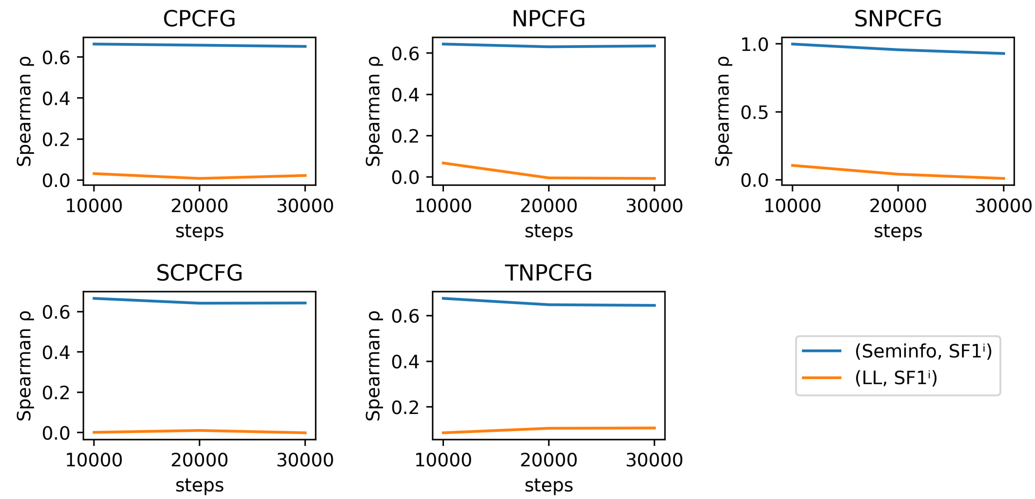

Figure 7: Sentence-level Spearman correlations for models trained for 10k steps, 20k steps, and 30k steps.

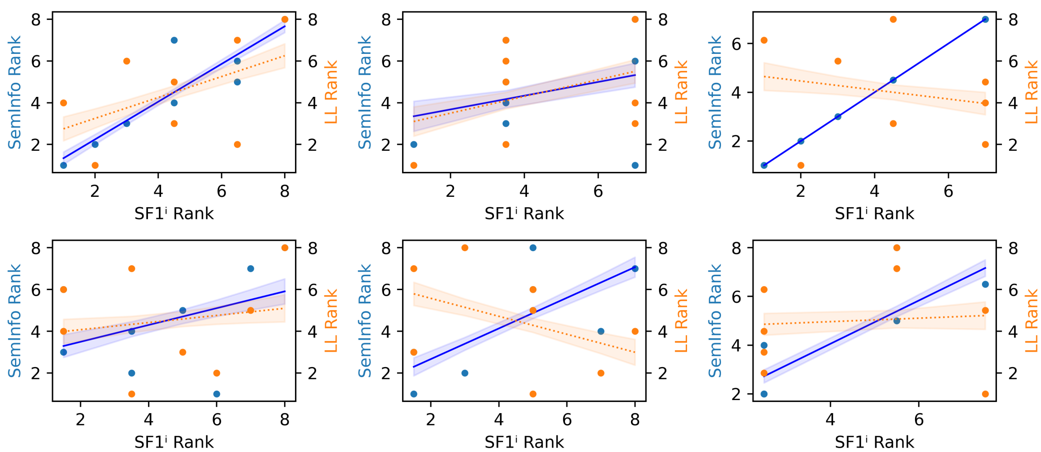

Figure 8: Sentence-level correlation on six random sentences.

in Equation 17, we can derive the formula shown in Equation 13.

$$P(s \text{ is a constituent}|x) = \frac{1}{Z(x)} \sum_{A \in NT} P(S \to x_1, ..., x_{i-1}, A, x_{j+1}, ...) P(A \to s) \tag{15}$$

$$= \frac{1}{Z(x)} \sum_{A \in NT} \frac{\partial Z(x)}{\partial \beta(s, A)} \beta(s, A) \tag{16}$$

$$= \frac{1}{Z(x)} \sum_{A \in NT} Z(x) \frac{\partial \log Z(x)}{\partial \log \beta(s, A)} \frac{1}{\beta(s, A)} \beta(s, A) \tag{17}$$

$$= \sum_{A \in NT} \frac{\partial \log Z(X)}{\partial \log \beta(s, A)} \tag{18}$$

□

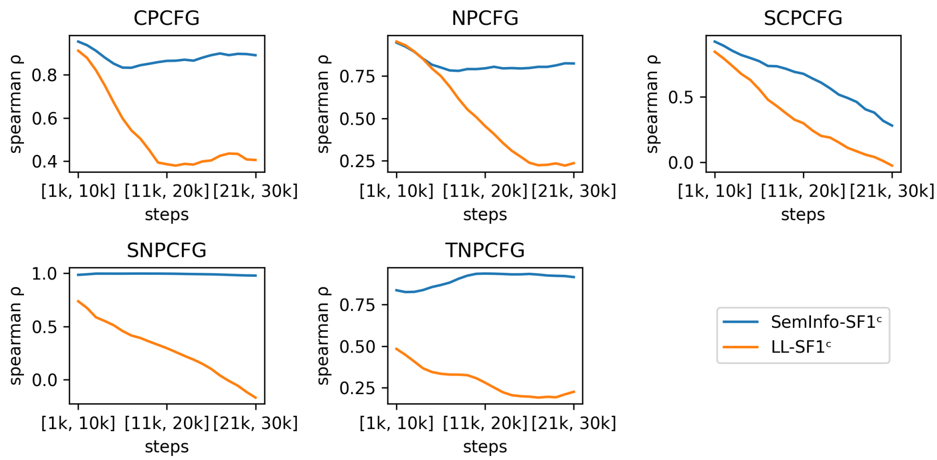

Figure 9: Corpus-level Spearman correlation in different training stages.

A.2 Sentence-level Correlation in different training stages

In Table 2, we showed a strong sentence-level correlation between SemInfo and SF1^i but a weak correlation between LL and SF1^i. Nevertheless, the question of whether the correlation gap is related to the number of training steps remains. Figure 7 excludes the number of training steps as a factor in the correlation gap. In this experiment, we calculate the correlation coefficient for models trained for 10k steps, 20k steps, and 30k steps. We can observe that, for all PCFG variants, the correlation coefficients for (SemInfo, SF1^i) are consistently over 0.6, while the coefficients for (LL, SF1^i) are consistently below 0.1. This result underscores our conclusion that SemInfo can serve as an accurate of parsing accuracy.

A.3 More detailed analysis for corpus-level correlation

Figure 9 shows the corpus-level correlation in different training stages for all five PCFG variants. We observe the same phenomenon explained in Section 5.3.2 for CPCFG, NPCFG, SNPCFG, and TNPCFG. The correlation coefficients for (SemInfo, SF1^c) are consistently above 0.75, whereas the coefficients for (LL, SF1^c) drop quickly as the training progresses. We can observe the stronger correlation between SemInfo and SF1^cin Figure 10. The figure plots the training curves of the corpus-level SF1^cscore, the average SemInfo value, and the average LL value over the English validation set. For example, we can see that SemInfo ranks the NPCFG models represented by the green and grey lines as the lowest, and those represented by the purple and blue lines as the highest. This largely agrees with the SF1^cscores, where the NPCFG models represented by the green and grey lines are among the bottom three worst-performing models, and the models represented by the blue and purple lines are among the top three best-performing models. In comparison, we see that all models have similar LL scores, which indicates LL's inability to rank models in accordance with their parsing performance. These results underscore our conclusion that SemInfo ranks PCFG models better than LL.

In Figure 9, we observe that the correlation strength for (SemInfo, SF1^c) also drops as training processes in SCPCFG. One reason is that SCPCFG failed to explore constituent structures with high SemInfo values. As shown in Figure 10, the average SemInfo value across the eight models is around 42 for SCPCFG, while the average SemInfo value is greater or equal to 45 for the other four PCFG variants. This result indicates that the constituent information provided in low SemInfo regions might contain more noise than the information provided in high SemInfo regions.

A.4 Robustness against Paraphrasing Noises

Table 4 compares the parsing accuracy of NPCFG models trained using seven paraphrasing models. These models are split into three groups: large models (`gpt4o`, `gpt-4o-mini`, `gpt-3.5`),

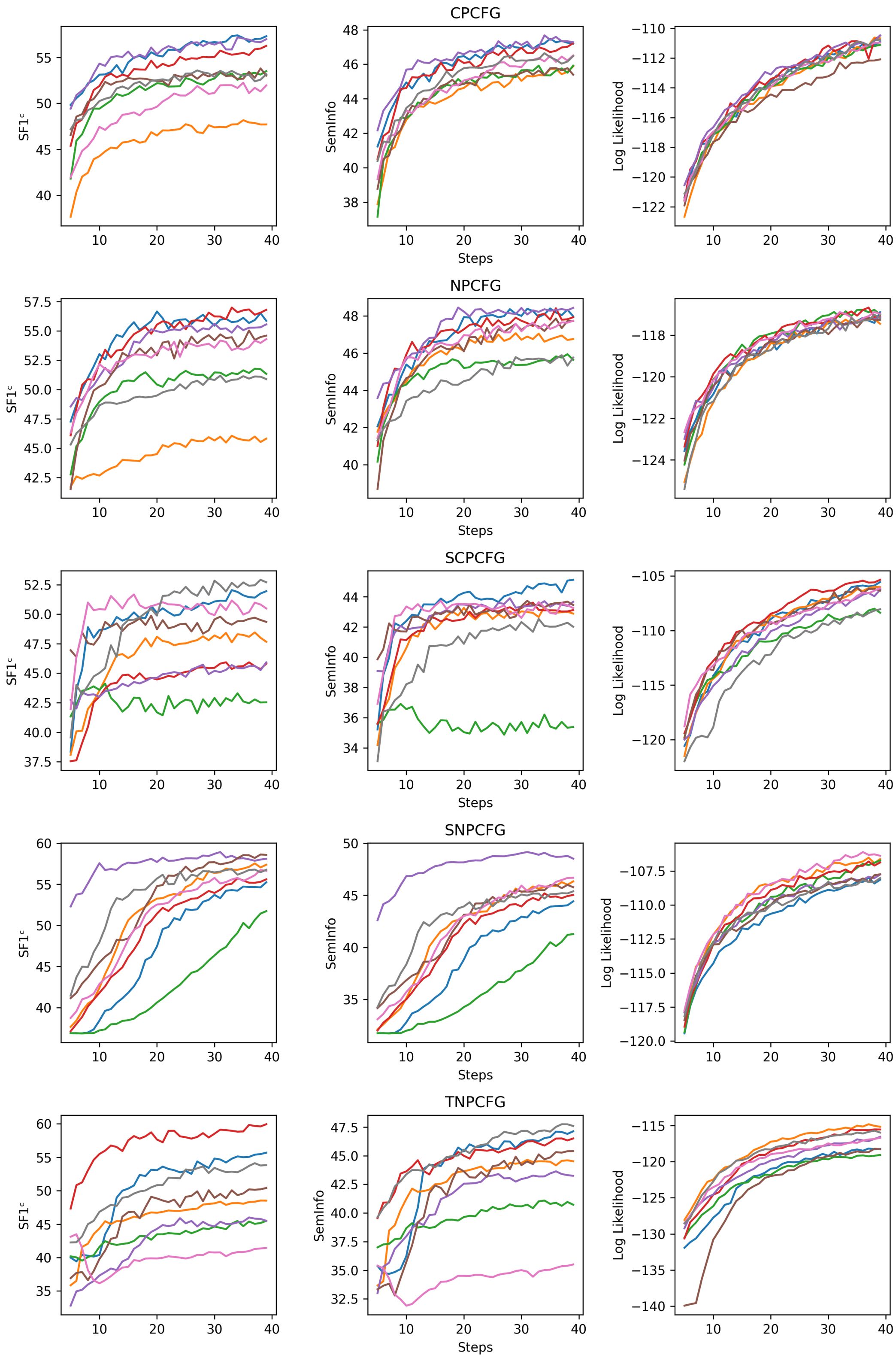

Figure 10: Training curves of SemInfo, LL, and SF1^c. Each line represents the curve for a single PCFG model.

	Paraphrasing Model Variations						
	Large Models			Medium Models		Small Models	
	gpt35	gpt4o	gpt4omini	llama3.2-3b	qwen2.5-3b	llama3.2 1b	qwen2.5-0.5b
SemInfo-NPCFG	66.85±0.25	65.19±0.54	64.45±1.13	63.78±0.55	63.58±0.13	63.10±0.70	59.01±0.24
SemInfo-MTD	55.56	59.45	58.28	55.17	55.03	48.5	43.3
LL-NPCFG	50.96±1.82						
Right Branching	38.4						

Table 4: $SF1^c$ of the NPCFG and MaxTreeDecoding (MTD) parsers using SemInfo values obtained from seven paraphrasing models. LL-NPCFG indicates the $SF1^c$ score of the LL-trained NPCFG parser.

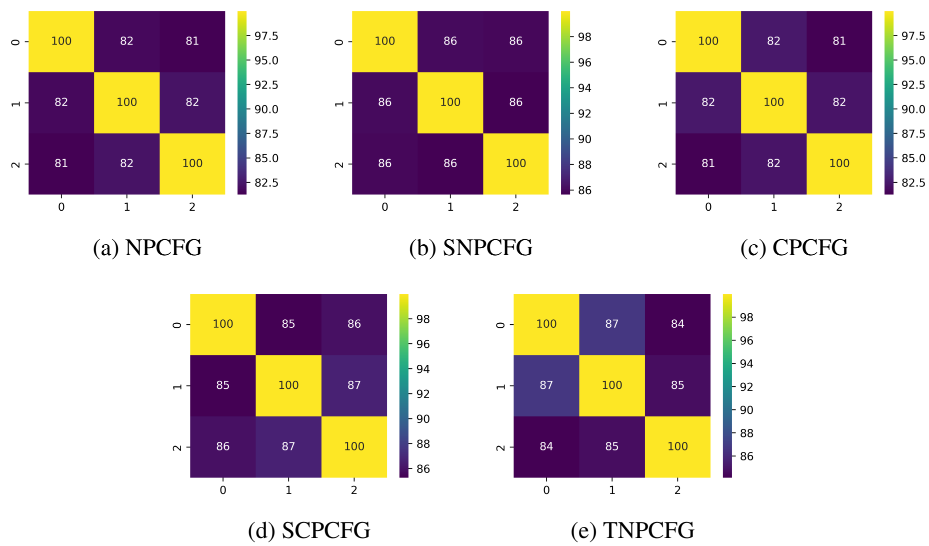

Figure 11: PCFG agreements between independent training runs.

medium models (`llama3.2-3b` and `qwen2.5-3b`), and small models (`llama3.2-1b` and `qwen2.5-0.5b`), each representing paraphrasing models with different levels of noises. The table also includes a MaximumTreeDecoding (MTD) parser, an LL-trained NPCFG parser and a trivial right-branching parser for reference. We use the MTD parser to reflect the paraphrasing quality because its parsing accuracy depends solely on the paraphrasing quality.

We can observe that the SemInfo-trained NPCFG parsers are robust against paraphrasing noises. The accuracy gap between the best (`gpt4o`) and the worst (`qwen2.5-0.5b`) performing MTD parser is 16.15 $SF1^c$ score. In comparison, the gap between the best and worst performing SemInfo-trained NPCFG parser is 7.84 $SF1^c$ score, less than half of the gap in the MTD parser. In addition, we can observe that the PCFG parser can benefit from the SemInfo maximization training, even when using noisy paraphrases. all SemInfo-trained PCFG parsers significantly outperform the baseline LL-trained parser by a large margin. The SemInfo-trained PCFG parser outperforms the LL-trained parser by 9 points when the SemInfo-trained parser is trained using the noisiest paraphrasing model (`qwen2.5-0.5b`). The noise level is significant in the `qwen2.5-0.5b` model because the corresponding MTD parser performs more similarly to the trivial right branching parser than other parsers.

A.5 Potential for Ensembling

Figure 11, Figure 12, and Figure 13 suggests that the SemInfo-trained PCFG would benefit from parser ensembling (Shayegh et al., 2024). We calculate the parser agreement score by evaluating the $SF1^c$ of one parser's prediction against another, following Shayegh et al. (2024). A higher agreement score indicates that the two parsers tend to make more similar predictions. In analyzing the agreement between heterogeneous parsers, we use the `gpt-4o-mini` for training the five PCFG

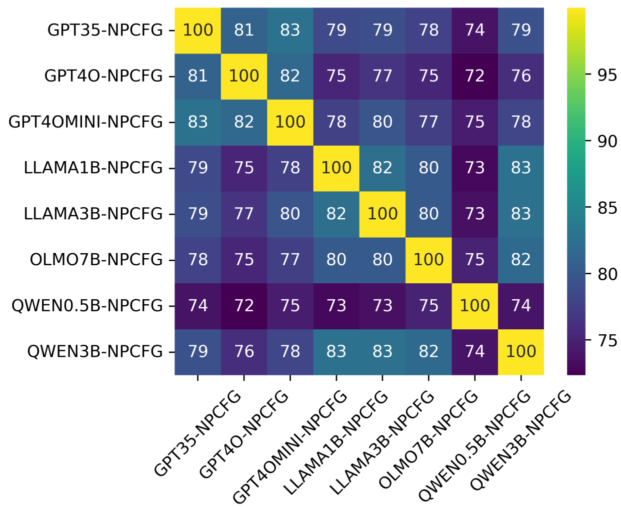

Figure 12: NPCFG parser agreement when trained with different paraphrasing models

	CPCFG		NPCFG		SCPCFG		SNPCFG		TNPCFG		Δ by Type
	SemInfo (Ours)	LL	SemInfo	LL	SemInfo	LL	SemInfo	LL	SemInfo	LL	
NP	$88.88\pm_{0.06}$	$79.77\pm_{1.58}$	$88.98\pm_{0.34}$	$80.63\pm_{2.10}$	$87.45\pm_{1.16}$	$79.41\pm_{1.47}$	$86.51\pm_{0.18}$	$70.95\pm_{1.64}$	$87.89\pm_{1.23}$	$77.73\pm_{5.72}$	+10.90
VP	$71.19\pm_{1.10}$	$40.79\pm_{1.49}$	$65.69\pm_{2.06}$	$28.29\pm_{3.24}$	$73.80\pm_{1.65}$	$28.53\pm_{1.15}$	$76.35\pm_{2.18}$	$80.21\pm_{0.51}$	$72.23\pm_{2.19}$	$45.82\pm_{7.52}$	+26.65
PP	$68.22\pm_{5.68}$	$72.27\pm_{0.47}$	$70.15\pm_{5.42}$	$75.15\pm_{0.83}$	$79.75\pm_{0.57}$	$73.83\pm_{8.94}$	$80.26\pm_{1.45}$	$78.85\pm_{0.98}$	$78.51\pm_{0.83}$	$71.07\pm_{8.49}$	+2.09
SBAR	$80.99\pm_{1.40}$	$52.18\pm_{2.15}$	$80.37\pm_{3.48}$	$56.32\pm_{6.03}$	$84.16\pm_{0.56}$	$40.81\pm_{12.99}$	$82.17\pm_{0.91}$	$81.28\pm_{1.06}$	$82.45\pm_{1.55}$	$54.46\pm_{4.92}$	+22.67
ADVP	$91.87\pm_{0.56}$	$88.38\pm_{0.97}$	$91.48\pm_{0.61}$	$89.78\pm_{1.17}$	$92.22\pm_{1.01}$	$88.57\pm_{4.53}$	$92.11\pm_{0.74}$	$89.67\pm_{0.93}$	$90.93\pm_{1.59}$	$88.07\pm_{0.71}$	+4.48
ADJP	$71.82\pm_{1.43}$	$63.08\pm_{1.90}$	$75.18\pm_{2.85}$	$61.66\pm_{9.97}$	$78.39\pm_{1.78}$	$60.40\pm_{8.03}$	$75.77\pm_{3.74}$	$75.55\pm_{2.18}$	$72.90\pm_{4.19}$	$65.40\pm_{6.60}$	+7.93
Δ by Model	+12.42		+13.05		+20.14		+3.90		+12.76		

Table 5: Recall on six most frequent constituent types. The recall data is calculated over the English test set. Δ by Type indicates the average recall improvement for the constituent type. Δ by Model indicates the average recall improvement for the PCFG variant.

variants. We use the parsing predictions released by Shayegh et al. (2024) for `CPCFG`, `Constest`, `ContextDistort`, `DIORA`, `NPCFG`, and `SDIORA` parsers.

Figure 11 and Figure 12 illustrate the parser agreement between three independent training runs and the agreement among parsers using different paraphrasing models, respectively. The agreement score between independent runs (80-87) might be too high to benefit from the ensemble method. However, the agreement score among parsers using different paraphrasing models (70-83) is similar to the homogeneous parser agreement reported by Shayegh et al. (2024) (74-75). This similarity suggests that ensembling SemInfo-trained PCFG parsers using different paraphrasing models would improve accuracy.

Figure 13 illustrates the agreement among heterogeneous parsers. We can observe that the agreement score between our SemInfo-trained PCFG parsers ranges from 54-58, significantly lower than the agreement scores between runs and paraphrasing models. The agreement score is in the same range as the score among heterogeneous parsers, which indicates a potential accuracy improvement by ensembling our parser with other heterogeneous parsers.

A.6 Recall on six most-frequent constituent types

Table 5 shows the recall of the six most frequent constituent types on the English test set, following Yang et al. (2021b). We see that PCFGs trained with SemInfo achieves significant improvement in Noun Phrases (NP), Verb Phrase (VP), and Subordinate Clauses (SBAR). These three constituents are the most typical constituents that carry semantic information. The significant improvement underscores the importance of semantic information in identifying the constituent structure.

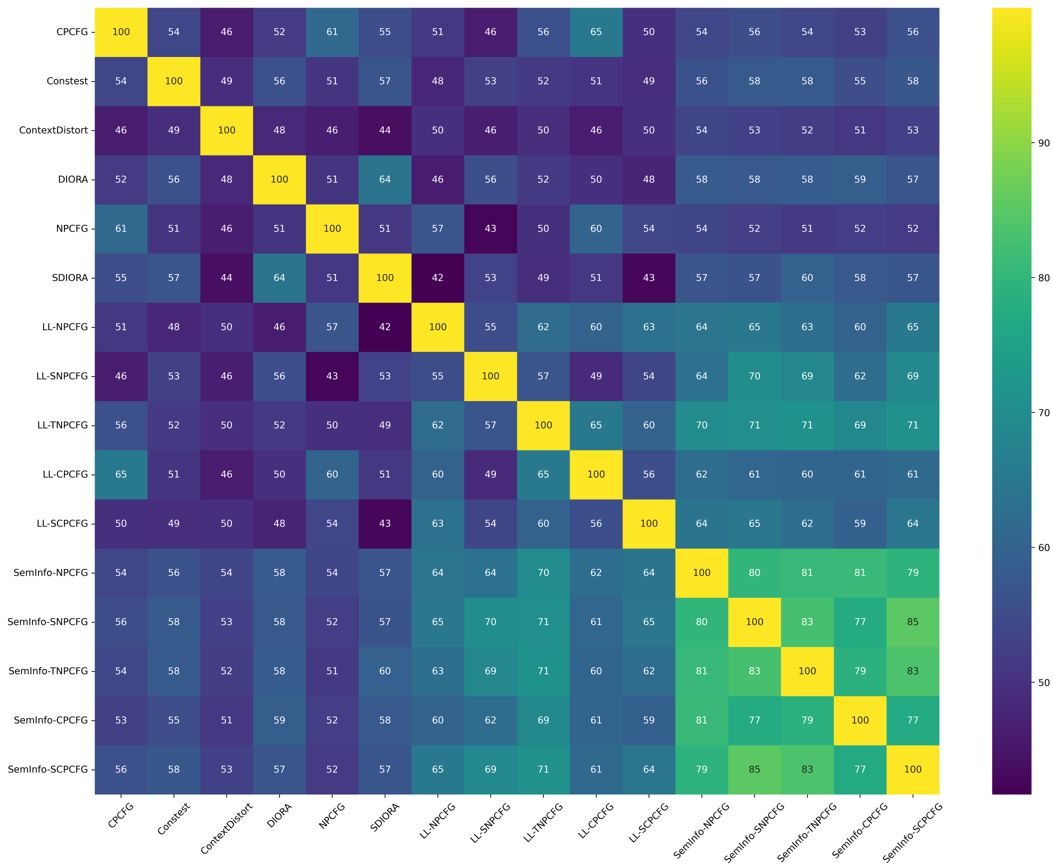

Figure 13: Agreement between heterogeneous parsers

A.7 PARAPHRASING PROMPTS

We use the below prompts to generate paraphrases from the `gpt-4o-mini-2024-07-18` model. {lang} is a placeholder for langauges. For example, we set {lang}="English" when collecting English paraphrases.

- Create grammatical sentences by shuffling the phrases in the below sentence. The generated sentences must be in {lang}. Use the same word as in the original sentence
- Create grammatical sentences by changing the tense in the below sentence. The generated sentences must be in {lang}. Use the same word as in the original sentence.
- Create grammatical sentences by restating the below sentences in passive voice. The generated sentences must be in {lang}. Use the same word as in the original sentence.
- Create grammatical sentences by restating the below sentences in active voice. The generated sentences must be in {lang}. Use the same word as in the original sentence.
- Create grammatical clefting sentences based on the below sentence. The generated sentences must be in {lang}. Use the same word as in the original sentence.
- Create pairs of interrogative and its answers based on the below sentence. The generated sentences must be grammatically correct and be explicit. The sentences must be in {lang}. Use the same word as in the original sentence. The answer to the questions should be a substring of the given sentence.
- Create pairs of confirmatory questions and its answers based on the below sentence. The generated sentences must be grammatically correct and textually diverse. The sentences must be in {lang}. Use the same word as in the original sentence. The answer to the questions should be a substring of the given sentence.
- Create grammatical sentences by performing the topicalization transformation to the below sentence. The sentences must be in {lang}. Use the same word as in the original sentence.
- Create grammatical sentences by performing the heavy NP shift transformation to the below sentence. The sentences must be in {lang}. Use the same word as in the original sentence.

A.8 EXAMPLES OF THE COLLECTED PARAPHRASES

The below list contains examples of our collected paraphrases for *Such agency ‘ self-help ’ borrowing is unauthorized and expensive , far more expensive than direct Treasury borrowing , said Rep. Fortney Stark -LRB- D. , Calif. -RRB- , the bill ’s chief sponsor ..*

- ’Self-help’ borrowing by such agency is unauthorized and expensive, far more expensive than direct Treasury borrowing,’ said Rep. Fortney Stark -LRB- D., Calif. -RRB-, the bill’s chief sponsor.
- Far more expensive than direct Treasury borrowing is such agency ’ self-help ’ borrowing, unauthorized and expensive, said Rep. Fortney Stark -LRB- D., Calif. -RRB-, the bill ’s chief sponsor.
- Yes, he said it is far more expensive than direct Treasury borrowing.
- What is unauthorized and expensive is such agency ’self-help’ borrowing, far more expensive than direct Treasury borrowing, according to Rep. Fortney Stark.
- ’Self-help’ borrowing by such agency is considered unauthorized and is regarded as expensive, far more expensive than direct Treasury borrowing,” said Rep. Fortney Stark -LRB- D., Calif. -RRB-, who is the chief sponsor of the bill.
- According to Rep. Fortney Stark -LRB- D. , Calif. -RRB- , the bill ’s chief sponsor , such agency ’self-help’ borrowing is unauthorized and far more expensive than direct Treasury borrowing.