Investigating Non-local Features for Neural Constituency Parsing

Anonymous ACL submission

Abstract

Thanks to the strong representation power of neural encoders, neural chart-based parsers have achieved highly competitive performance by using local features. Recently, it has been shown that non-local features in CRF structures lead to improvements. In this paper, we investigate injecting non-local features into the training process of a local span-based parser, by predicting constituent n-gram non-local patterns and ensuring consistency between non-local patterns and local constituents. Results show that our simple method gives better results than the self-attentive parser on both PTB and CTB. Besides, our method achieves state-of-the-art BERT-based performance on PTB (95.92 F1) and strong performance on CTB (92.31 F1). Our parser also achieve better or competitive performance in multilingual and zero-shot cross-domain settings compared with the baseline.

1 Introduction

Constituency parsing is a fundamental task in natural language processing, which provides useful information for downstream tasks such as machine translation (Wang et al., 2018), natural language inference (Chen et al., 2017), text summarization (Xu and Durrett, 2019). Over the recent years, with advance in deep learning and pre-training, neural chart-based constituency parsers (Stern et al., 2017a; Kitaev and Klein, 2018) have achieved highly competitive results on benchmarks like Penn Treebank (PTB) and Penn Chinese Treebank (CTB) by solely using local span prediction.

The above methods take the contextualized representation (e.g., BERT) of a text span as input, and use a local classifier network to calculate the scores of the span being a syntactic constituent, together with its constituent label. For testing, output layer uses a non-parametric dynamic programming algorithm (e.g., CKY) to find the highest-scoring tree.

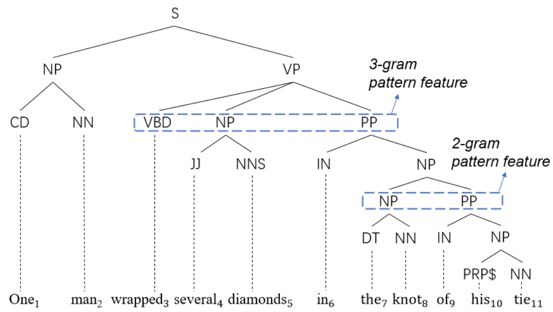


Figure 1: An example of the non-local n-gram *pattern* features: the 3-gram pattern $(3, 11, \{\texttt{VBD NP PP}\})$ is composed of two constituent nodes and one part-of-speech node; the 2-gram pattern $(7, 11, \{\texttt{NP PP}\})$ is composed of two constituent nodes.

Without explicitly modeling structure dependencies between different constituents, the methods give competitive results compared to non-local discrete parsers (Stern et al., 2017a; Kitaev and Klein, 2018). One possible explanation for their strong performance is that the powerful neural encoders are capable of capturing implicit output correlation of the tree structure (Stern et al., 2017a; Gaddy et al., 2018; Teng and Zhang, 2018).

Recent work has shown that modeling non-local output dependencies can benefit neural structured prediction tasks, such as NER (Ma and Hovy, 2016), CCG supertagging (Cui and Zhang, 2019) and dependency parsing (Zhang et al., 2020a). Thus, an interesting research question is whether injecting non-local tree structure features is also beneficial to neural chart-based constituency parsing. To this end, we introduce two auxiliary training objectives. The first is *Pattern Prediction*. As shown in Figure 1, we define *pattern* as the n-gram constituents sharing the same parent.[1] We ask the model to predict the pattern based on its span representation, which directly injects the non-local constituent tree structure to the encoder.

[1] Patterns are mainly composed of n-gram constituents but also include part-of-speech tags as auxiliary.

To allow stronger interaction between non-local patterns and local constituents, we further propose a *Consistency* loss, which regularizes the co-occurrence between constituents and patterns by collecting corpus-level statistics. In particular, we count whether the constituents can be a sub-tree of the pattern based on the training set. For instance, `NP` is legal to occur as a sub-tree of 3-gram pattern `VBD NP PP`, while `S` or `ADJP` cannot be contained within this pattern. The *consistency* loss can be considered as injecting prior linguistic knowledge to our model, which forces the encoder to understand the grammar rules. Non-local dependencies among the constituents that share the same pattern are thus explicitly modeled. We denote our model as Injecting **N**on-local **F**eatures for neural **C**hart-based parsers (NFC).

We conduct experiments on both PTB and CTB. Equipped with BERT, NFC achieves 95.92 F1 on PTB test set, which is the best reported performance for BERT-based single-model parsers. For Chinese constituency parsing, NFC achieves highly competitive results (92.31 F1) on CTB, outperforming the baseline self-attentive parser (91.98 F1) and a 0-th order neural CRF parser (92.27 F1) (Zhang et al., 2020b). To further test the generalization ability, we annotate a multi-domain test set in English, including dialogue, forum, law, literature and review domains. Experiments demonstrate that NFC is robust in zero-shot cross-domain settings. Finally, NFC also performs competitively with other languages using the SPMRL 2013/2014 shared tasks, establishing the best reported results on three rich resource languages. We release our code and models at `https://anonymous`.

2 Related Work

Constituency Parsing. There are mainly two lines of approaches for constituency parsing. Transition-based methods process the input words sequentially and construct the output constituency tree incrementally by predicting a series of local transition actions (Zhang and Clark, 2009; Cross and Huang, 2016; Liu and Zhang, 2017). For these methods, the sequence of transition actions make traversal over a constituent tree. Although transition-based methods directly model partial tree structures, their local decision nature may lead to error propagation (Goldberg and Nivre, 2013) and worse performance compared with methods that model long-term dependencies (McDonald and Nivre, 2011; Zhang and Nivre, 2012). Similar to transition-based methods, NFC also directly models partial tree structures. The difference is that we inject tree structure information using two additional loss functions. Thus, our integration of non-local constituent features is implicit in the encoder, rather than explicit in the decoding process. While the relative effectiveness is empirical, it could potentially alleviate error propagation.

Chart-based methods score each span independently and perform global search over all possible trees to find the highest-score tree given a sentence. Durrett and Klein (2015) represented nonlinear features to a traditional CRF parser computed with a feed-forward neural network. Stern et al. (2017b) first used LSTM to represent span features. Kitaev and Klein (2018) adopted a self-attentive encoder instead of the LSTM encoder to boost parser performance. Mrini et al. (2020) proposed label attention layers to replace self-attention layers. Zhou and Zhao (2019) integrated constituency and dependency structures into head-driven phrase structure grammar. Tian et al. (2020) used span attention to produce span representation to replace the subtraction of the hidden states at the span boundaries. Despite their success, above work mainly focuses on how to better encode features over the input sentence. In contrast, we take the encoder of Kitaev and Klein (2018) intact, being the first to explore new ways to introduce non-local training signal into the local neural chart-based parsers.

Modeling Label Dependency. There is a line of work focusing on modeling non-local output dependencies. Zhang and Zhang (2010) used a Bayesian network to encode the label dependency in multi-label learning. For neural sequence labeling, Zhou and Xu (2015) and Ma and Hovy (2016) built a CRF layer on top of neural encoders to capture label transition patterns. Pislar and Rei (2020) introduced a sentence-level constraint to encourage the model to generate coherent NER predictions. Cui and Zhang (2019) investigated label attention network to model the label dependency by producing label distribution in sequence labeling tasks. Gui et al. (2020) proposed a two-stage label decoding framework based on Bayesian network to model long-term label dependencies. For syntactic parsing, Zhang et al. (2020b) demonstrated that structured Tree CRF can boost parsing performance over graph-based dependency parser. Our work is in line with these in the sense that we consider non-local structure information for neural structure prediction. To our knowledge, we are the first to inject sub-tree structure into neural chart-based encoders for constituency parsing.

3 Baseline

Our baseline is adopted from the parsing model of Kitaev and Klein (2018) and Kitaev et al. (2019). Given a sentence $X = \{x_1, ..., x_n\}$, its corresponding constituency parse tree T is composed by a set of labeled spans

$$T = \{(i_t, j_t, l_t^{\text{c}})\}|_{t=1}^{|T|} \tag{1}$$

where i_t and j_t represent the t-th constituent span's fencepost positions and l_t^c represents the constituent label. The model assigns a score $s(T)$ to tree T, which can be decomposed as

$$s(T) = \sum_{(i,j,l)\in T} s(i, j, l^{\text{c}}) \tag{2}$$

Following Kitaev et al. (2019), we use BERT with a self-attentive encoder as the scoring function $s(i, j, \cdot)$, and a chart decoder to perform a global-optimal search over all possible trees to find the highest-scoring tree given the sentence. In particular, given an input sentence $X = \{x_1, ..., x_n\}$, a list of hidden representations $\mathbf{H}_1^n = \{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_n\}$ is produced by the encoder, where $\mathbf{h}_i$ is a hidden representation of the input token x_i. Following previous work, the representation of a span (i, j) is constructed by:

$$\mathbf{v}_{i,j} = \mathbf{h}_j - \mathbf{h}_i \tag{3}$$

Finally, $\mathbf{v}_{i,j}$ is fed into an MLP to produce real-valued scores $s(i, j, \cdot)$ for all constituency labels:

$$s(i, j, \cdot) = \mathbf{W}_2^{\text{c}}\text{ReLU}(\mathbf{W}_1^{\text{c}}\mathbf{v}_{i,j} + \mathbf{b}_1^{\text{c}}) + \mathbf{b}_2^{\text{c}} \tag{4}$$

where $\mathbf{W}_1^{\text{c}}$, $\mathbf{W}_2^{\text{c}}$, $\mathbf{b}_1^{\text{c}}$ and $\mathbf{b}_2^{\text{c}}$ are trainable parameters, $\mathbf{W}_2^{\text{c}} \in \mathbb{R}^{|H|\times|L^{\text{c}}|}$ can be considered as the constituency label embedding matrix (Cui and Zhang, 2019), where each column in $\mathbf{W}_2^{\text{c}}$ corresponds to the embedding of a particular constituent label. $|H|$ represents the hidden dimension and $|L^{\text{c}}|$ is the size of the constituency label set.

Training. The model is trained to satisfy the margin-based constraints

$$s(T^*) \geq s(T) + \Delta(T, T^*) \tag{5}$$

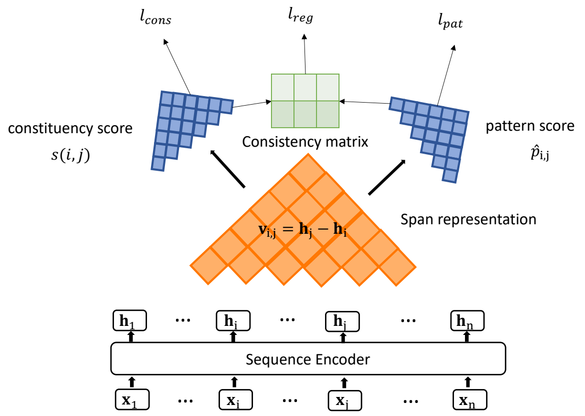


Figure 2: The three training objectives in NFC.

where T^* denotes the gold parse tree, and Δ is Hamming loss. The hinge loss can be written as

$$\mathcal{L}_{\text{cons}} = \max\left(0, \max_{T \neq T^*}[s(T) + \Delta(T, T^*)] - s(T^*)\right) \tag{6}$$

During inference time, the most-optimal tree

$$\hat{T} = \underset{T}{\text{argmax}}\, s(T) \tag{7}$$

is obtained using a CKY-like algorithm.

4 Additional Training Objectives

We propose two auxiliary training objectives to inject non-local features into the encoder, which rely only on the annotations in the constituency treebank, but not external resources.

4.1 Instance-level Pattern Loss

We define n-gram constituents, which shares the same parent node, as a pattern. We use a triplet $(i^{\text{p}}, j^{\text{p}}, l^{\text{p}})$ to denote a pattern span beginning from the i^{p}-th word and ending at j^{p}-th word. l^{p} is the corresponding pattern label. Given a constituency parse tree in Figure 1, $(3, 11, \{$`VBD NP PP`$\})$ is a 3-gram pattern.

Similar to Eq 4, an MLP is used for transforming span representations to pattern prediction probabilities:

$$\hat{p}_{i,j} = \text{Softmax}\left(\mathbf{W}_2^{\text{p}}\text{ReLU}(\mathbf{W}_1^{\text{p}}\mathbf{v}_{i,j} + \mathbf{b}_1^{\text{p}}) + \mathbf{b}_2^{\text{p}}\right) \tag{8}$$

where $\mathbf{W}_1^{\text{p}}$, $\mathbf{W}_2^{\text{p}}$, $\mathbf{b}_1^{\text{p}}$ and $\mathbf{b}_2^{\text{p}}$ are trainable parameters, $\mathbf{W}_2^{\text{p}} \in \mathbb{R}^{|H|\times|L^{\text{p}}|}$ can be considered as the pattern label embedding matrix, where each column in $\mathbf{W}_2^{p}$ corresponds to the embedding of a particular pattern label. $|L^{\text{p}}|$ represents the size of the pattern label set. For each instance, the cross-entropy loss between the predicted patterns and the gold patterns are calculated as

$$\mathcal{L}_{pat} = -\sum_{i=1}^{n}\sum_{j=1}^{n} p_{i,j} \log \hat{p}_{i,j} \quad (9)$$

We use the span-level cross-entropy loss for patterns (Eq 9) instead of the margin loss in Eq 6, because our pattern-prediction objective aims to augment span representations via greedily classifying each pattern span, rather than to reconstruct the constituency parse tree through dynamic programming.

4.2 Corpus-level Consistency Loss

Constituency scores and pattern probabilities are produced based on a shared span representation; however, the two are subsequently separately predicted. Therefore, although the span representations contain both constituent and pattern information, the dependencies between constituent and pattern predictions are not explicitly modeled. Intuitively, constituents are distributed non-uniformly in patterns, and such correlation can be obtained in the corpus-level statistic. We propose a consistency loss, which explicitly models the non-local dependencies among constituents that belong to the same pattern.

This loss can be understood first at the instance level. In particular, if a constituent span $(i_t, j_t, l_t^{\mathrm{c}})$ is a subtree of a pattern span $(i_{t'}, j_{t'}, l_{t'}^{\mathrm{p}})$, i.e. $i_t >= i_{t'}$ and $j_t <= j_{t'}$, where $l_t^{\mathrm{c}} = L^{\mathrm{c}}[a]$ (the a-th constituent label in L^{c}) and $l_{t'}^{\mathrm{p}} = L^{\mathrm{p}}[b]$ (the b-th pattern label in L^{p}), we define $L^{\mathrm{c}}[a]$ and $L^{\mathrm{p}}[b]$ to be *consistent* (denoted as $y_{a,b} = 1$). Otherwise we consider it to be *non-consistent* (denoted as $y_{a,b} = 0$). This yields a consistency matrix $\mathbf{Y} \in \mathbb{R}^{|L^c| \times |L^{\mathrm{p}}|}$ for each instance. The gold consistency matrix $\mathbf{Y}$ provides information regarding non-local dependencies among constituents and patterns.

An intuitive method to predict the consistency matrix $\mathbf{Y}$ is to make use of the constituency label embedding matrix $\mathbf{W}_2^{\mathrm{p}}$, the pattern label embedding matrix $\mathbf{W}_2^{\mathrm{c}}$ and the span representations $\mathbf{V}$:

$$\hat{\mathbf{Y}} = \mathrm{Sigmoid}\big((\mathbf{W}_2^{\mathrm{c}}\mathbf{U}_1\mathbf{V})(\mathbf{V}^{\mathsf{T}}\mathbf{U}_2{\mathbf{W}_2^{\mathrm{p}}}^{\mathsf{T}})\big) \quad (10)$$

where $\mathbf{U}_1, \mathbf{U}_2 \in \mathbb{R}^{|H| \times |H|}$ are trainable parameters.

Eq 10 can be predicted on the instance-level for ensuring consistency between patterns and constituent. However, this naive method is difficult for training, and computationally infeasible, because the span representation matrix $\mathbf{V} \in \mathbb{R}^{|H| \times n^2}$ is composed of n^2 span representations $\mathbf{v}_{i,j} \in \mathbb{R}^{|H|}$ and the asymptotic complexity is:

$$O\Big((|L^{\mathrm{p}}| + |L^{\mathrm{c}}|)(|H|^2 + n^2|H|) + |L^{\mathrm{p}}||L^{\mathrm{c}}|n^2\Big) \quad (11)$$

for a single training instance. We instead use a corpus-level constraint on the non-local dependencies among constituents and patterns. In this way, Eq 10 is reduced to be independent of individual span representations:

$$\hat{\mathbf{Y}} = \mathrm{Sigmoid}\big(\mathbf{W}_2^{\mathrm{c}}\mathbf{U}{\mathbf{W}_2^{\mathrm{p}}}^{\mathsf{T}}\big) \quad (12)$$

where $\mathbf{U} \in \mathbb{R}^{|H| \times |H|}$ is trainable.

This trick decreases the asymptotic complexity to $O(|L^{\mathrm{c}}||H|^2 + |L^{\mathrm{p}}||L^{\mathrm{c}}||H|)$. The cross-entropy loss between the predicted consistency matrix and gold consistency labels is used to optimize the model:

$$\mathcal{L}_{reg} = -\sum_{a=1}^{|L^{\mathrm{c}}|}\sum_{b=1}^{|L^{\mathrm{p}}|} y_{a,b} \log \hat{y}_{a,b} \quad (13)$$

The corpus-level constraint can be considered as a prior linguistic knowledge statistic from the treebank, which forces the encoder to understand the grammar rules.

4.3 Training

Given a constituency treebank, we minimize the sum of the three objectives to optimize the parser:

$$\mathcal{L} = \mathcal{L}_{cons} + \mathcal{L}_{pat} + \mathcal{L}_{reg} \quad (14)$$

4.4 Computational Cost

The number of training parameters increased by NFC is $\mathbf{W}_1^{\mathrm{p}} \in \mathbb{R}^{|H| \times |H|}$, $\mathbf{W}_2^{\mathrm{p}} \in \mathbb{R}^{|H| \times |L^p|}$, $\mathbf{b}_1^{\mathrm{p}} \in \mathbb{R}^{|H|}$ and $\mathbf{b}_2^{\mathrm{p}} \in \mathbb{R}^{|L^p|}$ in Eq 8 and $\mathbf{U} \in \mathbb{R}^{|H| \times |H|}$ in Eq 12. Taking training model on PTB as an example, NFC adds less than 0.7M parameters to 342M parameters baseline model (Kitaev and Klein, 2018) based on BERT-large-uncased during training. NFC is identical to our baseline self-attentive parser (Kitaev and Klein, 2018) during inference.

5 Experiments

We empirically compare NFC with the baseline parser in different settings, including in-domain, cross-domain and multilingual benchmarks.

Data	Lang / Domain	# Train	# Dev	# Test
PTB	English	39,832	1,700	2,416
CTB	Chinese	17,544	352	348
SPMRL	French	14,759	1,235	2,541
SPMRL	German	40,472	5,000	5,000
SPMRL	Korean	23,010	2,066	2,287
SPMRL	Basque	7,577	948	946
SPMRL	Polish	6,578	821	822
SPMRL	Hungarian	8,146	1,051	1,009
MCTB	Dialogue	-	-	1,000
MCTB	Forum	-	-	1,000
MCTB	Law	-	-	1,000
MCTB	Literature	-	-	1,000
MCTB	Review	-	-	1,000

Table 1: Dataset statistics. # - number of sentences.

	PTB	CTB
w/o pattern	95.65	94.11
2-gram	95.67	**94.29**
3-gram	**95.77**	94.14
4-gram	95.70	93.91
2-gram & 3-gram	95.68	94.16
3-gram & 4-gram	95.71	93.97

Table 2: F1 score on the development set of PTB and CTB using different n-gram pattern features with consistency loss. w/o pattern indicates the baseline parser.

5.1 Dataset

Table 1 shows the detailed statistic of our datasets. We conduct experiments on both English and Chinese, using the Penn Treebank (Marcus et al., 1993) as our English dataset, with standard splits of section 02-21 for training, section 22 for development and section 23 for testing. For Chinese, we split the Penn Chinese Treebank (CTB) 5.1 (Xue et al., 2005), taking articles 001-270 and 440-1151 as training set, articles 301-325 as development set and articles 271-300 as test set.

In the multilingual settings, we select three rich resource language from the SPMRL 2013-2014 shared task (Seddah et al., 2013): French, German and Korean, which include at least 10,000 training instances, and three low-resource language: Hungarian, Basque and Polish.

Cross-domain Dataset. To test the robustness of our methods across difference domains, we further annotate five test set in dialogue, forum, law, literature and review domains. For the dialogue domain, we randomly sample dialogue utterances from Wizard of Wikipedia (Dinan et al., 2019), which is a chit-chat dialogue benchmark produced by humans. For the forum domain, we use users' communication records from Reddit, crawled and released by Völske et al. (2017). For the law domain, we sample text from European Court of Human Rights Database (Stiansen and Voeten, 2019), which includes detailing judicial decision patterns. For the literature domain, we download literary fictions from Project Gutenberg[2]. For the review domain, we use plain text across a variety of product genres, released by SNAP Amazon Review Dataset (He and McAuley, 2016). After obtaining the plain text, we ask linguistic experts to annotate constituency parse tree by strictly following the PTB guideline. We name our dataset as **M**ulti-domain **C**onstituency **T**ree**b**ank (MCTB). More details of the dataset will be documented separately.

5.2 Development Experiments

The sizes of non-local n-gram windows may have an essential influence on parser performance. Intuitively, larger n-gram window sizes allow capturing more global information. We perform development experiments to decide the window size of non-local pattern features for both PTB and CTB. As shown in Table 2, 3-gram pattern features give the best performance for PTB while 2-gram works best for CTB. We thus choose the settings with the best development performance for our experiments. We conduct multilingual experiments following the setting for PTB.

5.3 Setup

Our code is based on the open-sourced code of Kitaev and Klein (2018)[3]. The training process gets terminated if no improvement on development F1 is obtained in the last 60 epochs. We evaluate the models which have the best F1 on the development set. For fair comparison, all reported results and baselines are augmented with BERT. We adopt `BERT-large-uncased` for English, `BERT-base` for Chinese and `BERT-multi-lingual-uncased` for other languages. Most of our hyper-parameters are adopted from Kitaev and Klein (2018) and Fried et al. (2019). For scales of the two additional losses, we set the scale of pattern loss to 1.0 and the scale of consistency loss to 5.0 for all experiments.

To reduce the model size, we filter out those non-local pattern features that appear less than 5 times in the PTB training set and those that account for less than 0.5% of all pattern occurrences in the CTB training set. The out-of-vocabulary patterns are set as $<$ UNK $>$. This results in moderate pattern

[2] https://www.gutenberg.org/

[3] Available at https://github.com/nikitakit/self-attentive-parser.

Model	LR	LP	F1
Liu and Zhang (2017) ⋄	-	-	95.71
Kitaev and Klein (2018)	95.46	95.73	95.59
Zhou and Zhao (2019)	95.51	95.93	95.72
Zhou and Zhao (2019) *	95.70	95.98	95.84
Zhang et al. (2020b)	95.53	95.85	95.69
Nguyen et al. (2020)	-	-	95.48
Tian et al. (2020)	95.58	96.11	95.85
This work			
Kitaev and Klein (2018) †	95.56	95.89	95.72
NFC w/o $\mathcal{L}_{reg}$	95.49	96.07	95.78
NFC	**95.70**	**96.14**	**95.92**

Table 3: Performance (w/ BERT) on the test set of PTB. † indicates our reproduced results, which is also the baseline that our method is built upon. * indicates training with extra supervision from dependency parsing data. ⋄ indicates that the results are reported by the re-implementation of Fried et al. (2019).

Model	LR	LP	F1
Liu and Zhang (2017) ⋄	-	-	91.81
Kitaev and Klein (2018)	91.55	91.96	91.75
Zhang et al. (2020b)	92.04	92.51	92.27
Zhou and Zhao (2019)	91.14	93.09	92.10
Tian et al. (2020)	92.14	92.25	92.20
This work			
Kitaev and Klein (2018) †	91.80	92.23	91.98
NFC w/o $\mathcal{L}_{reg}$	91.87	92.40	92.13
NFC	**92.17**	**92.45**	**92.31**
w/ External Dependency Supervision			
Zhou and Zhao (2019) *	92.03	92.33	92.18
Mrini et al. (2020)*	91.85	93.45	92.64

Table 4: Constituency parsing performance (w/ BERT) on the test set of CTB 5.1. The symbols (†, * and ⋄) are explained in Table 3.

vocabulary sizes of 841 for PTB and 514 for CTB. For evaluation on PTB, CTB and cross-domain dataset, we use the EVALB script for evaluation. For the SPMRL datasets, we follow the same setup in EVALB as Kitaev and Klein (2018).

5.4 In-domain Experiments

We report the performance of our method on the test sets of PTB and CTB in Table 3 and 4, respectively. Compared with the baseline parser (Kitaev and Klein, 2018), our method obtains an absolute improvement of 0.20% F1 on PTB (p<0.01) and 0.33% F1 on CTB (p<0.01), which verifies the effectiveness of injecting non-local features into neural local span-based constituency parsers. Note that the proposed method adds less than 0.7M parameters to the 342M parameter baseline model using BERT-large.

The parser trained with both the pattern loss (Section 4.1) and consistency loss (Section 4.2) outperforms the one trained only with pattern loss by 0.14% F1 (p<0.01). This suggests that the constraints between constituents and non-local pattern features are crucial for injecting non-local features into local span-based parsers. One possible explanation for the improvement is that the constraints may bridge the gap between local and non-local supervision signals, since these two are originally separately predicted while merely sharing the same encoder in the training phase.

We further compare our method with the recent state-of-the-art parsers on PTB and CTB. Liu and Zhang (2017) propose an in-order transition-based constituency parser. Kitaev and Klein (2018) use self-attentive layers instead of LSTM layers to boost performance. Zhou and Zhao (2019) jointly optimize constituency parsing and dependency parsing objectives using head-driven phrase structure grammar. Mrini et al. (2020) extend Zhou and Zhao (2019) by introducing label attention layers. Zhang et al. (2020b) integrate a CRF layer to a chart-based parser for structural training (without non-local features). Tian et al. (2020) use span attention for better span representation.

Compared with these methods, the proposed method achieves an F1 of 95.92%, which exceeds previous best numbers for BERT-based single-model parsers on the PTB test set. We further compare experiments for five runs, and find that NFC significantly outperforms Kitaev and Klein (2018) (p<0.01). The test score of 92.31% F1 on CTB significantly outperforms the result (91.98% F1) of the baseline (p<0.01). Compared with the CRF parser of Zhang et al. (2020b), our method gives better scores without global normalization in training. This shows the effectiveness of integrating non-local information during training using our simple regularization. The result is highly competitive with the current best result (Mrini et al., 2020), which is obtained by using external dependency parsing data.

5.5 Cross-domain Experiments

We compare the generalization of our methods with baselines in Table 5. In particular, all the parsers are trained on PTB training and validated on PTB development, and are tested on cross-domain test in the zero-shot setting. As shown in the table, our model achieves 5 best-reported results among 6 cross-domain test sets with an averaged F1 score of 87.03%, outperforming our baseline parser by

Model	In-domain	Cross-domain						
	PTB	Bio	Dialogue	Forum	Law	Literature	Review	Avg
Liu and Zhang (2017)	95.65	86.33	79.89	83.02	90.66	84.68	78.83	83.90
Zhou and Zhao (2019)	95.84	86.14	81.34	82.73	89.86	84.95	79.65	84.11
Kitaev and Klein (2018)	95.72	**86.61**	82.53	84.59	92.37	87.56	80.64	85.72
NFC	**95.92**	86.43	**84.10**	**86.08**	**92.64**	**90.65**	**82.30**	**87.03**

Table 5: Constituency parsing results with BERT (F1 scores) on the cross-domain test set.

Model	Rich resource				Low Resource				Avg
	French	German	Korean	Avg	Hungarian	Basque	Polish	Avg	
Kitaev and Klein (2018)	87.42	90.20	88.80	88.81	94.90	91.63	**96.36**	94.30	91.55
Nguyen et al. (2020)	86.69	90.28	88.71	88.56	94.24	**92.02**	96.14	94.13	91.34
Kitaev and Klein (2018) †	87.38	90.25	88.91	88.85	94.56	91.66	96.14	94.12	91.48
NFC	**87.51**	**90.43**	**89.07**	**89.00**	**94.95**	91.73	96.33	**94.34**	**91.67**

Table 6: Multilingual Experiment results on SPMRL test-sets. † indicates our reproduced baselines.

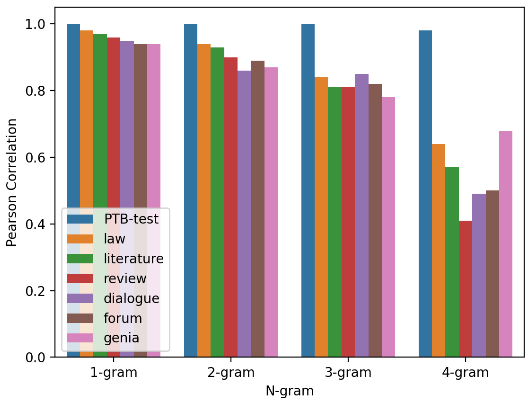
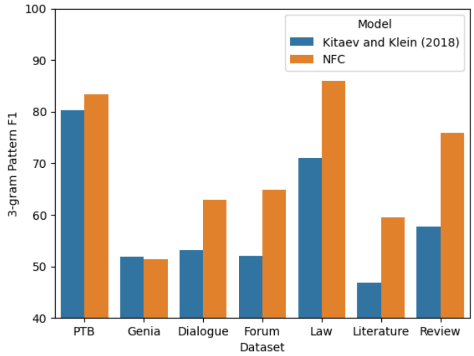


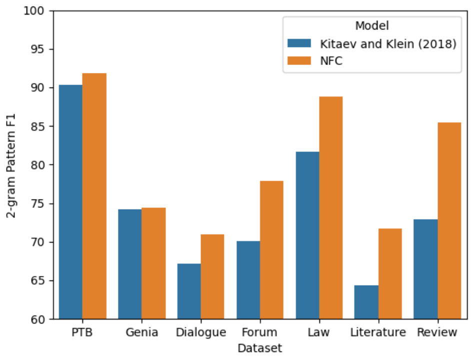
Figure 3: Pearson correlation of n-gram pattern distribution between PTB training set and different test set.

1.31% points. This shows that structure information is useful for improving cross-domain performance, which is consistent with findings from previous work (Fried et al., 2019).

To better understand the benefit of pattern features, we calculate Pearson correlation of n-gram pattern distributions between the PTB training set and various test sets in Figure 3. First, we find that the correlation between the PTB training set and the PTB test set is close to 1.0, which verifies the effectiveness of the corpus-level pattern knowledge during inference. Second, the 3-gram pattern correlation of all domains exceeds 0.75, demonstrating that n-gram pattern knowledge is robust across domains, which supports the strong performance of NFC in the zero-shot cross-domain setting. Third, pattern correlation decreases significantly as n increases, which suggests that transferable non-local information is limited to a certain window size of n-gram constituents.



(a) F1 scores measured by 3-gram pattern.



(b) F1 scores measured by 2-gram pattern.

Figure 4: Pattern-level F1 on different English datasets. Noted that we train NFC based on 3-gram pattern in English. There is no direct supervision signal for 2-gram pattern.

5.6 Multilingual Experiments

We compare NFC with Kitaev and Klein (2018) and Nguyen et al. (2020) on SPMRL. The results are shown in Table 6. Nguyen et al. (2020) use pointer network to predict a sequence of pointing decisions for constituency parsing. As can be seen, Nguyen et al. (2020) do not show obvious advan-

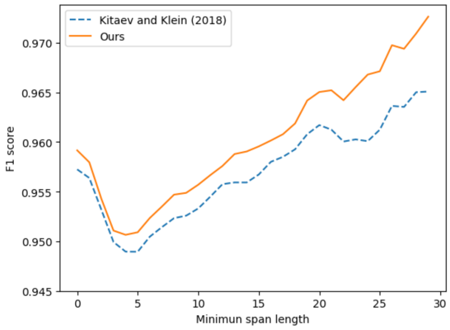


Figure 5: F1 scores versus minimum constituent span length on PTB test set. Note that constituent spans shorter than 30 accounts for approximately 98.5% of all for the PTB test set.

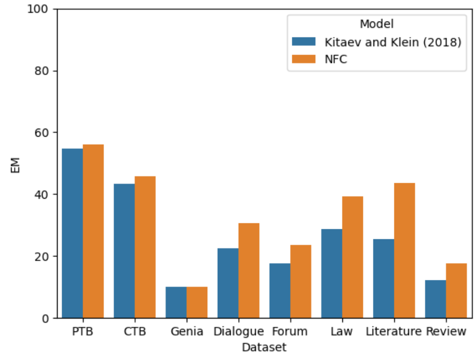


Figure 6: Exact matching (EM) score across different domains. EM indicates the percentage of sentences whose predicted trees are entirely correct.

tages over Kitaev and Klein (2018). NFC outperforms these two methods on three rich resource languages. For example, NFC achieves 89.07% F1 on Korean, outperforming Kitaev and Klein (2018) by 0.27% points, suggesting that NFC is generally effective across languages. However, NFC does not give better results compared with Kitaev and Klein (2018) on low-resource languages. One possible explanation is that it is difficult to obtain prior linguistic knowledge from corpus-level statistics by using a relatively small number of instances.

6 Analysis

6.1 n-gram Pattern Level Performance

Figure 4 shows the pattern-level F1 before and after introducing the two auxiliary training objectives. In particular, we calculate the pattern-level F1 by calculating the F1 score for pattern prediction. Although our baseline parser with BERT achieves 95.76% F1 scores on PTB, the pattern-level F1 is 80.28% measured by 3-gram. When testing on the dialogue domain, the result is reduced to only 53.15% F1, which indicates that even a strong neural encoder still has difficulties capturing constituent dependency from the input sequence alone. After introducing the pattern and consistency losses, NFC significantly outperforms the baseline parser measured by 3-gram pattern F1. Though there is no direct supervision signal for 2-gram pattern, NFC also gives better results on pattern F1 of 2-gram, which are subsumed by 3-gram patterns. This suggests that NFC can effectively represent sub-tree structures.

6.2 F1 against Span Size

We compare the performance of the baseline and our method on constituent spans with different lengths. Figure 5 shows the trends of F1 scores on the PTB test set as the minimum constituent span length increases. Our method shows a minor improvement at the beginning, but the gap becomes more evident when the minimum span length increases, demonstrating its advantage in capturing more sophisticated constituency label dependency.

6.3 Exact Match

Exact match score represents the percentage of sentences whose predicted trees are entirely the same as the golden trees. Producing exact matched trees could improve user experiences in practical scenarios and benefit downstream applications on other tasks (Petrov and Klein, 2007; Kummerfeld et al., 2012). We compare exact match scores of NFC with that of the baseline parser. As shown in Figure 6, NFC achieves large improvements in exact match score for all domains. For instance, NFC gets 43.65% exact match score in the literature domain, outperforming the baseline by 25.42% points. We assume that this results from the fact that NFC successfully ensure the output tree structure by modeling non-local correlation.

7 Conclusion

We investigated graph-based constituency parsing with non-local features – both in the sense that features are not restricted to one constituent, and in the sense that they are not restricted to each training instance. Experimental results verify the effectiveness of injecting non-local features to neural chart-based constituency parsing. Equipped with pre-trained BERT, our method achieves 95.92% F1 on PTB and 92.31% F1 on CTB. We further demonstrated that the proposed method gives better or competitive results in multilingual and zero-shot cross-domain settings.

References

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668, Vancouver, Canada. Association for Computational Linguistics.

James Cross and Liang Huang. 2016. Span-based constituency parsing with a structure-label system and provably optimal dynamic oracles. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1–11, Austin, Texas. Association for Computational Linguistics.

Leyang Cui and Yue Zhang. 2019. Hierarchically-refined label attention network for sequence labeling. 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 4115–4128, Hong Kong, China. Association for Computational Linguistics.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *International Conference on Learning Representations*.

Greg Durrett and Dan Klein. 2015. Neural CRF parsing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 302–312, Beijing, China. Association for Computational Linguistics.

Daniel Fried, Nikita Kitaev, and Dan Klein. 2019. Cross-domain generalization of neural constituency parsers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 323–330, Florence, Italy. Association for Computational Linguistics.

David Gaddy, Mitchell Stern, and Dan Klein. 2018. What's going on in neural constituency parsers? an analysis. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 999–1010, New Orleans, Louisiana. Association for Computational Linguistics.

Yoav Goldberg and Joakim Nivre. 2013. Training deterministic parsers with non-deterministic oracles. *Transactions of the Association for Computational Linguistics*, 1:403–414.

Tao Gui, Jiacheng Ye, Qi Zhang, Zhengyan Li, Zichu Fei, Yeyun Gong, and Xuanjing Huang. 2020. Uncertainty-aware label refinement for sequence labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2316–2326, Online. Association for Computational Linguistics.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 507–517, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Nikita Kitaev, Steven Cao, and Dan Klein. 2019. Multilingual constituency parsing with self-attention and pre-training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505, Florence, Italy. Association for Computational Linguistics.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.

Jonathan K. Kummerfeld, David Hall, James R. Curran, and Dan Klein. 2012. Parser showdown at the Wall Street corral: An empirical investigation of error types in parser output. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1048–1059, Jeju Island, Korea. Association for Computational Linguistics.

Jiangming Liu and Yue Zhang. 2017. In-order transition-based constituent parsing. *Transactions of the Association for Computational Linguistics*, 5:413–424.

Xuezhe Ma and Eduard H. Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *CoRR*, abs/1603.01354.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Ryan McDonald and Joakim Nivre. 2011. Analyzing and integrating dependency parsers. *Computational Linguistics*, 37(1):197–230.

Khalil Mrini, Franck Dernoncourt, Quan Hung Tran, Trung Bui, Walter Chang, and Ndapa Nakashole. 2020. Rethinking self-attention: Towards interpretability in neural parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 731–742, Online. Association for Computational Linguistics.

Thanh-Tung Nguyen, Xuan-Phi Nguyen, Shafiq Joty, and Xiaoli Li. 2020. Efficient constituency parsing by pointing. In *Proceedings of the 58th Annual*

Meeting of the Association for Computational Linguistics, pages 3284–3294, Online. Association for Computational Linguistics.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 404–411, Rochester, New York. Association for Computational Linguistics.

Miruna Pislar and Marek Rei. 2020. Seeing both the forest and the trees: Multi-head attention for joint classification on different compositional levels. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3761–3775, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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. 2013. Overview of the SPMRL 2013 shared task: A cross-framework evaluation of parsing morphologically rich languages. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 146–182, Seattle, Washington, USA. Association for Computational Linguistics.

Mitchell Stern, Jacob Andreas, and Dan Klein. 2017a. A minimal span-based neural constituency parser. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 818–827, Vancouver, Canada. Association for Computational Linguistics.

Mitchell Stern, Jacob Andreas, and Dan Klein. 2017b. A minimal span-based neural constituency parser. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 818–827, Vancouver, Canada. Association for Computational Linguistics.

Øyvind Stiansen and Erik Voeten. 2019. ECtHR judgments.

Zhiyang Teng and Yue Zhang. 2018. Two local models for neural constituent parsing. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 119–132, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Yuanhe Tian, Yan Song, Fei Xia, and Tong Zhang. 2020. Improving constituency parsing with span attention. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1691–1703, Online. Association for Computational Linguistics.

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. TL;DR: Mining Reddit to Learn Automatic Summarization. In *Workshop on New Frontiers in Summarization at EMNLP 2017*, pages 59–63. Association for Computational Linguistics.

Xinyi Wang, Hieu Pham, Pengcheng Yin, and Graham Neubig. 2018. A tree-based decoder for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4772–4777, Brussels, Belgium. Association for Computational Linguistics.

Jiacheng Xu and Greg Durrett. 2019. Neural extractive text summarization with syntactic compression. 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 3292–3303, Hong Kong, China. Association for Computational Linguistics.

Naiwen Xue, Fei Xia, Fu-dong Chiou, and Marta Palmer. 2005. The penn chinese treebank: Phrase structure annotation of a large corpus. *Nat. Lang. Eng.*, 11(2):207–238.

Min-Ling Zhang and Kun Zhang. 2010. Multi-label learning by exploiting label dependency. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, page 999–1008, New York, NY, USA. Association for Computing Machinery.

Yu Zhang, Zhenghua Li, and Min Zhang. 2020a. Efficient second-order TreeCRF for neural dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3295–3305, Online. Association for Computational Linguistics.

Yu Zhang, Houquan Zhou, and Zhenghua Li. 2020b. Fast and accurate neural crf constituency parsing. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4046–4053. International Joint Conferences on Artificial Intelligence Organization.

Yue Zhang and Stephen Clark. 2009. Transition-based parsing of the Chinese treebank using a global discriminative model. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT'09)*, pages 162–171, Paris, France. Association for Computational Linguistics.

Yue Zhang and Joakim Nivre. 2012. Analyzing the effect of global learning and beam-search on transition-based dependency parsing. In *Proceedings of COLING 2012: Posters*, pages 1391–1400, Mumbai, India. The COLING 2012 Organizing Committee.

Jie Zhou and Wei Xu. 2015. End-to-end learning of semantic role labeling using recurrent neural networks. In *Proceedings of the 53rd Annual Meeting of the*

Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 1127–1137, Beijing, China. Association for Computational Linguistics.

Junru Zhou and Hai Zhao. 2019. Head-Driven Phrase Structure Grammar parsing on Penn Treebank. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2396–2408, Florence, Italy. Association for Computational Linguistics.