Structured Mean-Field Variational Inference for Higher-Order Span-Based Semantic Role Labeling

Wei Liu, Songlin Yang, Kewei Tu*
School of Information Science and Technology, ShanghaiTech University
Shanghai Engineering Research Center of Intelligent Vision and Imaging
`{liuwei4, yangsl,tukw}@shanghaitech.edu.cn`

Abstract

In this work, we enhance higher-order graph-based approaches for span-based semantic role labeling (SRL) by means of structured modeling. To decrease the complexity of higher-order modeling, we decompose the edge from predicate word to argument span into three different edges, predicate-to-head (P2H), predicate-to-tail (P2T), and head-to-tail (H2T), where head/tail means the first/last word of the semantic argument span. As such, we use a CRF-based higher-order dependency parser and leverage Mean-Field Variational Inference (MFVI) for higher-order inference. Moreover, since semantic arguments of predicates are often constituents within a constituency parse tree, we can leverage such nice structural property by defining a TreeCRF distribution over all H2T edges, using the idea of partial marginalization to define structural training loss. We further leverage structured MFVI to enhance inference. We experiment on span-based SRL benchmarks, showing the effectiveness of both higher-order and structured modeling and the combination thereof. In addition, we show superior performance of structured MFVI against vanilla MFVI. Our code is publicly available at `https://github.com/VPeterV/Structured-MFVI`.

1 Introduction

Semantic role labeling (SRL) aims to recognize the predicate-argument structures for a given sentence. SRL structures have found various applications in downstream natural language understanding tasks, e.g., machine translation (Marcheggiani et al., 2018), question answering (Khashabi et al., 2018), machine reading comprehension (Zhang et al., 2020c).

There are two types of formalisms in SRL, namely dependency-based and span-based SRL, where the argument is a word in the former case and a contiguous sequence of words (i.e., a span) in the latter case. Span-based SRL is more difficult as it needs to identify two boundaries of a span instead of an argument word, resulting in a much larger search space. We focus on span-based SRL in this work.

Span-based SRL is traditionally tackled by BIO-based sequence labeling approaches (Zhou and Xu, 2015). Later, researchers turn to graph-based methods (He et al., 2018; Ouchi et al., 2018; Li et al., 2019) wherein graph nodes are argument spans and predicate words. Recently, researchers show that higher-order graph-based methods achieve state-of-the-art performance (Jia et al., 2022; Zhou et al., 2022; Zhang et al., 2022). For higher-order graph-based methods, the main difficulty is that there are in total $O(n^3)$ predicate-argument pairs and thereby $O(n^5)$ second-order parts (Jia et al., 2022), making them computationally infeasible to model. To resolve this issue, Jia et al. (2022) prune the number of candidate argument spans from $O(n^2)$ to $O(n)$, and consequently, reduce the number of second-order parts from $O(n^5)$ to $O(n^3)$. On the other hand, Zhou et al. (2022) decompose the original edge (between the predicate word and the argument span) into two word-to-word edges, namely predicate-to-head (predicate word to the first word of argument span, P2H) and predicate-to-tail (predicate word to the last word of argument span, P2T), so the total number of second-order parts reduces from $O(n^5)$ to $O(n^3)$ as well. Both of these two works use Conditional Random Fields (CRF) for probabilistic modeling and Mean-Field Variational Inference (MFVI) for higher-order statistical inference in cubic time. Without MFVI, exact higher-order inference with CRF is NP-hard. Moreover, MFVI is fully differentiable and thus can be incorporated into neural networks as an RNN layer (Zheng et al., 2015) for end-to-end training. Hence, MFVI becomes increasingly popular in solving NLP structured prediction tasks

*Corresponding author

together with higher-order CRF-based modeling (Wang et al., 2019; Wang and Tu, 2020; Zhou et al., 2022).

Besides higher-order modeling, structured modeling has also been shown to be useful in span-based SRL (Zhang et al., 2021; Liu et al., 2022). Span-based SRL has a nice *structural property* that argument spans would not cross to each other in general [1], since gold annotations of argument spans are mostly extracted from existing constituency parse trees. As such, we can build a *partially-observed* constituency parse tree (Fu et al., 2021) wherein observed nodes correspond to gold argument spans. Notably, this is also the case for nested named entity recognition (Fu et al., 2021; Lou et al., 2022) and coreference resolution (Liu et al., 2022). To leverage such structural information (for free) while eliminating the need of obtaining full constituency parse trees (which could be expensive), prior works perform *latent-variable* probabilistic modeling with *partial marginalization* based on dynamic programming (i.e., the inside or CKY algorithm for full constituency parsing).

Concretely, they train a span-based TreeCRF model (Zhang et al., 2020b), either maximizing the probabilities of all compatible trees (to the set of observed arguments or entity spans) via the *masked inside algorithm* (Fu et al., 2021; Lou et al., 2022) or defining training loss based on span marginal probabilities (Liu et al., 2022). These works show that structured modeling indeed improves performance for aforementioned tasks.

Our desiderata in this work is to combine the best of two worlds, performing joint higher-order and structured modeling in a probabilistically principled manner under the CRF framework. To decrease the high complexity of higher-order inference, we use a strategy similar to Zhou et al. (2022) and introduce an additional type of edges for modeling argument spans, namely head-to-tail (the first word to the last word of the argument span, H2T). Without H2T edges, there could be potential ambiguities in the decoding process. More importantly, H2T edges are the bridge for structured modeling, on which we define a span-based TreeCRF distribution. To combine higher-order and structured modeling, inspired by (Domke, 2011; Blondel et al., 2020), we perform MFVI for several steps to obtain approximated marginals, on which we define structured loss for the argument span parts. However, (vanilla) MFVI uses fully-factorized distributions to approximate the otherwise complex true posterior, damaging the quality of higher-order inference. To solve this issue, we further adopt structured MFVI (Wainwright and Jordan, 2008b) to enhance inference, leveraging the underlying tree structures of argument spans for more delicate structured modeling.

We experiment on two benchmarks of span-based SRL: ConLL05 and ConLL12, obtaining state-of-the-art performances on five out of six evaluation metrics. Ablation studies confirm the effectiveness of both higher-order and structured modeling, their combination thereof, and the use of structured MFVI.

[1] However, there could still be a very few number of counterexamples. See (Liu et al., 2022).

2 Method

2.1 Graph encoding and decoding

For each edge connecting a predicate-argument pair, we decompose it into three edges: a P2H edge from predicate to the first word of argument span, a P2T edge from predicate to the last word of argument span, and a H2T edge from the first word to the last word of argument span. Fig. 1 shows an example. After transformation, we build a large graph consisting of three subgraphs, and adopt a two-stage strategy for decoding. In the first stage, we predict unlabeled dependency edges, and then find out all predicate-argument pairs whose corresponding three types of edges are all correctly predicted. As such, our model does not have ambiguity problems in the decoding process, while Zhou et al. (2022) need to propose another constrained Viterbi algorithm to resolve such ambiguities, which is unnecessary when H2T edges are incorporated (Wang et al., 2020). In the second stage, we predict the corresponding label of predicted pairs based on the representations of predicate and argument span.

2.2 Higher-order Modeling

2.2.1 Scoring

For a sentence of length n, we use three indicator matrices (whose entries are either 0 or 1) $y^H, y^T, y^A \in \mathbb{R}^{n\times n}$ to represent P2H, P2T, and H2T edges, respectively. For example, $y^H_{ij} = 1$ iff there is an P2H edge (i, j), and $y^H_{ij} = 0$ otherwise. We use $y = [y^H; y^T, y^A] \in \mathbb{R}^{n\times 3n}$ to represent the entire (multi)graph.

We first define the first-order edge-factorized

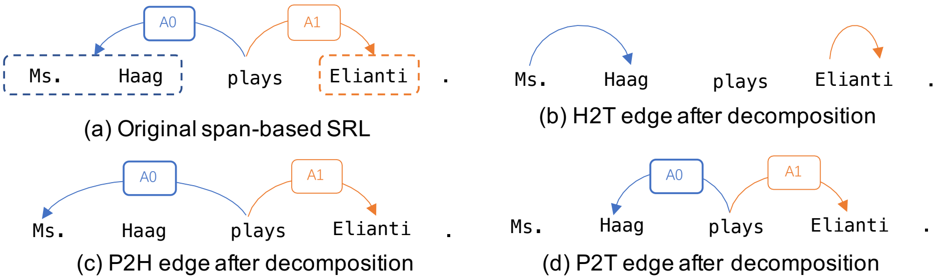


Figure 1: An example of span-based SRL. We transform predicate-argument pairs into three different types of edges, casting span-based SRL as a dependency graph parsing problem. Figure (a) is original predicate-argument pairs, where phrases or words included inside boxes with dash line are argument spans. Figure (b)-(d) are corresponding edges after decomposition.

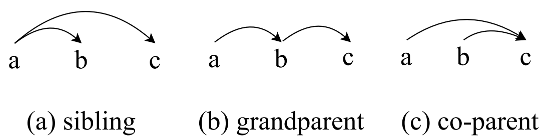


Figure 2: Three different higher-order relationships.

score for y:

$$s^{1o}(y) = \sum_{ij}(s^H_{ij}y^H_{ij} + s^T_{ij}y^T_{ij} + s^A_{ij}y^A_{ij})$$

Then we consider the following higher-order scores based on sibling (sib), co-parent (cop), and grandparent (gp) relationships (Fig. 2):

- $s^{h,sib}_{ij,ik}$, $s^{h,cop}_{ik,jk}$: sibling and co-parent scores between two P2H edges.
- $s^{t,sib}_{ij,ik}$, $s^{t,cop}_{ik,jk}$: sibling and co-parent scores between two P2T edges.
- $s^{a,gp}_{ij,jk}$, $s^{a,cop}_{ik,jk}$: scores between a P2H or a P2T edge and a H2T edge.

For example, $s^{a,gp}_{ij,jk}$ measures how likely a P2H edge (i, j) and a H2T edge (j, k) coexist. Since $i \rightarrow j \rightarrow k$ forms a grandparent relationship, we mark the score with a *gp* suffix.

The total second-order scores [2] for each type are:

$$s^{2o,h}(y) = \frac{1}{2}(\sum_{ijk} s^{h,sib}_{ij,ik}y^H_{ij}y^H_{ik} + \sum_{ijk} s^{h,cop}_{ik,jk}y^H_{ik}y^H_{jk})$$

$$s^{2o,t}(y) = \frac{1}{2}(\sum_{ijk} s^{t,sib}_{ij,ik}y^T_{ij}y^T_{ik} + \sum_{ijk} s^{t,cop}_{ik,jk}y^T_{ik}y^T_{jk})$$

$$s^{2o,a}(y) = \sum_{ijk} s^{a,gp}_{ij,jk}y^H_{ij}y^A_{jk} + \sum_{ijk} s^{a,cop}_{ik,jk}y^T_{ik}y^A_{jk}$$

[2] We force higher-order scores to be symmetric, e.g., $s^{t,sib}_{ij,ik} = s^{t,sib}_{ik,ij}$, using the strategy of (Wang et al., 2019). We multiple $\frac{1}{2}$ for $s^{2o,h}$, $s^{2o,t}$ because sibling and co-parent scores are symmetric w.r.t. $y^{H/T}$ and thus are computed twice.

Finally, the score of y is the sum of the first-order score and all the higher-order scores:

$$s(y) = s^{1o}(y) + s^{2o,h}(y) + s^{2o,t}(y) + s^{2o,a}(y)$$

2.2.2 CRF and MFVI

We define a conditional random field (CRF) over all possible y:

$$p(y) = \frac{\exp(s(y))}{Z}$$

where Z is the partition function. Since Z is intractable to compute, we resort to MFVI to generate lower bounds of Z and thus obtain approximations to the true marginals (Wainwright and Jordan, 2008b), and then define the loss in terms of the approximated marginals (posteriors) (Domke, 2011).

MFVI uses simple and tractable posterior distribution family [3] $\{p_{\theta_0}\}_{\theta_0}$ to approximate the true posterior. There is a one-to-one correspondence between an instantiation p_{θ_0} and a mean-vector (i.e., marginal) μ_0 (Wainwright and Jordan, 2008b, Prop. 3.2), and we denote the set of all realizable mean-vectors as $\mathcal{M}$, i.e., the marginal polytope. Wainwright and Jordan (2008b); Lê-Huu and Karteek (2021) show that MFVI update is equal to the following variational representation:

$$y^{(m+1)} = \arg\max_{y \in \mathcal{M}} \langle Q^{(m)}, y\rangle - A^{\star}_{\mathcal{M}}(y) \quad (1)$$

where m is the iteration number; $Q^{(m)} := \nabla s(y^{(m)})$ is the gradient of $s(y^{(m)})$ w.r.t. $y^{(m)}$; $\langle \cdot \rangle$ is inner product; $A^{\star}_{\mathcal{M}}$ is the conjugate dual function satisfying that:

$$A^{\star}_{\mathcal{M}}(y) = -H(p_{\theta}(\cdot))$$

[3] We assume it is parameterized as a minimal exponential family.

for some p_{θ_0} coupled to y (Wainwright and Jordan, 2008b, Thm. 3.4) and H denotes the entropy thereof.

Vanilla mean-field uses a fully-factorized posterior distribution (i.e., product of Bernoulli distribution) to approximate the true posterior distribution. Therefore, in this case $\mathcal{M} = [0,1]^{n \times 3n}$, and

$$A^{\star}_{\mathcal{M}}(y) = \sum_{ij} y_{ij} \log y_{ij} + (1 - y_{ij}) \log(1 - y_{ij})$$

Then Eq. 1 is the variational representation of sigmoid function (Wainwright and Jordan, 2008b, Example 5.2) and thus the solution is attained at:

$$\begin{aligned} y_{ij}^{(m+1)} &= \frac{\exp\{Q_{ij}^{(m)}\}}{\exp\{Q_{ij}^{(m)}\} + 1} \\ &= \operatorname{sigmoid}(Q_{ij}^{(m)}) \end{aligned} \tag{2}$$

Recall that $Q^{(m)} = [Q^{H(m)}; Q^{T(m)}; Q^{A(m)}] = \nabla s(y^{(m)})$, we have:

$$\begin{aligned} Q_{ij}^{H(m)} &= s_{ij}^{H} + \sum_k (y_{jk}^{A(m)} s_{ij,jk}^{a,gp} + \\ & y_{ik}^{H(m)} s_{ij,ik}^{h,sib} + y_{kj}^{H(m)} s_{ij,kj}^{h,cop}) \end{aligned} \tag{3}$$

$$\begin{aligned} Q_{ij}^{T(m)} &= s_{ij}^{T} + \sum_k (y_{kj}^{A(m)} s_{ij,kj}^{a,cop} + \\ & y_{ik}^{T(m)} s_{ij,ik}^{t,sib} + y_{kj}^{T(m)} s_{ij,kj}^{t,cop}) \end{aligned} \tag{4}$$

$$\begin{aligned} Q_{ij}^{A(m)} &= s_{ij}^{A} + \sum_k (y_{ki}^{H(m)} s_{ki,ij}^{a,gp} + \\ & y_{kj}^{T(m)} s_{ij,kj}^{a,cop}) \end{aligned} \tag{5}$$

We use $Q^{(0)} := [s^H; s^T; s^A]$ for initialization. Then MFVI performs Eq. 2 (posterior update) and Eq. 3-5 (score aggregation) alternately in each iteration. Note that these steps are fully differentiable, so one can unroll several inference steps for end-to-end learning (Domke, 2011).

2.3 Structured Modeling

A key observation provided by Liu et al. (2022) is that semantic-argument spans are often constituents in a constituency tree. It is thus beneficial to model the underlying partially-observed constituency tree (Fu et al., 2021), in which the observed nodes correspond to gold semantic arguments. We follow Lou et al. (2022) to use a 0-1 labeling strategy, i.e., assigning label 1 to the observed parts and 0 to the unobserved parts of a partially-observed tree t, and use an order-3 *binary* tensor $T \in \mathbb{R}^{n \times n \times 2}$ to represent t where $T_{ijk} = 1$ iff there is a span from x_i to x_j with label $k \in \{0, 1\}$ in t. Then we define the score as:

$$s(T) = \sum_{ijk} T_{ijk} s_{ijk}$$

where $s \in \mathbb{R}^{n \times n \times 2}$ is all span scores. Denote the set of gold unlabeled semantic argument spans as $\boldsymbol{y} = \{(i,j) \cdots\}$, and the set of compatible tree indicators as $\tilde{T}(\boldsymbol{y})$. We say $T \in \tilde{T}(\boldsymbol{y})$ iff $T_{ij1} = 1$ for all $(i,j) \in \boldsymbol{y}$, $T_{ij0} = 1$ for all rest spans $(i,j) \in t; (i,j) \notin \boldsymbol{y}$, and $T_{ijk} = 0$ otherwise. Partially-observed TreeCRF (PO-TreeCRF) (Fu et al., 2021) aims to maximize the log-likelihood of all compatible trees:

$$s(\boldsymbol{y}) = \log \sum_{T \in \tilde{T}(\boldsymbol{y})} \exp(s(T)) \tag{6}$$

$$\log p(\boldsymbol{y}) = s(\boldsymbol{y}) - \log Z \tag{7}$$

where $\log Z$ is the log-partition function which can be computed via the inside algorithm. $s(\boldsymbol{y})$ can be computed efficiently via the masked inside algorithm (Fu et al., 2021; Lou et al., 2022), where all incompatible span nodes crossing any span in $\boldsymbol{y}$ are masked (i.e., set to negative infinity in log-domain) before running the inside algorithm. See (Fu et al., 2021) for more details.

2.4 Joint Higher-order and Structured Modeling

We can simply combines (vanilla) MFVI with PO-TreeCFR to achieve joint higher-order and structured modeling as follows.

After running k iterations of MFVI, we obtain a set of un-normalized scores $Q^{(k)}$ and approximated marginals $y^{(k+1)}$, on which our loss is based. It is worth mentioning that designing the loss by means of $Q^{(k)}$ in many cases is equivalent to designing the loss by means of $y^{(k+1)}$ (Blondel et al., 2020). , so we essentially design the loss in terms of approximated marginals produced by truncated MFVI (Domke, 2012).

For H2T edges, we feed un-normalized score $[Q^{A(k)}; s^B]$ as span score into a PO-TreeCRF to compute log-likelihood of all compatible trees (Eq. 7), then taking the negative to define the loss:

$$L^A = -\log p(\boldsymbol{y}). \tag{8}$$

where $\boldsymbol{y}$ is the set of gold unlabeled argument spans.

For P2H and P2T edges, we use the binary cross-entropy loss:

$$L^{H/T} = -\sum_{ij}\Big(\hat{y}_{ij}^{H/T}\log y_{ij}^{H/T(k+1)} + (1-\hat{y}_{ij}^{H/T})\log(1-y_{ij}^{H/T(k+1)})\Big) \tag{9}$$

where $\hat{y}_{ij}^{H/T} \in \{0,1\}$ indicates the existence of P2H/P2T edge (i,j).

2.5 Structured MFVI

Vanilla MFVI uses a fully-factorized distribution to approximate the true posterior, ignoring the inherent tree structures in span-based SRL. To better leverage the inherent tree structures, we propose to adopt structured MFVI (Saul and Jordan, 1995; Wainwright and Jordan, 2008b; Burkett et al., 2010), using TreeCRFs (Zhang et al., 2020b) — instead of product of Bernoulli distribution as used in vanilla MFVI—to parameterize the posterior distribution regarding H2T edges.

To deal with 0-1 labeled constituency trees, we let y^A corresponding to label-1 spans, and use an auxiliary $y^B \in \mathbb{R}^{n\times n}$ to represent label-0 spans with first-order scores $s^B \in \mathbb{R}^{n\times n}$. We denote $z := [y^B; y^A]$ and use a TreeCRF to parameterize their posterior distribution. Then the posterior update of z is:

$$z^{(m+1)} = \arg\max_{z\in\mathcal{T}}\langle F^{(m)}, z\rangle - A^{\star}_{\mathcal{T}}(z) \tag{10}$$

where $\mathcal{T}$ is the *structured* marginal polytope of 0-1 labeled binary trees (Rush et al., 2010; Martins and Filipe, 2012), $A^{\star}_{\mathcal{T}}(z)$ equals to the negative entropy of the TreeCRF distribution p_{θ_0} for some θ_0 coupled to z (Martins et al., 2010, Prop. 1); $F^{(m)}_{i,j} = [Q^{B(m)}_{ij}; Q^{A(m)}_{ij}]$.

The solution of Eq. 10 is attained at the mean-vector regarding the TreeCRF distribution (Wainwright and Jordan, 2008b; Paulus et al., 2020), i.e., span marginals, which can be computed efficiently by back-propagating through the inside algorithm (Eisner, 2016; Rush, 2020). Since there are no higher-order scores associated with y^B, we have $Q^{B(m)}_{ij} = s^B_{ij}$ and Eq. 3-5 remain intact. Besides, since we do not couple y^B, y^A with y^H, y^T, the posterior update of y^H, y^T remains the same.

As such, the posterior update of y^A is structure-aware, well-respecting the constituency tree constraint. The tree-structured information is propagated from y^A to y^H, y^T through Eq. 3-4 thanks to the higher-order factors $s^{a,cop}$, $s^{a,gp}$ connecting them.

3 Model Architecture

We depict our model architecture in Fig. 3.

Encoding. Given the sentence $\boldsymbol{x} = \{x_0, x_1, ..., x_n\}$, we feed it into BERT (Devlin et al., 2019) and apply mean-pooling to the last four layers to obtain *word-level* representations $\boldsymbol{h} = \{h_0, h_1, ..., h_n\}$. If we use pre-identified predicates, we concatenate h with an indicator embedding additionally.

First-order scores. We use deep biaffine attention (Dozat and Manning, 2017) to compute s^H, s^T and s^A:

$$\boldsymbol{r}_i^{p/h/t} = \mathrm{MLP}^{p/h/t}(\boldsymbol{h}_i)$$

$$s_{ij}^{H/T/A} = \begin{bmatrix}\boldsymbol{r}_i^{p/p/h}\\ 1\end{bmatrix}^{\top}\boldsymbol{W}^{H/T/A}\begin{bmatrix}\boldsymbol{r}_j^{h/t/t}\\ 1\end{bmatrix}$$

where $\boldsymbol{r}^{p/h/t}$ are type-specific representations for predicates and head/tail words of argument spans, respectively; $\mathrm{MLP}^{p/h/t}$ are multi-layer perceptrons which transform $\boldsymbol{h}_i$ to d-dimensional spaces; $\boldsymbol{W}^{\mathrm{H/T/A}} \in \mathbb{R}^{(d+1)\times(d+1)}$ are trainable parameters.

Higher-order scores. We use deep Triaffine attention (Wang et al., 2019; Zhang et al., 2020a) to compute higher-order scores:

$$\begin{aligned}\hat{\boldsymbol{r}}_i^{p/h/t} &= \hat{\mathrm{MLP}}^{p/h/t}(\boldsymbol{h}_i)\\ s^{a,gp}_{ij,jk}/s^{a,cop}_{ik,jk} &= \mathrm{TriAFF}^{gp/cop1}(\hat{\boldsymbol{r}}_i^p, \hat{\boldsymbol{r}}_j^h, \hat{\boldsymbol{r}}_k^t)\\ s^{h,sib}_{ij,ik}/s^{h,cop}_{ik,jk} &= \mathrm{TriAFF}^{sib1/cop2}(\hat{\boldsymbol{r}}_i^p, \hat{\boldsymbol{r}}_j^{h/p}, \hat{\boldsymbol{r}}_k^h)\\ s^{t,sib}_{ij,ik}/s^{t,cop}_{ik,jk} &= \mathrm{TriAFF}^{sib2/cop3}(\hat{\boldsymbol{r}}_i^p, \hat{\boldsymbol{r}}_j^{t/p}, \hat{\boldsymbol{r}}_k^t)\end{aligned}$$

where

$$\mathrm{TriAFF}(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3) = \begin{bmatrix}\boldsymbol{v}_3\\ 1\end{bmatrix}^{\top}\boldsymbol{v}_1^{\intercal}\boldsymbol{W}'\begin{bmatrix}\boldsymbol{v}_2\\ 1\end{bmatrix}$$

with $\boldsymbol{W}' \in \mathbb{R}^{(d+1)\times(d)\times(d+1)}$.

Label Scores and Label Loss. Following Jia et al. (2022), we use Coherent (Seo et al., 2019) span representation to compute the label scores. Given an argument span $\overline{a}_{ij} = (w_i, ..., w_j)$ obtained by first-stage, we encode the two endpoints w_i, w_j as $\boldsymbol{g}_i, \boldsymbol{g}_j \in \mathbb{R}^r$. We split each $\boldsymbol{g}_k$ into four parts: $\boldsymbol{g}_k = [\boldsymbol{g}_k^1; \boldsymbol{g}_k^2; \boldsymbol{g}_k^3; \boldsymbol{g}_k^4]$, where $\boldsymbol{g}_k^1, \boldsymbol{g}_k^2 \in$

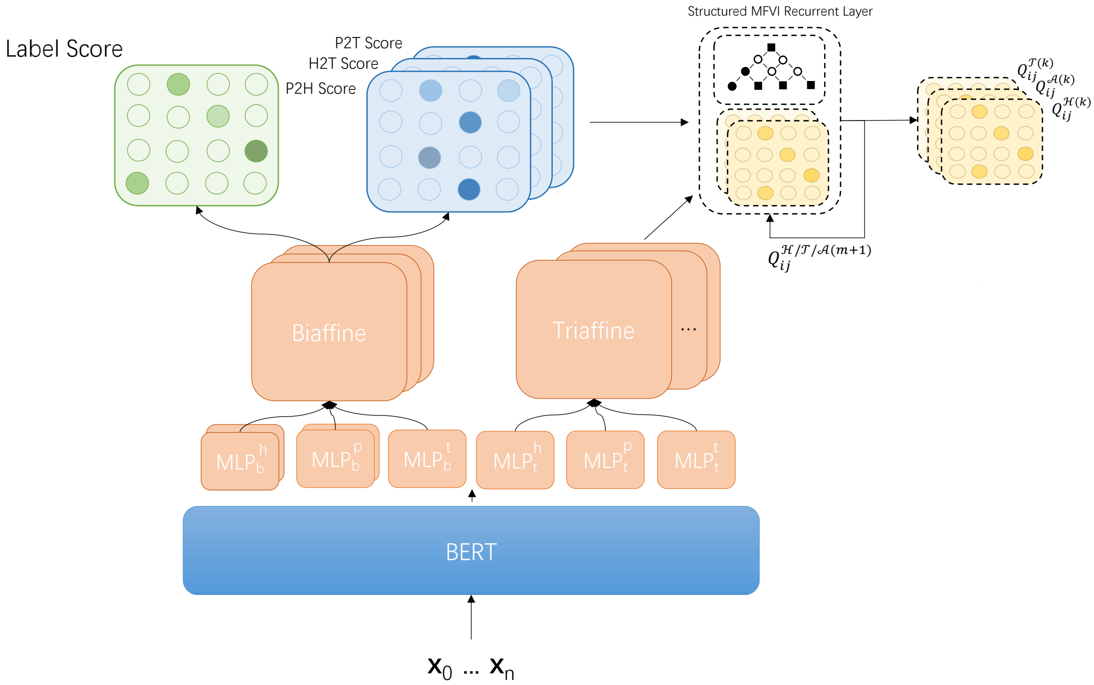


Figure 3: Illustration of our model.

$\mathbb{R}^a, \boldsymbol{g}_k^3, \boldsymbol{g}_k^4 \in \mathbb{R}^b$ and $2(a+b) = r$. Then we can represent span as:

$$\boldsymbol{a} = [\boldsymbol{g}_i^1; \boldsymbol{g}_j^2; \boldsymbol{g}_i^3 \cdot \boldsymbol{g}_j^4]$$

where dot product $\boldsymbol{g}_i^3 \cdot \boldsymbol{g}_j^4$ is called coherence term. Then we use biaffine attention to compute label score s_{ijkl}^{label}:

$$s_{ijkl}^{label} = \begin{bmatrix} \boldsymbol{r}_i^p \\ 1 \end{bmatrix}^\top \boldsymbol{W}_l^{label} \begin{bmatrix} \boldsymbol{a}_{jk} \\ 1 \end{bmatrix}$$

We use cross-entropy to compute corresponding label loss,

$$L_{label} = -\sum_{ijk} \mathbf{1}(\hat{y}_{ijk}) \log \frac{\exp(s_{ijkl_{ijk}}^{label})}{\sum_l \exp(s_{ijkl}^{label})} \quad (11)$$

where $\hat{y}_{ijk} \in \{0,1\}$ indicates the existence of predicate-argument pairs. l_{ijk} is the gold label for pair of the predicate-argument pair (i, jk).

Total Training Loss We optimize the weighted average of the above losses according to Eq 8 9 11.

$$L = \lambda_1 L_{label} + (1-\lambda_1) L_{edge}$$
$$L_{edge} = \lambda_2 L^A + (1-\lambda_2)(L^H + L^T)$$

where λ_1 and λ_2 are hyper-parameters.

4 Experiments

Settings. Following previous works, we conduct experiments on two benchmarks: CoNLL05 (Palmer et al., 2005) and CoNLL12 (Pradhan et al., 2012) English datasets, where CoNLL05 include two test datasets WSJ (in-domain) and BROWN (out-of-domain). We adopt official data splits and evaluate our model using the official evaluation script [4], reporting the micro-average F1 score averaged over three different runs with different random seeds. We conduct experiments under two settings, i.e., *with (w/) gold predicates* and *without (w/o) gold predicate*. Following most previous works, we use `Bert-large-cased` (Devlin et al., 2019) as the backbone. We refer readers to Appendix A for our implementation details.

Main Results. Table 1 shows the main results on test sets of benchmarks. Our baseline model is *1O* trained with local binary cross-entropy loss for all three types of edges without higher-order and structured modeling. Our proposed model clearly outperforms the baseline, obtaining state-of-the-art performances (when using `Bert-large-cased`) on five out of six evaluation metrics.

[4] https://www.cs.upc.edu/~srlconll/soft.html#srlconll

	CoNLL05-WSJ			CoNLL05-Brown			CoNLL12		
	P	R	F1	P	R	F1	P	R	F1
w/o gold predicates									
He et al. (2017)	80.20	82.30	81.20	67.60	69.60	68.50	78.60	75.10	76.80
He et al. (2018) + ELMO	84.80	87.20	86.0	73.90	78.40	76.10	81.90	84.00	82.90
Jia et al. (2022) + BERT	–	–	86.70	–	–	78.58	–	–	84.22
Zhou et al. (2022) + BERT	87.15	88.44	87.79	79.44	80.85	80.14	83.91	85.61	84.75
Zhang et al. (2022) + BERT	87.00	**88.76**	87.87	79.08	81.50	80.27	84.53	**86.41**	**85.45**
1O + BERT	87.11	87.40	87.25	79.89	79.93	79.91	84.76	84.42	84.59
Ours + BERT	**88.05**	88.61	**88.33**	**81.13**	**81.58**	**81.36**	**84.95**	85.85	85.40
w/ gold predicates									
He et al. (2017)	85.00	84.30	84.60	74.90	72.40	73.60	83.50	83.30	83.40
He et al. (2018) + ELMO	–	–	87.40	–	–	80.40	–	–	85.50
Shi and Lin (2019) + BERT	88.60	**89.00**	88.80	81.90	**82.10**	82.00	85.90	87.00	86.50
Conia and Navigli (2020) + BERT	–	–	–	–	–	–	86.90	87.70	87.30
Blloshmi et al. (2021) + BART	–	–	–	–	–	–	87.80	86.80	87.30
Liu et al. (2022) + SpanBERT	–	–	–	–	–	–	–	–	87.50
Jia et al. (2022) + BERT	–	–	88.25	–	–	81.90	–	–	87.18
Zhou et al. (2022) + BERT	89.03	88.53	88.78	83.22	81.81	82.51	87.26	87.05	87.15
Zhang et al. (2022) + BERT	89.00	**89.03**	89.02	82.81	82.35	82.58	87.52	**87.79**	87.66
1O + BERT	89.09	87.57	88.32	83.30	79.49	81.35	87.45	86.75	87.10
Ours + BERT	**89.77**	88.46	**89.11**	**83.96**	**81.76**	**82.85**	**88.10**	87.38	**87.74**

Table 1: Comparison of our model and other models on test sets of CoNLL05-WSJ, CoNLL05-Brown, and CoNLL12.

Model	P	R	F1
Unstructured(1O)	87.11	87.40	87.25
Unstructured(2O)	87.21	88.34	87.77
$1O_{+\text{TreeCRF}}$	87.79	87.57	87.68
$2O_{\text{VMF+TreeCRF}}$	87.53	88.26	87.90
$2O_{\text{SMF+TreeCRF}}$ (Final)	**88.05**	**88.61**	**88.33**

Table 2: Ablation studies on CoNLL05-WSJ dataset. VMF indicates vanilla mean-field and SMF indicates structured mean-filed.

Ablation studies. To better understand the source of improvement, we conduct ablation studies on CoNLL05-WSJ test set. Table 2 shows the results. As we can see, compared with *1O*, using higher-order inference alone leads to 0.52 F1 score improvement; using PO-TreeCRF structured loss alone leads to 0.43 F1 score improvement, proving the effectiveness of both higher-order and structured modeling. When combining vanilla mean-field-based higher-order inference and structured loss, we have 0.65 F1 score improvement compared to *1O*, showing that it is beneficial to combine both higher-order and structured modeling. We then replace the vanilla mean-field with structured mean-field, resulting in further improvement of 0.43 F1 score, showing the effectiveness of structured MFVI.

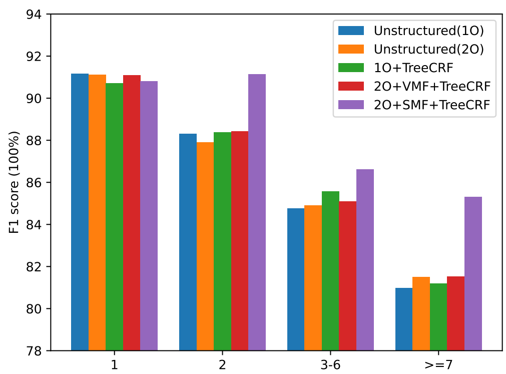


Figure 4: F1 score regarding to different argument span length. The x-axis denotes the length of argument spans. The y-axis denotes the F1 score.

F1 against argument span length. Fig. 4 shows the F1 scores with the change of argument span length. As we can see, our full model performs the best when the span length is large, especially when > 7. We hypothesis that this is due to that in structured mean-filed inference, the *global* tree structure information is propagated among variables.

5 Related Work

In recent years, graph-based (or span-based) models become popular in span-based SRL thanks to their ability in encoding rich span features. Ouchi et al. (2018) exhaustively search predicate-argument pairs. He et al. (2018) use a pruning strategy to reduce the search complexity. They then use a neural network to predicate the relationship between candidate predicates and candidate argument spans. Li et al. (2019) extend their work by using deep biaffine attention (Dozat and Manning, 2017) for scoring, and tackling both span-based and dependency-based SRL under a single unified framework. He et al. (2019) prune argument spans via syntactic rules for multilingual SRL. Zhang et al. (2021) point out that the way to extract spans has a huge impact on the final performance. Instead of taking top-k candidate spans (i.e., beam pruning) as in He et al. (2018), they use a two-stage strategy where the first stage finds all headwords, and the second stage predicates span boundaries based on predicted headwords. They use either gold heads from dependency-SRL annotations or automatically-learned heads by using the "bag loss" proposed in Lin et al. (2019). They show their two-stage strategy is better than beam pruning in different settings.

Thanks to the advance in second-order semantic dependency parsing (Wang et al., 2019) where they unroll several mean-field inference steps for end-to-end training, researchers adopt this technique to improvement the performance of span-based SRL. Direct second-order modeling leads to a $O(n^5)$ search space, which is computationally prohibitive. Jia et al. (2022) thus use a beam pruning strategy to select $O(n)$ candidate spans to decrease the complexity of second-order inference. Zhou et al. (2022) decompose predicate-argument pairs into dependency edges. By doing so, they cast span-based SRL to a dependency graph parsing technique, and thus can directly use the method of Wang et al. (2019) without much adaptation. Since there are total $O(n^2)$ edges, there is no need for pruning as exhaustive search is relative cheap.

Semantic arguments are often constituents. This is very similar to the case in nested named entity recognition (NER) where named entities are mainly extracted from constituency trees; and in coreference resolution where mentions are often constituents. This means that, one can embed these named entities or semantic arguments or mentions into constituency trees for structured modeling. Finkel and Manning (2009) use a constituency parser to jointly model constituents and named entities, however their approach needs tree annotations, which are difficult to obtain. To resolve this problem, Fu et al. (2021); Lou et al. (2022) view named entities as partially-observed constituency trees, and design masked inside algorithms for partial marginalization to train their TreeCRF models. Liu et al. (2022) propose structured span selectors for span-based SRL and coreference resolution, training weighted context-free grammars (or essentially, TreeCRFs) by partial marginalization akin to Fu et al. (2021); Lou et al. (2022). They leverage the CYK algorithm to produce $O(n)$ structure-aware candidate spans, outperforming the beam pruning strategy.

Structured mean-field variational inference is well-studied in the literature of graphical models (Wainwright and Jordan, 2008a), but we only find few applications in the NLP community, e.g. in Burkett et al. (2010). We believe structured mean-field variational inference can be used more frequently and in this work we demonstrate its usage in span-based SRL.

6 Conclusion

In this work, we tackled span-based SRL using a graph-based approach, combining the advantage of higher-order and structured modeling. In addition, we leveraged structured MFVI to respect the constituency tree constraint of argument spans during inference. We showed the effectiveness of these components experimentally.

Limitations

The main concern regarding our model is the computational complexity. higher-order MFVI has a complexity of $O(n^3)$, which admits fully parallel computation and thus is fast on GPUs. The complexity of structured inference of TreeCRF is also $O(n^3)$. However, due to the dynamic programming computation restriction, only $O(n^2)$ out of $O(n^3)$ can be computed in parallel using parallel parsing techniques (Rush, 2020), slowing down the running speed. Besides, differentiating through the TreeCRF marginals needs many GPU memories (Kim et al., 2017), as automatic differentiation saves all intermediate dynamic programming items for back-propagation, which cause plenty of waste of GPU memories. In this work, since the memory problem is not too severe, we use automatic differentiation for simplicity. One solution is to manually implement the outside algorithm to mitigate the memory problem (Kim et al., 2017).

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61976139).

References

Rexhina Blloshmi, Simone Conia, Rocco Tripodi, and Roberto Navigli. 2021. Generating senses and roles: An end-to-end model for dependency- and span-based semantic role labeling. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3786–3793. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Mathieu Blondel, André F. T. Martins, and Vlad Niculae. 2020. Learning with fenchel-young losses. *J. Mach. Learn. Res.*, 21:35:1–35:69.

David Burkett, John Blitzer, and Dan Klein. 2010. Joint parsing and alignment with weakly synchronized grammars. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–135, Los Angeles, California. Association for Computational Linguistics.

Simone Conia and Roberto Navigli. 2020. Bridging the gap in multilingual semantic role labeling: a language-agnostic approach. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1396–1410, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Justin Domke. 2011. Parameter learning with truncated message-passing. *CVPR 2011*, pages 2937–2943.

Justin Domke. 2012. Generic methods for optimization-based modeling. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2012, La Palma, Canary Islands, Spain, April 21-23, 2012*, volume 22 of *JMLR Proceedings*, pages 318–326. JMLR.org.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Jason Eisner. 2016. Inside-outside and forward-backward algorithms are just backprop (tutorial paper). In *Proceedings of the Workshop on Structured Prediction for NLP*, pages 1–17, Austin, TX. Association for Computational Linguistics.

Jenny Rose Finkel and Christopher D. Manning. 2009. Nested named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 141–150, Singapore. Association for Computational Linguistics.

Yao Fu, Chuanqi Tan, Mosha Chen, Songfang Huang, and Fei Huang. 2021. Nested named entity recognition with partially-observed treecrfs. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 12839–12847. AAAI Press.

Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2018. Jointly predicting predicates and arguments in neural semantic role labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 364–369, Melbourne, Australia. Association for Computational Linguistics.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 473–483, Vancouver, Canada. Association for Computational Linguistics.

Shexia He, Zuchao Li, and Hai Zhao. 2019. Syntax-aware multilingual semantic role 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 5350–5359, Hong Kong, China. Association for Computational Linguistics.

Zixia Jia, Zhaohui Yan, Haoyi Wu, and Kewei Tu. 2022. Span-based semantic role labeling with argument pruning and second-order inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI Press.

Daniel Khashabi, Tushar Khot, Ashish Sabharwal, and Dan Roth. 2018. Question answering as global reasoning over semantic abstractions. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1905–1914. AAAI Press.

Yoon Kim, Carl Denton, Luong Hoang, and Alexander M. Rush. 2017. Structured attention networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Đ.Khuê Lê-Huu and Alahari Karteek. 2021. Regularized frank-wolfe for dense crfs: Generalizing mean field and beyond. In *NeurIPS*.

Zuchao Li, Shexia He, Hai Zhao, Yiqing Zhang, Zhuosheng Zhang, Xi Zhou, and Xiang Zhou. 2019. Dependency or span, end-to-end uniform semantic role labeling. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6730–6737. AAAI Press.

Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. 2019. Sequence-to-nuggets: Nested entity mention detection via anchor-region networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5182–5192, Florence, Italy. Association for Computational Linguistics.

Tianyu Liu, Yuchen Jiang, Ryan Cotterell, and Mrinmaya Sachan. 2022. A structured span selector. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2629–2641, Seattle, United States. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Chao Lou, Songlin Yang, and Kewei Tu. 2022. Nested named entity recognition as latent lexicalized constituency parsing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6183–6198, Dublin, Ireland. Association for Computational Linguistics.

Diego Marcheggiani, Jasmijn Bastings, and Ivan Titov. 2018. Exploiting semantics in neural machine translation with graph convolutional networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 486–492, New Orleans, Louisiana. Association for Computational Linguistics.

André Martins, Noah Smith, Eric Xing, Pedro Aguiar, and Mário Figueiredo. 2010. Turbo parsers: Dependency parsing by approximate variational inference. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 34–44, Cambridge, MA. Association for Computational Linguistics.

Torres Martins and André Filipe. 2012. The geometry of constrained structured prediction: Applications to inference and learning of natural language syntax.

Hiroki Ouchi, Hiroyuki Shindo, and Yuji Matsumoto. 2018. A span selection model for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1630–1642, Brussels, Belgium. Association for Computational Linguistics.

Martha Palmer, Paul R. Kingsbury, and Daniel Gildea. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31:71–106.

Max B. Paulus, Dami Choi, Daniel Tarlow, Andreas Krause, and Chris J. Maddison. 2020. Gradient estimation with stochastic softmax tricks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Alexander Rush. 2020. Torch-struct: Deep structured prediction library. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 335–342, Online. Association for Computational Linguistics.

Alexander M. Rush, David Sontag, Michael Collins, and Tommi Jaakkola. 2010. On dual decomposition and linear programming relaxations for natural language processing. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1–11, Cambridge, MA. Association for Computational Linguistics.

Lawrence K. Saul and Michael I. Jordan. 1995. Exploiting tractable substructures in intractable networks. In *Advances in Neural Information Processing Systems 8, NIPS, Denver, CO, USA, November 27-30, 1995*, pages 486–492. MIT Press.

Minjoon Seo, Jinhyuk Lee, Tom Kwiatkowski, Ankur Parikh, Ali Farhadi, and Hannaneh Hajishirzi. 2019. Real-time open-domain question answering with dense-sparse phrase index. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4430–4441, Florence, Italy. Association for Computational Linguistics.

Peng Shi and Jimmy Lin. 2019. Simple BERT models for relation extraction and semantic role labeling. *CoRR*, abs/1904.05255.

Martin J. Wainwright and M.I. Jordan. 2008a. Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.*, 1:1–305.

Martin J. Wainwright and Michael I. Jordan. 2008b. Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.*, 1(1-2):1–305.

Xinyu Wang, Jingxian Huang, and Kewei Tu. 2019. Second-order semantic dependency parsing with end-to-end neural networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4609–4618, Florence, Italy. Association for Computational Linguistics.

Xinyu Wang and Kewei Tu. 2020. Second-order neural dependency parsing with message passing and end-to-end training. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 93–99, Suzhou, China. Association for Computational Linguistics.

Yucheng Wang, Bowen Yu, Yueyang Zhang, Tingwen Liu, Hongsong Zhu, and Limin Sun. 2020. TPLinker: Single-stage joint extraction of entities and relations through token pair linking. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1572–1582, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Biao Zhang, Ivan Titov, and Rico Sennrich. 2020a. Fast interleaved bidirectional sequence generation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 503–515, Online. Association for Computational Linguistics.

Yu Zhang, Qingrong Xia, Shilin Zhou, Yong Jiang, Guohong Fu, and Min Zhang. 2022. Semantic role labeling as dependency parsing: Exploring latent tree structures inside arguments. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4212–4227, Gyeongju, Republic of Korea. International Committee on 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 2020*, pages 4046–4053. ijcai.org.

Zhisong Zhang, Emma Strubell, and Eduard Hovy. 2021. Comparing span extraction methods for semantic role labeling. In *Proceedings of the 5th Workshop on Structured Prediction for NLP (SPNLP 2021)*, pages 67–77, Online. Association for Computational Linguistics.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020c. Semantics-aware BERT for language understanding. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9628–9635. AAAI Press.

Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr. 2015. Conditional random fields as recurrent neural networks. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1529–1537.

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.

Shilin Zhou, Qingrong Xia, Zhenghua Li, Yu Zhang, Yu Hong, and Min Zhang. 2022. Fast and accurate end-to-end span-based semantic role labeling as word-based graph parsing. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4160–4171, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

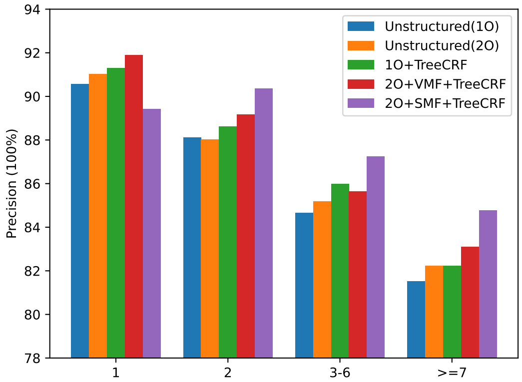
A Implementation Details

We use BERT[5] (`bert-large-cased`) as encoders to obtain word representations. We use deep biaffine attention (Dozat and Manning, 2017) with 500 dimensions and deep triaffine attention with 100 following previous work (Wang et al., 2019). We set iteration number of MFVI as 3. To prevent overfitting, we set dropout ratio 0.1 for encoders and 0.1 for every MLP layers. Regarding training, we set learning rate for encoder layers as $5e-5$ and the rest layers as $1e-3$. We train our model for 10 epochs with max words 1000 using AdamW (Loshchilov and Hutter, 2019) optimizer. We adopt linear warmup scheduler for 10% training steps. Following previous works (Zhou et al., 2022; Fu et al., 2021) ,we set the hyper-parameters λ_1 and λ_2 as 0.06 and 0.1. All experiments run on NVIDIA TITAN RTX and NVIDIA A40 gpus.

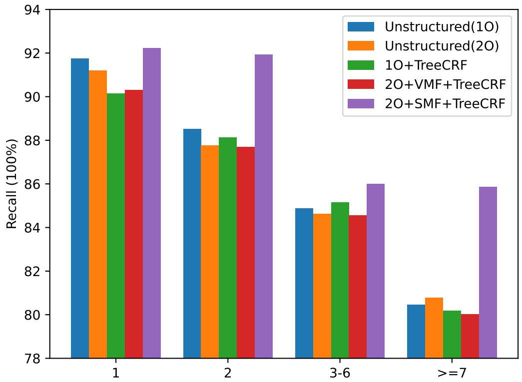
B Recall and Precision Regarding to Argument Width

The corresponding precision and recall of F1 score in Fig 4 with different argument span length are shown as Fig 5.

[5] `https://huggingface.co/bert-large-cased`



(a) Precision



(b) Recall

Figure 5: The precision and recall with different argument spans length.