Conversational Semantic Role Labeling with Predicate-Oriented Latent Graph

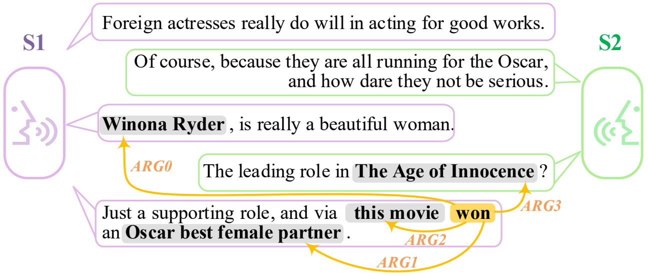
Hao Fei[1] , **Shengqiong Wu**[1] , **Meishan Zhang**[2] , **Yafeng Ren**[3*] and **Donghong Ji**[1*]
[1]Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, China
[2]Institute of Computing and Intelligence, Harbin Institute of Technology (Shenzhen), China
[3]Laboratory of Language and Artificial Intelligence, Guangdong University of Foreign Studies, China
{hao.fei, whuwsq, renyafeng, dhji}@whu.edu.cn, mason.zms@gmail.com

Abstract

Conversational semantic role labeling (CSRL) is a newly proposed task that uncovers the shallow semantic structures in a dialogue text. Unfortunately several important characteristics of the CSRL task have been overlooked by the existing works, such as the structural information integration, near-neighbor influence. In this work, we investigate the integration of a latent graph for CSRL. We propose to automatically induce a predicate-oriented latent graph (POLar) with a predicate-centered Gaussian mechanism, by which the nearer and informative words to the predicate will be allocated with more attention. The POLar structure is then dynamically pruned and refined so as to best fit the task need. We additionally introduce an effective dialogue-level pretrained language model, CoDiaBERT, for better supporting multiple utterance sentences and handling the speaker coreference issue in CSRL. Our system outperforms best-performing baselines on three benchmark CSRL datasets with big margins, especially achieving over 4% F1 score improvements on the cross-utterance argument detection. Further analyses are presented to better understand the effectiveness of our proposed methods.

1 Introduction

Semantic Role Labeling (SRL) as a shallow semantic structure parsing task aims to find all the arguments for a given predicate [Gildea and Jurafsky, 2000; Marcheggiani and Titov, 2017; Strubell *et al.*, 2018; Fei *et al.*, 2020d; Fei *et al.*, 2021b]. Conversational SRL (CSRL) is a newly proposed task by Xu *et al.* [2021], which extends the regular SRL into multi-turn dialogue scenario. As illustrated in Fig. 1, CSRL is characterized by that, the predicate is given at current utterance, while the correlated arguments are scattered in the history utterances of the dialogue that are generated by two speakers. So far, few attempts have been made for CSRL [Xu *et al.*, 2021; Wu *et al.*, 2021b; Wu *et al.*, 2021a], where, unfortunately, several key CSRL characteristics are still remained unexploted, which may hamper the further task improvements.



Foreign actresses really do will in acting for good works.

Of course, because they are all running for the Oscar, and how dare they not be serious.

Winona Ryder , is really a beautiful woman.

The leading role in **The Age of Innocence** ?

Just a supporting role, and via **this movie** **won** an **Oscar best female partner** .

Figure 1: Illustration of conversational SRL by two speakers. Word 'won' in yellow background is the predicate, linking to its different types of arguments by arrows. The arugments in the same utterance of the predicate are called intra-utterance arugment; those in different dialogue turns are marked as cross-utterance arugment.

First of all, intuitively SRL structure echoes much with the syntactic dependency structure [Strubell *et al.*, 2018; Marcheggiani and Titov, 2017], and the existing regular SRL works frequently employ external structural information for performance enhancement, i.e., providing additional prior links between predicates and arguments. However, it is quite intractable to directly employ the external syntax knowledge into CSRL for some reasons. For examples, a dependency tree takes one single sentence piece as a unit, while a dialogue could contain multiple utterance sentences; the parse trees from third-party parsers inevitably involve noises; only a small part of the dependency structure can really offer helps, rather than the entire tree [He *et al.*, 2018]. **Second**, the predicate-argument structures in CSRL are broken down and scattered into different utterances, which makes the detection of the CSRL more challenging. Actually the chances are much higher for the predicate to find its arguments when they are being closer, i.e., near-neighbor influence. In other words, nearer history utterances will show more impacts to the latest utterance.[1] Fig. 1 exemplifies the case.

Based on the above observations, in this paper we present an effective CSRL method with an innovative predicate-

[*] Corresponding author

[1] Our data statistics shows that, cross-one-utterance arguments account for 60.3% among all cross-turn arguments; while the ratio decreases to 30.3% for cross-two-utterance arguments.

oriented latent graph (namely, POLar). Unlike the explicit syntactic structures, we make use of a two-parameter *Hard-Kuma* distribution [Bastings *et al.*, 2019] to automatically induce latent graph from task's need (cf. §4). Particularly, we propose a predicate-centered Gaussian inducer for yielding the latent edges, by which the nearer and informative words to the predicate will be placed with more considerations. The POLar is then dynamically pruned, so that only the task-relevant structure will be built, while the irrelevant edges are droped. The overall CSRL framework is differentiable and performs predictions end-to-end (cf. Fig. 2).

The BERT [Devlin *et al.*, 2019] pre-trained language model (PLM) is extensively employed in existing works for CSRL performance boosts [Xu *et al.*, 2021; Wu *et al.*, 2021a]. Nevertheless, it could be problematic to directly leverage BERT for CSRL. On the one hand, one entire dialog often consists of far more than two utterance sentences, while the raw BERT restricts the input with at maximum two sentence pieces, which consequently limits the PLM's utility. Therefore, we consider adopting the DiaBERT [Liu and Lapata, 2019; Li *et al.*, 2020], which is designed for well supporting multiple utterance inputs and thus yields better dialogue-level representations. On the other hand, we note that in CSRL both two speakers use the personal pronoun in their own perspective (i.e., 'I', 'you'), and directly concatenating the multi-turn utterances into PLM will unfortunately hurt the speaker-role consistency, i.e., speaker coreference issue. Therefore, we introduce a coreference-consistency-enhanced DiaBERT (namely CoDiaBERT, cf. Fig. 3) that enhances the speaker-role sensitivity of PLM with a pronoun-based speaker prediction (PSP) strategy.

Our system significantly outperforms strong-performing baselines with big margins on three CSRL benchmarks. In particular, over 4% F1 score of improvement is achieved for detecting the cross-utterance type of arguments. Further analyses reveal the usefulness of the proposed latent graph and the dynamic pruning method, as well as the CoDiaBERT PLM. Also we show that our model effectively solves long-range dependence issue. Overall, we make these contributions:

- We for the first time propose to improve the CSRL task by incorporating a novel latent graph structure.
- We construct a predicate-oriented latent graph via a predicate-centered Gaussian inducer. The structure is dynamically pruned and refined for best meeting the task need.
- We introduce a PLM for yielding better dialogue-level text representations, which supports multiple utterance sentences, and is sensitive to the speaker roles.
- Our framework achieves new state-of-the-art CSRL results on three benchmark data.

2 Related Work

The SRL task aims at uncovering the shallow semantic structure of text, i.e. '*who did what to whom where and when*'. As a fundamental natural language processing (NLP) task, SRL can facilitate a broad range of downstream applications [Shen and Lapata, 2007; Liu and Gildea, 2010; Wang *et al.*, 2015]. By installing the current neural models, the current standard SRL has secured strong task performances [Strubell *et al.*, 2018; Li *et al.*, 2019; Fei *et al.*, 2021c]. Recently, Xu *et al.* [2021] pioneer the task of CSRL by extending the regular SRL into multi-turn dialogue scenario, in which they provide benchmark datasets and CSRL neural model. Later a limited number of subsequent works have explored this task [Wu *et al.*, 2021b; Wu *et al.*, 2021a], where unfortunately several important features of CSRL are not well considered. In this work, we improve the CSRL by fully uncovering the task characteristics.

This work also closely relate to the line of syntax-driven SRL [Marcheggiani and Titov, 2017; Fei *et al.*, 2020c; Fei *et al.*, 2020b]. For the regular SRL, the external syntactic dependency structure is a highly-frequently equipped feature for performance enhancement, as the SRL shares much underlying structure with syntax [He *et al.*, 2018; Fei *et al.*, 2020a; Fei *et al.*, 2021a]. However, it could be problematic for CSRL to directly benefit from such convient syntactic knowledge, due to the dialogue nature of the text as we revealed earlier. We thus propose to construct a latent structure at dialogue level, so as to facilitate the CSRL task with structural knowledge. In recent years, constructing latent graph for downstream NLP tasks has received certain research attention [Choi *et al.*, 2018]. As an alternative to the pre-defined syntactic dependency structure yielded from third-party parsers, latent structure induced from the task context could effectively reduce noises [Corro and Titov, 2019], and meanwhile enhance the efficacy (i.e., creating task-relevant connections) [Chen *et al.*, 2020]. In this work, we revisit the characteristic of CSRL, and based on the two-parameter Hard-Kuma distribution [Bastings *et al.*, 2019] investigate a predicate-oriented latent graph by proposing a predicate-centered Gaussian inducer.

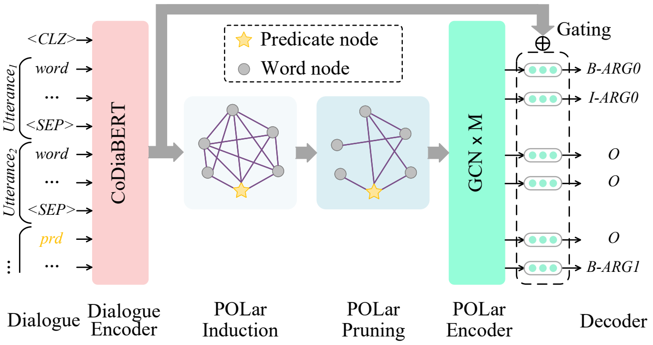


Figure 2: The overall CSRL framework.

3 CSRL Framework

Task modeling. Consider a conversation text $U{=}\{u_t\}_{t=1}^{T}$ (T is the total utterance number), with each utterance $u_t{=}\{w_0, w_1, \cdots\}$ a sequence of words (w_0 is the utterance speaker). In CSRL the predicate prd is labeled as input at the current (lastest) utterance u_T. We follow Xu *et al.* [2021], modeling the task as a sequence labeling problem with a *BIO* tagset. CSRL system identifies and classifies the arguments of a predicate into semantic roles, such as *A0*, *A1*, *AM-LOC*, etc, where we denote the complete role set as $\mathcal{R}$. Given U

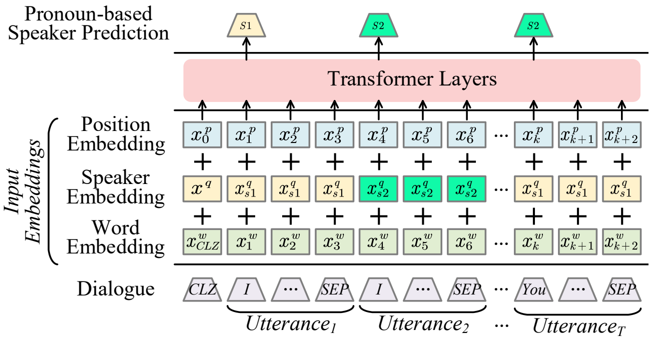

Pronoun-based Speaker Prediction
$s1$
$s2$
$s2$
Transformer Layers
Input Embeddings
Position Embedding
x_0^p x_1^p x_2^p x_3^p x_4^p x_5^p x_6^p ··· x_k^p x_{k+1}^p x_{k+2}^p
Speaker Embedding
x^q x_{s1}^q x_{s1}^q x_{s1}^q x_{s2}^q x_{s2}^q x_{s2}^q ··· x_{s1}^q x_{s1}^q x_{s1}^q
Word Embedding
x_{CLZ}^w x_1^w x_2^w x_3^w x_4^w x_5^w x_6^w ··· x_k^w x_{k+1}^w x_{k+2}^w
Dialogue
CLZ I ··· SEP I ··· SEP ··· You ··· SEP
Utterance$_1$
Utterance$_2$
···
Utterance$_T$

Figure 3: Illustration of the CoDiaBERT.

and the predicate prd, the system finally assigns each word w a label $\hat{y} \in \mathcal{Y}$, where $\mathcal{Y}$=$(\{B, I\}\times\mathcal{R}) \cup \{O\}$.

Framework overview. Our overall CSRL framework is illustrated in Fig. 2. The dialogue encoder first yields contextual representations for the input dialogue texts. Then, the system generates the predicate-oriented latent graph (i.e., POLar induction), and performs structure pruning. Afterwards, GCN layers encode the POLar into feature representations, based on which the predictions are finally made.

3.1 CoDiaBERT: Dialogue Encoder

Contextualized word representations from BERT have brought great benefits to CSRL [Xu *et al.*, 2021; Wu *et al.*, 2021b; Wu *et al.*, 2021a]. In this work, we follow them by borrowing the advances from PLM as well. However, we notice that the raw BERT limits the input with maximum two sentence pieces, while often a conversation text can comprise far more than two utterance sentences. Directly using BERT can thus lead to discourse information incoherency. We thus leverage a dialogue-level BERT-like PLM *DiaBERT* [Liu and Lapata, 2019]. Technically, we pack the utterance with its speaker as a group, and concatenate those groups into a whole (separated with *SEP* tokens), and feed into the PLM encoder.

The speaker coreference issue in conversational context may quite confuse the model. For example, speaker #1 would call speaker #2 'you' in speaker #1's utterance, while both speaker #1 and speaker #2 call themselves with the first-person pronoun 'I'. To strengthen the sensitivity of the speaker role, we further retrofit the DiaBERT so as to enhance the coreference consistency, i.e., CoDiaBERT. Specifically, we based on the well-trained DiaBERT perform a *pronoun-based speaker prediction* (PSP) upon DiaBERT, as shown in Fig. 3. We first concatenate different utterance texts into a whole piece that are separated with <*SEP*> token. Then we prepare three types of embeddings for each input token: 1) word embedding $\boldsymbol{x}^w$, 2) speaker id embedding $\boldsymbol{x}^q$, and 3) position embedding $\boldsymbol{x}^p$, all of which are fed into PLM for PSP:

$$\begin{aligned} \boldsymbol{x}_i &= [\boldsymbol{x}^p; \boldsymbol{x}^q; \boldsymbol{x}^w]_i \,, \\ \{\cdots, \boldsymbol{h}_i, \cdots\} &= \text{CoDiaBERT}^{\text{PSP}}(\{\cdots, \boldsymbol{x}_i, \cdots\}) \,. \end{aligned} \tag{1}$$

Based on the pronoun representation (i.e., the corresponding word is a pronoun), we encourage the PLM to predict the speaker id.

After PSP, the CoDiaBERT could yields better dialogue representations. In our CSRL framework, CoDiaBERT will take as input the conversation texts (including the speaker id) as well as the predicate word annotation:

$$\begin{aligned} \boldsymbol{x}_i &= [\boldsymbol{x}^p; \boldsymbol{x}^q; \boldsymbol{x}^w; \boldsymbol{x}^{prd}]_i \,, \\ \{\cdots, \boldsymbol{h}_i, \cdots\} &= \text{CoDiaBERT}^{\text{enc}}(\{\cdots, \boldsymbol{x}_i, \cdots\}) \,. \end{aligned} \tag{2}$$

where $\boldsymbol{x}^{prd}$ is the predicate binary embeddings $\{0, 1\}$ indicating the presence or absence of the predicate word prd. $\boldsymbol{h}_i$ denotes the output representation for the input token w_i.

3.2 Latent Graph Encoder

Based on the CoDiaBERT representation[2] we can construct the POLar structure, which we will elaborate in the next section (cf. §4). In the POLar $\mathcal{G} = (V, E)$, each edge $\pi_{i,j} \in E$ is a real value that denotes a latent connection between node $v_i \in V$ to node $v_j \in V$ with a connecting intensity, i.e., E is a $K \times K$ adjacent matrix ($|V| = K$).[3] Once we obtain the POLar we encode it into feature representations. Specifically, we employ a multi-layer (M) graph convolutional network (GCN) [Marcheggiani and Titov, 2017]. We denote the m-th layer of GCN hidden representation of node v_i as $\boldsymbol{r}_i^m$:

$$\boldsymbol{r}_i^m = \text{ReLU}(\textstyle\sum_{j=1}^{K} \bar{A}_{i,j} \boldsymbol{W}_1^m \boldsymbol{r}_j^{m-1} / d_i + b^m) \,, \tag{3}$$

where $\bar{A} = E + I$ (I is a $K \times K$ identity matrix), $d_i = \sum_{j=1}^{K} E_{i,j}$ is for node normalization. Note that the input of the initial layer is the CoDiaBERT representations, i.e., $\boldsymbol{r}_i^0 = \boldsymbol{h}_i$ After total M layers of message propagations, we expect the GCN can sufficiently capture the structural features.

3.3 Decoder and Training

To take the full advantages of the global dialogue contextual features, we create a residual connection from CoDiaBERT to the end of the GCN layer:

$$\boldsymbol{e}_i = g_i \odot \boldsymbol{r}_i^M + (1 - g_i) \odot \boldsymbol{h}_i \,, \tag{4}$$

where $\boldsymbol{e}_i$ is the final feature representation, which fuses both the contextual features and the structure-aware features. g_i is a gate mechanism that is learned dynamically:

$$g_i = \sigma(\boldsymbol{W}_2 \cdot [\boldsymbol{r}_i^M; \boldsymbol{h}_i]) \,. \tag{5}$$

Based on $\boldsymbol{e}_i$ we adopt a Softmax classifier to predict the labels for tokens:

$$\hat{y}_i = \text{Softmax}(\boldsymbol{e}_i) \,. \tag{6}$$

Also the Viterbi algorithm is used to search for the highest-scoring tag sequence $\hat{\boldsymbol{Y}} = \{\hat{y}_1, \cdots, \hat{y}_K\}$.

Our training objective is to minimize the cross-entropy loss between the predictions $\hat{\boldsymbol{Y}}$ and the gold labels $\boldsymbol{Y}$.

$$\mathcal{L} = -\frac{1}{K} \textstyle\sum_{j=1}^{K} y_j \log \hat{y}_j \,, \tag{7}$$

where K is the total sequence length (i.e., $|V|$).

4 Predicate-Oriented Latent Graph Induction

Since the goal of CSRL is to find the arguments of the predicate, it is crucial to treat the predicate word as the pivot and

[2] We abandon the special sentinel tokens (e.g., <*CLZ*> and <*SEP*>), and only make use of the word and speaker tokens.

[3] The node could either be a word, or a speaker.

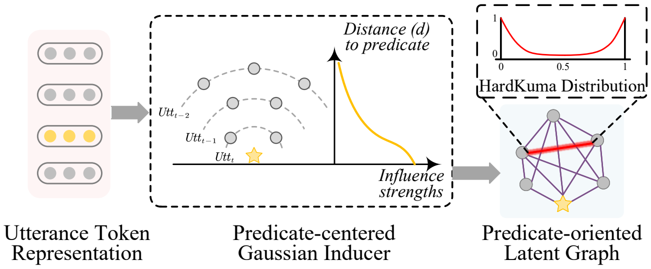

Distance (d) to predicate
Influence strengths
Utt_{t-2}
Utt_{t-1}
Utt_t
1
0
0
0.5
1
HardKuma Distribution
Utterance Token Representation
Predicate-centered Gaussian Inducer
Predicate-oriented Latent Graph

Figure 4: Induction of the predicate-oriented latent graph.

induce a predicate-oriented latent graph (POLar) to fully consider the near-neighbor influence. Here we demonstrate how to develop the POLar structure. First, we give a description on the theoretical fundamentation of the *HardKuma* distribution, upon which we build the latent strucutre. Then we introduce the predicate-centered Gaussian inducer. Finally we present the method for dynamically pruning the POLar.

4.1 HardKuma Distribution

HardKuma distribution [Bastings *et al.*, 2019] is derived from the *Kumaraswamy* distribution (namely Kuma) [Kumaraswamy, 1980], which is a two-parameters distribution over an open interval (0, 1), i.e., $t \sim \text{HardKuma}(a, b)$ where $a \in \mathbb{R}_{>0}$ and $b \in \mathbb{R}_0$ are the parameters controlling the shapes. However, the Kuma distribution does not cover the two discrete points 0 and 1. Thus, the HardKuma distribution adopts a *stretch-and-rectify* method to support the closed interval of [0, 1]. This feature allows to predict soft connections probabilities between input words, i.e., a latent graph, where the entire process is fully differentiable.

First, we sample a variable from a (0,1) distribution, i.e., $U \sim \mathcal{U}(0, 1)$, based on which we generate another variable from HardKuma's inverse CDF function:

$$k = \mathrm{F}_K^{-1}(u, a, b)\,. \tag{8}$$

Then we stretch the k into t:

$$t = l + (r - l) * k\,, \tag{9}$$

where $l < 0$ and $r > 1$ represent an open interval (l,r).[4] A Hard-Sigmoid function rectifies the t into h via

$$\mathrm{F}_T^{-1}(t; a, b, l, r) = \mathrm{F}_K(\frac{t - l}{r - l}; a, b)\,. \tag{10}$$

In short, we can summarize the HardKuma distribution as:

$$t \sim \text{HardKuma}(a, b, l, r)\,. \tag{11}$$

For more technical details we refer the readers to the raw papers [Bastings *et al.*, 2019].

4.2 Predicate-centered Gaussian Inducer

By sampling variables from *HardKuma* distribution with trained parameters a and b, we can generate the latent graph based upon the dialogue. Specifically, we present a predicate-centered Gaussian inducer (PGI), so that the near neighbors to predicate that carry more important information would serve more contributions.

As depicted in Fig. 4, we first upgrade each token representation into $\boldsymbol{h}_i^{'}$ with the prior of predicate word, via a predicate-centered Gaussian operator:

$$\begin{aligned} \boldsymbol{h}_i^{'} &= \text{PGI}(\boldsymbol{h}_i | \boldsymbol{h}_{i(prd)})\,, \\ &= \frac{f(d_{i,i(prd)})\text{Softmax}(\frac{\boldsymbol{h}_i \cdot \boldsymbol{h}_{i(prd)}}{\sqrt{d_{i,i(prd)}}})}{\sum_l f(d_{i,l})\text{Softmax}(\frac{\boldsymbol{h}_i \cdot \boldsymbol{h}_l}{\sqrt{d_{i,i(prd)}}})}\,, \end{aligned} \tag{12}$$

where $d = |i - i(prd)|$ is the edit distance between a token w_i and the predicate prd. Here $f(d)$ is a Gaussian distance, i.e., $f(d) = \exp(-\pi d^2)$. So $\boldsymbol{h}_i^{'}$ is reduced into:

$$\boldsymbol{h}_i^{'} = \text{Softmax}(-\pi d_{i,i(prd)}^2 + \frac{\boldsymbol{h}_i \cdot \boldsymbol{h}_l}{\sqrt{d_{i,i(prd)}}})\,. \tag{13}$$

Based on $\boldsymbol{h}_i^{'}$, we then create the parameter context representations (i.e., denoted as $\boldsymbol{s}^a$ and $\boldsymbol{s}^b$) via separate feedforward layers (i.e., $\boldsymbol{s}_i^{a/b}$=FNN$^{a/b}$($\boldsymbol{h}_i^{'}$)). Then we build the prior parameter representations of the distribution:

$$\begin{aligned} \boldsymbol{a} &= \text{Norm}(\boldsymbol{s}_i^a (\boldsymbol{s}_j^a)^T)\,, \\ \boldsymbol{b} &= \text{Norm}(\boldsymbol{s}_i^b (\boldsymbol{s}_j^b)^T)\,. \end{aligned} \tag{14}$$

Thereafter, we can sample a soft adjacency matrix between tokens, i.e., $\pi_{i,j} \in E$:

$$\pi_{i,j} = \text{HardKuma}(a_{i,j}, b_{i,j}, l, r)\,. \tag{15}$$

4.3 Dynamic Structural Pruning

There are high chances that the induced POLar structure is dense, which would introduce unnecessary paths that are less-informative to the task need, i.e., noises. Therefore, we adopt the α-Entrmax [Correia *et al.*, 2019] to prune the POLar. α-Entrmax imposes sparsity constraints on the adjacency matrix E, and the pruning process automatically removes irrelevant information according to the contexts dynamically:

$$E = \alpha\text{-Entrmax}(E)\,, \tag{16}$$

where α is a dynamic parameter controlling the sparsity. When α=2 the Entrmax becomes a Sparsemax mapping, while α=1 it degenerates into a Softmax mapping.

5 Experimentation

5.1 Setups

We conduct experiments on three CSRL datasets [Xu *et al.*, 2021], including DuConv, NewsDialog and PersonalDialog, with average 10.1, 5.2 and 6.1 utterances per dialogue, respectively. All the three data is in Chinese language. We take the default data split as in Xu *et al.* [2021], where DuConv has the 80%/10%/10% ratio of train/dev/test, while NewsDialog and PersonalDialog are taken as out-of-domain test set. Our CoDiaBERT shares the same architecture with the official BERT/DiaBERT (Base version), and is further post-trained on the CSRL data with PSP strategy. GCN hidden size is set as 350. We adopt Adam as the optimizer with an initial learning rate of 5e-4 with weight decay of 1e-5. The initial α value is 1.5. To alleviate overfitting, we use a dropout rate of 0.5 on the input layer and the output layer.

We mainly make comparisons with the existing CSRL baselines, including CSRL [Xu *et al.*, 2021], CSAGN [Wu

[4] Following the standard setup in Bastings *et al.* [2019], l=-0.1 and r=-1.1.

	DuConv			NewsDialog			PersonalDialog		
	F1$_{all}$	F1$_{cross}$	F1$_{intra}$	F1$_{all}$	F1$_{cross}$	F1$_{intra}$	F1$_{all}$	F1$_{cross}$	F1$_{intra}$
• w/ BERT									
SimplePLM [Shi and Lin, 2019]*	86.54	81.62	87.02	77.68	51.47	80.99	66.53	30.48	70.00
CSRL [Xu *et al.*, 2021]*	88.46	81.94	89.46	78.77	51.01	82.48	68.46	32.56	72.02
DAP [Wu *et al.*, 2021a]†	89.97	86.68	90.31	81.90	56.56	84.56	-	-	-
CSAGN [Wu *et al.*, 2021b]*	89.47	84.57	90.15	80.86	55.54	84.24	71.82	36.89	75.46
UE2E [Li *et al.*, 2019]	87.46	81.45	89.75	78.35	51.65	82.37	67.18	30.95	72.15
LISA [Strubell *et al.*, 2018]	89.57	83.48	91.02	80.43	53.81	85.04	70.27	32.48	75.70
SynGCN [Marcheggiani and Titov, 2017]	90.12	84.06	91.53	82.04	54.12	85.35	70.65	34.85	76.96
POLar	**92.06**	**90.75**	**92.64**	**83.45**	**60.68**	**87.96**	**73.46**	**40.97**	**78.02**
• w/ CoDiaBERT									
SimplePLM [Shi and Lin, 2019]	88.40	82.96	88.25	79.42	53.46	82.77	68.86	33.75	72.23
SynGCN [Marcheggiani and Titov, 2017]	91.34	86.72	91.86	82.86	56.75	85.98	72.06	37.76	77.41
POLar	**93.72**	**92.86**	**93.92**	**85.10**	**63.85**	**88.23**	**76.61**	**45.47**	**78.55**

Table 1: Main results on three datasets. Values with $*$ are copied from Wu *et al.* [2021b]; with † are copied from Wu *et al.* [2021a]; the rest are from our implementations.

	F1$_{all}$ (Δ)	F1$_{cross}$ (Δ)	F1$_{intra}$ (Δ)
POLar	**93.72**	**92.86**	**93.92**
• CoDiaBERT			
→BERT	92.70 (-1.02)	90.75 (-2.11)	93.04 (-0.88)
w/o PSP	92.98 (-0.74)	91.28 (-1.58)	93.37 (-0.55)
PSP→spk-lb	93.34 (-0.38)	92.04 (-0.82)	93.80 (-0.12)
• POLar			
w/o PGI	91.86 (-1.86)	87.28 (-5.58)	91.75 (-2.17)
w/o Pruning	92.25 (-1.47)	89.74 (-3.12)	92.21 (-1.17)
w/o g_i (Eq. 5)	93.26 (-0.46)	92.27 (-0.59)	93.50 (-0.42)

Table 2: Ablation results on DuConv dataset.

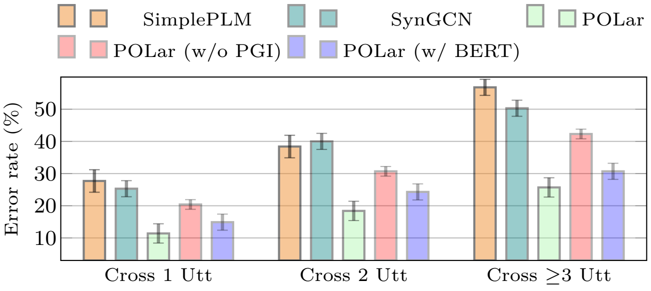


Figure 5: Error rate on cross-uttereance argument role detection.

et al., 2021b] and DAP [Wu *et al.*, 2021a]. Also we implement several representative and strong-performing models designed for regular SRL, including UE2E [Li *et al.*, 2019], LISA [Strubell *et al.*, 2018] and SynGCN [Marcheggiani and Titov, 2017], in which we concatenate the utterances into a long sequence. In particular, LISA and SynGCN use the external syntactic dependency trees. Follow Xu *et al.* [2021], we compute the F1 score for the detection of intra-/cross-utterance arguments (i.e., F1$_{intra}$ and F1$_{cross}$), and the overall performance (F1).

5.2 Results and Discussions

Main results. Table 1 presents the main performances by different models, from which we gain several observations. First of all, our proposed POLar system significantly outperforms all the baselines by large margins on both the in-domain and out-domain datasets, which demonstrates the efficacy of our method. Specifically, we notice that our model achieves at least 4.07%(=90.75-86.68) and at most 7.71%(=45.47-37.76) F1 improvements on the cross-utterance argument detection, over the corresponding best baselines. This significantly proves the superiority of our method on the cross-turn context modeling. Second, by comparing the results with BERT and with CoDiaBERT, we know that our proposed CoDiaBERT PLM is of prominent helpfulness for the task. Third, we see that with the aid of external syntactic dependency structure information, SynGCN and LISA models achieve considerable performance gains over the existing CSRL baselines (i.e., CSAGN, DAP). However, such improvements are limited to the detection of intra-utterance arugments, contributing less to the cross-utterance arugments. The possible reason is that, the dependency tree only works at sentence level, which fails to capture the cross-uttereance contexts. Fortunately, our proposed latent graph can nicely compensate for this.

Ablation study. In Table 2 we give the model ablation results with respect to the CoDiaBERT PLM and the POLar parts, respectively. We can observe that, by replacing the CoDiaBERT with a vanilla BERT or removing the pronoun-based speaker prediction policy (downgraded as DiaBERT), there can be considerable drops. If we strip off the PSP, and instead use the speaker id indicator to label the speaker pronoun (i.e., spk-lb), we also witness the drops.

Further, without the PGI for the latent graph induction, i.e., directly feeding the PLM representations $\boldsymbol{h}$ in Eq. 14 instead of $\boldsymbol{s}$, we can receive the most significant performance drops among all the other factors, i.e., -5.58%F1 on the cross-utterance arguments detection. This also reflects the importance to handle the near-neighbor influence of CSRL. Besides, the graph pruning is quite important to the results of cross-utterance arguments. The gating mechanism takes the positive roles to the system.

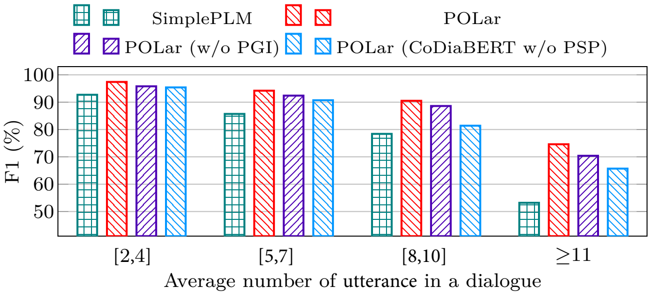


Figure 6: Influence of the utterance number in dialogue.

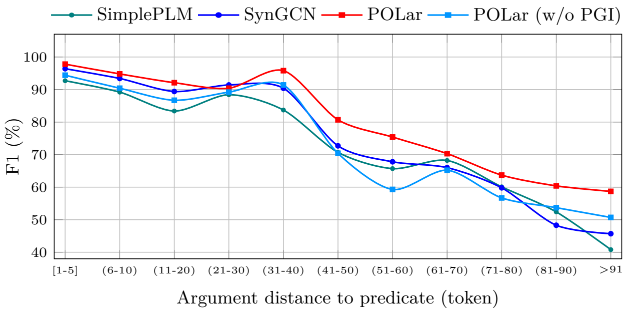


Figure 7: Influence of the argument-predicate distance.

Performances on cross-utterance argument detection. In Fig. 5 we study the error rate on the cross-utterance argument detection. We see that with the increase of the crossed utterances, the error for the argument detection grows universally. But in all the cases, our POLar system commits nearly half error rate, comparing to baselines. Also we notice that, both the PGI mechanism and the CoDiaBERT is important to our system, with the former more significant than the latter.

Impacts of utterance numbers. Intuitively the more the utterance in a dialogue the severe complexity of the speaker parties, i.e., due to the speaker coreference issue. Fig. 6 further plots the performances under different numbers of dialogue utterances. It is clear that increasing the utterance number in a dialogue worsens the overall results, especially when the number $\geq$11. In particular, the removal of PSP in CoDiaBERT shows greater impact to the removal of the PGI mechanism. This indirectly proves that CoDiaBERT can help solve the speaker coreference issue, which gives rise to the performance gains.

Solving long-range dependence issue. Structure information has been shown effective for relieving the long-range dependence issue in SRL [He *et al.*, 2018; Fei *et al.*, 2021a]. Here we explore the performances when the distances between the arguments and the predicates are different in the dialogue. Fig. 7 shows that, notably, our system equipped with the latent graph performs well for those super-long argument-predicate distances, where the other baselines could fail. Also the ablated POLar system (w/o PGI) reflects the importance of the predicate-certered Gaussian mechanism.

Study of the dynamic pruning for latent graph. Finally, we investigate the process of the dynamic pruning by studying the changing pattern of α-Entrmax (Eq. 16). Fig. 8 plots the learning trajectories of parameter α as well as the variations of the correlated task performances (on three datasets). We see that, along the training process, the α soon decreases to 1.35 from 1.5 at step 1,500, and then grow to 1.9, during which the latent graph becomes dense and then turns sparse gradually. At the meantime, the CSRL performances climb to the top slowly. This suggests that the dynamic pruning process improves the quality of POLar, which helps lead to better task demand of structure.

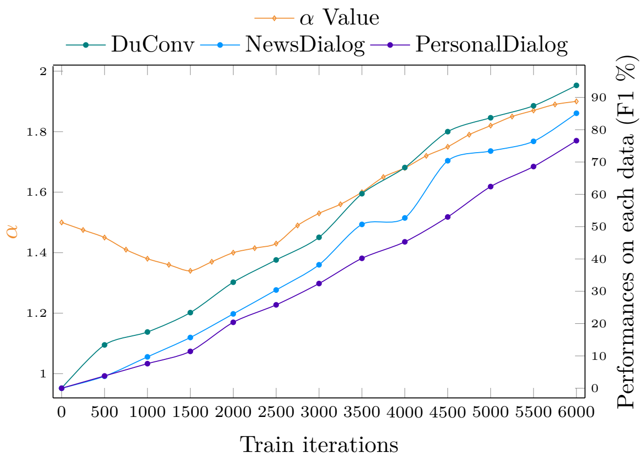


Figure 8: Trajectories of the changing pattern of α value, and the task performances on different data.

6 Conclusions

In this work we investigate the integration of a latent graph for conversational semantic role labeling. We construct a predicate-oriented latent graph based on the two-parameter HardKuma distribution, which is induced by a predicate-centered Gaussian mechanism. The structure is dynamically pruned and refined to best meet the task need. Also we introduce a dialogue-level PLM for yielding better conversational text representations, e.g., supporting multiple utterance sentences, and being sensitive to the speaker roles. Our system outperforms best-performing baselines with big margins, especially on the cross-utterance arguments. Further analyses demonstrate the efficacy of the proposed latent graph as well as the dialogue-level PLM, respectively. Automatically inducing task-oriented latent structure features for the structural parsing tasks is promising, which we leave as a future work.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No.61772378, No. 62176187), the National Key Research and Development Program of China (No. 2017YFC1200500), the Research Foundation of Ministry of Education of China (No.18JZD015), the Key Project of State Language Commission of China (No. ZDI135-112) and the Science of Technology Project of GuangZhou (No. 20210202607).

References

[Bastings *et al.*, 2019] Jasmijn Bastings, Wilker Aziz, and Ivan Titov. Interpretable neural predictions with differentiable binary variables. In *Proc. of ACL*, pages 2963–2977, 2019.

[Chen *et al.*, 2020] Chenhua Chen, Zhiyang Teng, and Yue Zhang. Inducing target-specific latent structures for aspect sentiment classification. In *Proc. of EMNLP*, pages 5596–5607, 2020.

[Choi *et al.*, 2018] Jihun Choi, Kang Min Yoo, and Sang-goo Lee. Learning to compose task-specific tree structures. In *Proc. of AAAI*, pages 5094–5101, 2018.

[Correia *et al.*, 2019] Gonçalo M. Correia, Vlad Niculae, and André F. T. Martins. Adaptively sparse transformers. In *Proc. of EMNLP*, pages 2174–2184, 2019.

[Corro and Titov, 2019] Caio Corro and Ivan Titov. Learning latent trees with stochastic perturbations and differentiable dynamic programming. In *Proc. of ACL*, pages 5508–5521, 2019.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL*, pages 4171–4186, 2019.

[Fei *et al.*, 2020a] Hao Fei, Yafeng Ren, and Donghong Ji. Improving text understanding via deep syntax-semantics communication. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 84–93, 2020.

[Fei *et al.*, 2020b] Hao Fei, Yafeng Ren, and Donghong Ji. Mimic and conquer: Heterogeneous tree structure distillation for syntactic NLP. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 183–193, 2020.

[Fei *et al.*, 2020c] Hao Fei, Yafeng Ren, and Donghong Ji. Retrofitting structure-aware transformer language model for end tasks. In *Proc. of EMNLP*, pages 2151–2161, 2020.

[Fei *et al.*, 2020d] Hao Fei, Meishan Zhang, and Donghong Ji. Cross-lingual semantic role labeling with high-quality translated training corpus. In *Proc. of AAAI*, pages 7014–7026, 2020.

[Fei *et al.*, 2021a] Hao Fei, Fei Li, Bobo Li, and Donghong Ji. Encoder-decoder based unified semantic role labeling with label-aware syntax. In *Proc. of AAAI*, pages 12794–12802, 2021.

[Fei *et al.*, 2021b] Hao Fei, Shengqiong Wu, Yafeng Ren, Fei Li, and Donghong Ji. Better combine them together! integrating syntactic constituency and dependency representations for semantic role labeling. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021*, pages 549–559, 2021.

[Fei *et al.*, 2021c] Hao Fei, Meishan Zhang, Bobo Li, and Donghong Ji. End-to-end semantic role labeling with neural transition-based model. In *Proc. of AAAI*, pages 12803–12811, 2021.

[Gildea and Jurafsky, 2000] Daniel Gildea and Daniel Jurafsky. Automatic labeling of semantic roles. In *Proc. of ACL*, pages 512–520, 2000.

[He *et al.*, 2018] Shexia He, Zuchao Li, Hai Zhao, and Hongxiao Bai. Syntax for semantic role labeling, to be, or not to be. In *Proc. of ACL*, pages 2061–2071, 2018.

[Kumaraswamy, 1980] Ponnambalam Kumaraswamy. A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2):79–88, 1980.

[Li *et al.*, 2019] Zuchao Li, Shexia He, Hai Zhao, Yiqing Zhang, Zhuosheng Zhang, Xi Zhou, and Xiang Zhou. Dependency or span, end-to-end uniform semantic role labeling. In *Proc. of AAAI*, pages 6730–6737, 2019.

[Li *et al.*, 2020] Jingye Li, Hao Fei, and Donghong Ji. Modeling local contexts for joint dialogue act recognition and sentiment classification with bi-channel dynamic convolutions. In *Proc. of COLING*, pages 616–626, 2020.

[Liu and Gildea, 2010] Ding Liu and Daniel Gildea. Semantic role features for machine translation. In *Proc. of COLING*, pages 716–724, 2010.

[Liu and Lapata, 2019] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. In *Proc. of EMNLP*, pages 3730–3740, 2019.

[Marcheggiani and Titov, 2017] Diego Marcheggiani and Ivan Titov. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proc. of EMNLP*, pages 1506–1515, 2017.

[Shen and Lapata, 2007] Dan Shen and Mirella Lapata. Using semantic roles to improve question answering. In *Proc. of EMNLP*, pages 12–21, 2007.

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

[Strubell *et al.*, 2018] Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. Linguistically-informed self-attention for semantic role labeling. In *Proc. of EMNLP*, pages 5027–5038, 2018.

[Wang *et al.*, 2015] Hai Wang, Mohit Bansal, Kevin Gimpel, and David McAllester. Machine comprehension with syntax, frames, and semantics. In *Proc. of ACL*, pages 700–706, 2015.

[Wu *et al.*, 2021a] Han Wu, Kun Xu, Linfeng Song, Lifeng Jin, Haisong Zhang, and Linqi Song. Domain-adaptive pretraining methods for dialogue understanding. In *Proc. of ACL*, pages 665–669, 2021.

[Wu *et al.*, 2021b] Han Wu, Kun Xu, and Linqi Song. CSAGN: Conversational structure aware graph network for conversational semantic role labeling. In *Proc. of EMNLP*, pages 2312–2317, 2021.

[Xu *et al.*, 2021] Kun Xu, Han Wu, Linfeng Song, Haisong Zhang, Linqi Song, and Dong Yu. Conversational semantic role labeling. *IEEE ACM Trans. Audio Speech Lang. Process.*, 29:2465–2475, 2021.