Joint Multi-Decoder Framework with Hierarchical Pointer Network for Frame Semantic Parsing

Xudong Chen[1,2], **Ce Zheng**[1], **Baobao Chang**[1,3*]
[1]The MOE Key Laboratory of Computational Linguistics, Peking University, China
[2]School of Software and Microelectronics, Peking University, China
[3]Peng Cheng Laboratory, Shenzhen, China
{xdc,zce1112zslx,chbb}@pku.edu.cn

Abstract

Current researches on frame semantic parsing include three subtasks, namely frame identification, argument identification and role classification. Most of previous systems process these subtasks independently and ignore their interactions. We introduce a novel architecture based on multi-decoder strategy to handle these subtasks together. The multi-decoder strategy strengthens the interactions. Moreover, we design a hierarchical pointer network for argument identification which reduces the computational complexity. To our best knowledge, it's the first practice to introduce the pointer network into frame semantic parsing. The experiments show improvement over state of the art models on FrameNet dataset.

1 Introduction

Frame semantic parsing is a fundamental study in Natural Language Processing. It aims to parse sentences into frame-style semantic structures defined in FrameNet (Baker et al., 1998).

An example of frame-style semantic structures is shown in Figure 1. The word ***write.v*** is a target that evokes the frame called ***Text_creation***. The phrases underlined with green lines are called arguments. ***Author***, ***Text*** and ***Form*** are roles (also called frame elements) the arguments play in this frame. Hence the frame semantic parsing contains three subtasks, namely frame identification, argument identification and role classification. For a sentence with a given target, the frame identification is to disambiguate the frame for the target based on its contextual information, the argument identification is to identify the boundaries of all the arguments, and the role classification is to assign a semantic role to each argument we have found.

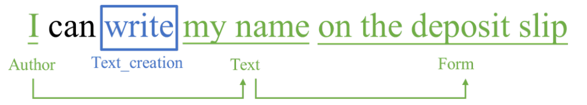

Figure 1: A sentence annotated the arguments and roles of frame ***Text_creation***. The arrow marks indicate the order of arguments identification and roles classification.

Early work (Hermann et al., 2014; FitzGerald et al., 2015; Hartmann et al., 2017) on frame semantic parsing adopts pipeline strategy. Their models apply independent models to handle different subtasks which ignore the interactions among subtasks. Moreover, the pipeline strategy usually causes error propagation problem. The accuracy of frame identification can become the bottleneck of the overall performance. Later work (Yang and Mitchell, 2017; Peng et al., 2018) processes all the subtasks jointly by optimizing them together during training. Their joint models show improvement over pipeline models, which demonstrates the benefit of joint training strategy. However, their systems don't have specific design to model the interactions among the subtasks.

To strengthen the interactions of subtasks, we propose a joint framework based on three task-specific decoders. The interactions in our framework are mainly reflected in two aspects. On one hand, the representations of both the target and its frame that derived from frame identification decoder are applied to predict the arguments and roles. On the other hand, the argument identification decoder and the role classification decoder work in an alternate way, and thus they interact with each other during the entire process of decoding.

The interactions bring two benefits. First, the frame information predicted by frame identification decoder makes the predictions of arguments and roles more frame-specific. Second, the alternate decoding strategy makes the current argument's and role's prediction influenced by all previous

[*] Corresponding author

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 2570–2578
August 1–6, 2021. ©2021 Association for Computational Linguistics

SENTENCE	Coming to Goodwill was the first step toward my becoming totally independent.
TARGET	**come.v**(Arriving), **to.prep**(Goal), **first.a**(Ordinal_numbers), **step.n**(Intentionally_act), **become.v**(Becoming), totally.adv(Degree)

Table 1: An example sentence with annotated targets and golden frames from FrameNet dataset. Bold words indicate that the target can evoke multiple frames.

decisions, which captures relations among different arguments and roles. For the sentence shown in Figure 1, the argument ***I*** and its role ***Author*** can contribute to the predictions of the argument ***my name*** and the role ***Text***. Therefore, the argument identification and the role classification can benefit from each other by considering arguments and roles already obtained.

For argument identification, previous models (Yang and Mitchell, 2017; Peng et al., 2018) enumerate all possible spans to identify the arguments, which brings high computational complexity. To reduce the computational complexity, we design a hierarchical pointer network in the argument identification decoder that predicts boundaries of arguments directly.

In addition, we design a target-aware attention mechanism. The target-aware attention mechanism aggregates all targets in the same sentence to model interactions among different targets, since frames evoked by different targets in the same sentence are usually closely related. Such interaction modeling could be helpful in frame identification. For example, the target ***to.prep*** in Table 1 can evoke the frame ***Goal*** or ***Locative_relation***, and other co-occurrence targets (such as ***come.v***) make ***to.prep*** more likely to evoke frame ***Goal*** instead of the other frame.

Overall, our main contributions can be summarized as follows:

- We design a novel multi-decoder framework to jointly process all the subtasks of frame semantic parsing. The multi-decoder strategy strengthens the interactions among frame identification, argument identification and role classification.

- We design a hierarchical pointer network that predicts the boundaries of arguments directly. The hierarchical pointer network predicts arguments within linear computational complexity. To our best knowledge, it's the first practice to introduce the pointer network into frame semantic parsing task.

- We design a target-aware attention mechanism to aggregate the semantic information of other targets in the same sentence.

We evaluate our model on FrameNet dataset, and the experiments show that our model outperforms state of the art models, which demonstrates the effectiveness of our model.

2 Related Work

Frame semantic parsing task is first proposed by Gildea and Jurafsky (2002) and has drawn attention since the SemEval 2007 shared task (Baker et al., 2007) was released. Early researches on frame semantic parsing focus on the feature-engineered methods(Johansson and Nugues, 2007; Das et al., 2010). Most of the early researches regard the frame semantic parsing as a pipeline of classification tasks and employ machine learning algorithms (such as Support Vector Machines etc.).

With the popularity of neural network and representation learning, neural network models are introduced to model frame semantic parsing problem. Hermann et al. (2014) uses distributed representations in frame identification and embedded both frames and the contextual representations of words into a shared low-dimension vector space. FitzGerald et al. (2015) uses a neural network to learn embeddings of both arguments and semantic roles, which adopts fine-grained similarity between roles to overcome the sparsity of some labeled data. Besides, a system based on pre-trained word distributed representations (Hartmann et al., 2017) is developed to improve the domain adaptation of their model. These models still work in pipeline way.

However, pipeline models usually ignore the interactions among subtasks and suffer from the problem of error propagation. Joint models are proposed to solve these problems. Yang and Mitchell (2017) proposes an ensemble strategy that that integrates two different models into an ensemble model. Peng et al. (2018) proposes a multi-task framework which jointly handles two different semantic pars-

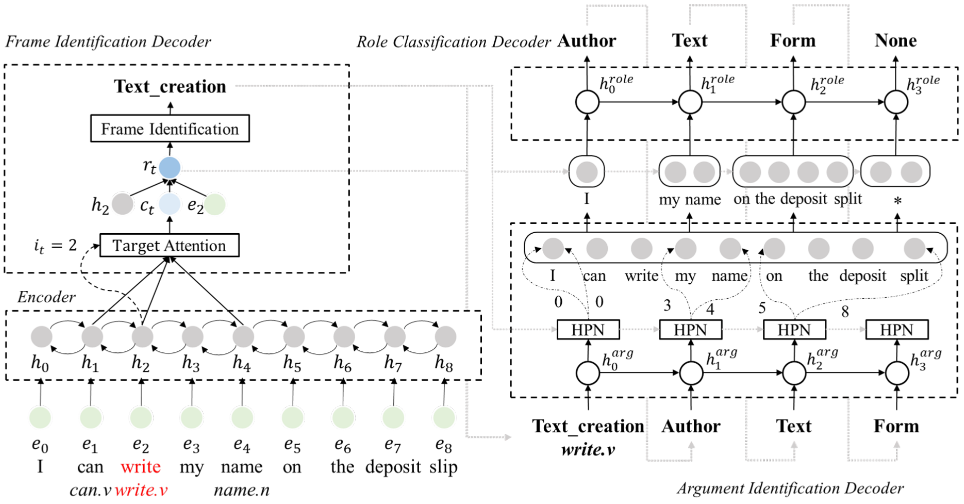

Figure 2: Our framework predicts frames, arguments and roles jointly based on three task-specific decoders. Frame identification module predicts frame ***Text_creation*** evoked by target ***write.v***. Then argument identification module with HPN(Hierarchical Pointer Network) and role classification module predict argument spans ***I***, ***my name*** and ***on the deposit split*** and their roles ***Author***, ***Text*** and ***Form*** alternately. The interactions in our framework are shown as gray dotted lines.

ing tasks from disjoint data. Both of their models process all the subtasks jointly by optimizing them together during training. Their experiments show improvement over previous pipeline models, which proves the benefit of joint training strategy. However their models don't have specific design for the interactions.

The common neural network architectures for frame semantic parsing can be divided into sequence labeling models and relational models. Yang and Mitchell (2017) proposes a sequence labeling model based on BIO tagging scheme. The model contains multiple LSTM layers and a Conditional Random Field (CRF) layer. Swayamdipta et al. (2017) adopts a segmentation RNN and a relational model to capture span-level dependencies between predicate and arguments. The relational model enumerates all possible spans to compute the matching scores. Both of these two type models above require $O(n^2)$ computational complexity. To reduce the high computational complexity, we design a hierarchical pointer network that achieves the identifying arguments within linear computational complexity.

3 Method

As is shown in Figure 2, our framework consists of four modules. (1) the encoder module (2) the frame identification decoder module. (3) the argument identification decoder module. (4) the role classification decoder module.

Specifically, given a target t and a sentence $S = w_0, \ldots, w_{n-1}$, the encoder module calculates the contextual representations $h_0, \ldots, h_{n-1}$, then all decoder modules handle three subtasks jointly. The frame identification decoder builds a target representation for t and identifies the frame $f \in F$ evoked by t. Suppose that there are k argument spans $a_0, \ldots, a_{k-1}$ of f in S. For each argument $a_\tau = w_{i^s_\tau}, \ldots, w_{i^e_\tau}$, the argument identification decoder identifies the boundaries i^s_τ and i^e_τ, and the role classification decoder assigns a semantic role $r_\tau \in R_f$ to a_τ. The F and R_f mentioned above are the sets of all frames and all semantic roles of f defined in FrameNet.

Three decoders interact with each other as follows:

- Frame identification decoder builds a target representation for t to identify the frame f. Both the target representation and the embedding of f will be taken as inputs to the other decoders for argument identification and role classification.

- Role classification decoder assigns a role r_τ to current argument span a_τ, and then the embedding of r_τ will be taken as an input to identify the boundary of next argument $a_{\tau+1}$. In other words, these two decoders work in an alternate way, so identifying $a_{\tau+1}$ and $r_{\tau+1}$ will consider all the historical information of $a_0, \ldots, a_\tau$ and their roles $r_0, \ldots, r_\tau$.

3.1 Encoder Module

Encoder module aims at converting the sentence $S = w_0, \ldots, w_{n-1}$ into a sequence of vectors $h_0, \ldots, h_{n-1}$, where h_i is the contextual representation of word w_i.

For each token, we concatenate its word embedding e_{w_i}, lemma embedding e_{l_i}, POS embedding e_{p_i} and a binary tag embedding e_{b_i}:

$$e_i = [e_{w_i}; e_{l_i}; e_{p_i}; e_{b_i}] \tag{1}$$

The binary tag embedding e_{b_i} is to distinguish t from other words in S. Let i_t be the position index of t in S, then we can calculate e_{b_i}:

$$e_{b_i} = \begin{cases} e_1, & i = i_t \\ e_0, & i \neq i_t \end{cases} \tag{2}$$

At last, e_i is fed to the encoder to get contextual representation h_i:

$$h_i = \text{Encoder}(e_i) \tag{3}$$

The word embedding and lemma embedding are initialized with Glove (Pennington et al., 2014) while POS embedding is randomly initialized. We use Bi-LSTM as the encoder in our experiment, which can be also replaced with any other encoder model such as Bert (Devlin et al., 2018). The dimension of e_i is d_e and the dimension of h_i is d_h.

3.2 Frame identification module

In frame identification module, we build a target representation r_t for t and identify the frame f based on r_t.

As there are likely to be multiple targets $t_0, \ldots, t_{m-1}$ in S evoking multiple frames $f_0, \ldots, f_{m-1}$ and we believe that other targets in S can contribute to identifying current frame f for target t, we design a target aware attention mechanism to aggregate contextual representations of all targets $T_S = \{t_0, \ldots, t_{m-1}\}$ in S (also contains the target t):

$$\alpha_i = \frac{\exp(h_i^\top W_1 h_{i_t})}{\sum_{j \in T_s} \exp(h_j^\top W_1 h_{i_t})} \tag{4}$$

$$c_t = \sum_{i \in T_s} \alpha_i h_i \tag{5}$$

For the target, we concatenate c_t, its contextual representation h_{i_t} and its embedding e_{i_t} to get the target representation r_t:

$$r_t = \text{Relu}(W_2 \cdot [e_{i_t}; h_{i_t}; c_t]) \tag{6}$$

With the target representation, we can generate the probability distribution of frames to identify f by argmax operation:

$$P(f|S, t) = \text{softmax}(W_3 \cdot r_t) \tag{7}$$

W_1, W_2, W_3 are three weight matrixes in $R^{d_h \times d_h}$, $R^{d_h \times (2d_h + d_e)}$ and $R^{|F| \times d_h}$ repectively, where $|F|$ is the size of F.

3.3 Argument Identification Module

Argument identification decoder module identifies the boundaries of argument spans $a_0, \ldots, a_{k-1}$ sequentially. For τ-th argument $a_\tau = w_{i_\tau^s}, \ldots, w_{i_\tau^e}$, the historical information of $a_0, \ldots, a_{\tau-1}$ and their roles $r_0, \ldots, r_{\tau-1}$ is supposed to be utilized. Hence We use LSTM_A to record the historical information, and similarly, another LSTM named LSTM_R is applied in role classification decoder. Argument identification decoder interacts with other decoders by taking their output as input, here is how LSTM_A works at τ-th argument:

$$x_\tau = \begin{cases} [r_t; e_f], & \tau = 0 \\ \left[h_{\tau-1}^{role}; e_{r_{\tau-1}}\right], & \tau > 0 \end{cases} \tag{8}$$

$$h_\tau^{arg} = \text{LSTM}_\text{A}\left(h_{\tau-1}^{arg}, x_\tau\right) \tag{9}$$

h_τ^{arg} and h_τ^{role} are hidden states at timestep τ of LSTM_A and LSTM_R. e_f represents the embedding of f, and $e_{r_{\tau-1}}$ represents the embedding of $r_{\tau-1}$.

As we want to identify the start and end positions of a_τ, namely i_τ^s and i_τ^e, we build two kinds of feature representations to extract boundary feature from h_τ^{arg} and e_f:

$$h_\tau^{STA} = \text{MLP}_\text{s}([h_\tau^{arg}; e_f]) \tag{10}$$

$$h_\tau^{END} = \text{MLP}_\text{e}([h_\tau^{arg}; e_f]) \tag{11}$$

The dimensions of both h_τ^{STA} and h_τ^{END} are the same as contextual representations $h_0, \ldots, h_{\tau-1}$, and the MLP in our experiment consists of two linear layers and a relu activation function in between.

3.3.1 Hierarchical pointer network

With two representations h_τ^{STA} and h_τ^{END}, we apply a hierarchical pointer network to identify i_τ^s and i_τ^e. The hierarchical pointer network contains two pointer networks as is shown in Figure 3. The hierarchical pointer network identifies the i_τ^s firstly and then identify the i_τ^e based on i_τ^s. If we identify them simultaneously, the i_τ^s and i_τ^e may sometimes

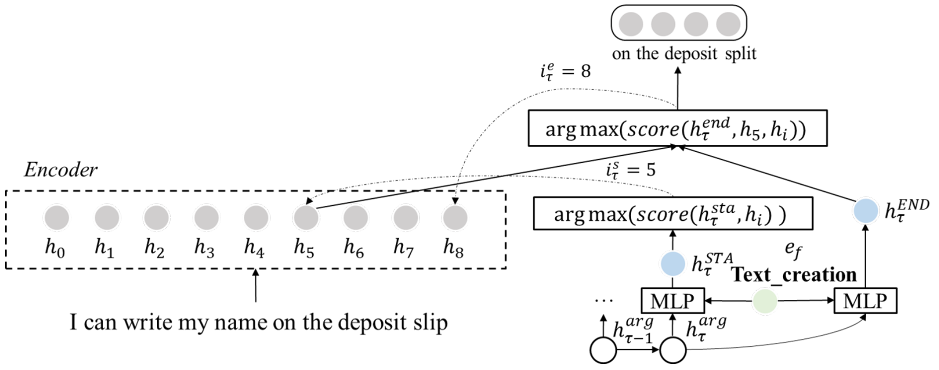
on the deposit split
$i_\tau^e = 8$
$\arg\max(score(h_\tau^{end}, h_5, h_i))$
Encoder
$i_\tau^s = 5$
$\arg\max(score(h_\tau^{sta}, h_i))$
h_τ^{END}
h_0 h_1 h_2 h_3 h_4 h_5 h_6 h_7 h_8
e_f
h_τ^{STA}
Text_creation
…
MLP
MLP
$h_{\tau-1}^{arg}$
h_τ^{arg}
I can write my name on the deposit slip

Figure 3: The prediction process of the argument ***on the deposit slip***.

be inconsistent. Besides, to avoid duplicate prediction, all spans in $a_0, \ldots, a_{\tau-1}$ are masked when identifying i_τ^s and i_τ^e. The prediction process is:

$$Score_{STA} = H^\top W_4 h_\tau^{STA} \tag{12}$$

$$P\left(i_\tau^s | S, t, f\right) = \text{softmax}(Score_{STA}) \tag{13}$$

$$i_\tau^s = \text{argmax}(P\left(i_\tau^s | S, t, f\right)) \tag{14}$$

$$Score_{END} = H^\top (W_5 h_\tau^{END} + W_6 h_{i_\tau^s}) \tag{15}$$

$$P\left(i_\tau^e | S, t, f\right) = \text{softmax}(Score_{END}) \tag{16}$$

$$i_\tau^e = \text{argmax}(P\left(i_\tau^e | S, t, f\right)) \tag{17}$$

H is $d_h \times n$ matrix $(h_0, \ldots, h_{n-1})$ that represents the encoder output of sentence S. W_4, W_5 and W_6 are $d_h \times d_h$ weight matrixes.

The hierarchical pointer network achieves arguments identification within linear computational complexity. For an n-tokens and k-arguments sentence, our model can identify the start and end positions of each argument with $O(2n)$ computational complexity, and $O(2n \cdot k)$ for all arguments.

3.4 Role Classification Module

The role classification module assigns semantic roles $r_0, \ldots, r_{k-1}$ to arguments $a_0, \ldots, a_{k-1}$. Assigning r_τ to a_τ also needs to consider the semantic information of $a_0, \ldots, a_{\tau-1}$ and their roles $r_0, \ldots, r_{\tau-1}$. Hence we use the same LSTM architecture named LSTM_R to record them. Both the contextual information of a_τ and the frame embedding e_f are used to predict r_τ:

$$y_\tau = W_7 \cdot [h_{i_\tau^e} + h_{i_\tau^s}; h_{i_\tau^e} - h_{i_\tau^s}; e_f] \tag{18}$$

$$h_\tau^{role} = \text{LSTM}_\text{R}(h_{\tau-1}^{role}, y_\tau) \tag{19}$$

$$P(r_\tau | S, t, f, a_\tau) = \text{MLP}([h_\tau^{role}; y_\tau]) \tag{20}$$

$h_{i_e} + h_{i_s}$ represents the boundary feature of a_τ and $h_{i_e} - h_{i_s}$ represents the inner feature of the span (Wang and Chang, 2016; Cross and Huang, 2016; Ouchi et al., 2018). With the probability distribution $P(r_\tau | S, t, f, a_\tau)$, we can predict the role r_τ.

Moreover, we add a special role 'None' at the final decoding step and let r_k be 'None'. During the inference stage, the role classification decoder and argument identification decoder will automatically stop when predicting 'None'.

4 Loss Function

We utilize cross-entropy loss to maximize the probability of the oracle frame type, span boundaries (start-end pair) and role types:

$$\mathscr{L}_{frame} = \log(P(\hat{f} | S, t)) \tag{21}$$

$$\begin{aligned} \mathscr{L}_{role} = \sum_{\tau=0}^{k-1} \log(P(\hat{r}_\tau | S, t, f, a_\tau)) + \\ \log(P(r_{None} | S, t, f, a_k)) \end{aligned} \tag{22}$$

$$\begin{aligned} \mathscr{L}_{span} = \sum_{\tau=0}^{k-1} \log(P(\hat{i}_\tau^s | S, t, f)) + \\ \sum_{\tau=0}^{k-1} \log(P(\hat{i}_\tau^e | S, t, f)) \end{aligned} \tag{23}$$

We optimize the losses of the three subtasks jointly:

$$\mathscr{L} = \alpha \mathscr{L}_{frame} + \beta \mathscr{L}_{span} + \gamma \mathscr{L}_{role} \tag{24}$$

α, β and γ are hyper-parameters that adjust the direction of training optimization.

5 Experiment

Dataset. We train and evaluate our model on FrameNet 1.5 dataset proposed by (Das and Smith, 2011) following previous work (Yang and Mitchell, 2017; Swayamdipta et al., 2017; Peng et al., 2018).

We also follow the same train/development/test split. Meanwhile, previous work adds the partially-annotated exemplar sentences (each exemplar sentence contains only one target). As is reported in previous work (Das et al., 2014; Yang and Mitchell, 2017; Swayamdipta et al., 2017), the exemplar sentences data can help to improve their models' performance. We add it as pre-train data for our model.

Pre-process. Previous work removes the argument spans longer than 20, which is a constraint that helps to reduce the computational complexity from $O(n^2)$ to $O(n)$. Though our model doesn't need such constraint because of better computational complexity, we hold the same setting as previous work for comparison. We also report the result that training our model without length constraint.

Setup. We train our models by two steps following previous work (Das et al., 2014). At first step, we pre-train our model with partially-annotated exemplar sentences data. Then we train the model on the offcial train set. We evaluate our model on development test and save the best performance model for test.

We use Glove (Pennington et al., 2014) to initialize the word embeddings, and average the existing embeddings for out-of-vocabulary words. We randomly initialize embeddings for part-of-speech tags, and token type tags. All the embeddings are learnable during training.

Other detail hyper-parameters are shown on Table 2.

Model. We compare our model with following previous models:

SEMAFOR: A widely known system(Chen et al., 2010) that uses a variety of syntactic features.

Framat: An open-source semantic role labeling tool proposed by Björkelund et al. (2010).

Framat+context: An extension version of Framat that adds extra context features by Roth and Lapata (2015).

Hermann et al.(2014): A frame identification model uses feature representation based on word embedding and WSABIE algorithm (Weston et al., 2011).

FitzGerald et al.(2015): A pipeline model that improves frame identification performance based on Hermann et al. (2014).

Open-SESAME: A pipeline model that predicts frame by FitzGerald et al. (2015) and designs a softmax-margin segmental RNN to improve argument identification.

Hyper-parameters	**Values**
Batch size	32
MLP layers	2
Encoder lstm layers	2
Word/lemma embedding	200
Token type embedding	100
POS embedding	64
Pre-train/train epochs	50 / 100
Pre-train/train optimizer	Adam
Activation Function	Relu
Encoder/Decoder hidden size	256
MLP/LSTM dropout rate	0.4 / 0.2
Pre-train/train learning rate	1e-4/6e-5
Learning rate decay	0.6 (every 30 epochs)
α, β, γ	0.1 / 0.3 / 0.3

Table 2: Details of hyperparameters (non-bert version).

Model	**All**	**Ambiguous**
SEMAFOR	83.6	69.2
Open-SESAME	87.0	-
Hartmann et al.	87.6	73.8
Yang and Mitchell	88.2	75.7
Hermann et al.	88.4	73.1
Peng et al.(BASIC)	89.2	76.3
Our Model	**89.4**	**76.7**
Our Model+*Bert*	**90.5**	**79.1**

Table 3: Frame identification accuracy result.

Yang and Mitchell (SEQ)(2017): A sequence tagging model for frame semantic parsing.

Yang and Mitchell (REL) (2017): A relation model that enumerates all possible spans and classify them.

Peng et al. (Basic) (2018): A single-task version of joint SRL model (without extra data). It is the current state of the art model on the task of frame semantic parsing.

5.1 Experiment Metrics And Result

We evaluate our model on the metrics of frame identification accuracy and full structure extraction. Note that previous systems may also report ensemble models based on different ensemble methods to improve models' performance, and the model (Peng et al., 2018) based on multi-task framework brings extra train data. For comparability, we only report the performance of single models that trained on framenet data only. We note that none of above-mentioned previous models are based on Bert which is widely applied in many NLP tasks. To explore the impact of Bert on frame semantic parsing, we implement a Bert-based version of our model and also report results of Bert-based model.

Model	**P**	**R**	**F1**
SEMAFOR	69.2	65.1	67.1
Framat	71.1	63.7	67.2
Framat+context	71.1	64.8	67.8
Open-SESAME	71.0	67.8	69.4
FitzGerald et al.	74.8	65.5	69.9
Yang and Mitchell (SEQ)	69.6	70.9	70.2
Yang and Mitchell(REL)	77.1	68.7	72.7
Peng et al.(BASIC)	**79.2**	71.7	75.3
Our Model	75.1	**76.9**	**76.0**
Our Model+*Bert*	78.2	**82.4**	**80.2**

Table 4: Full structure extraction result on the FN test set.

Model	**P**	**R**	**F1**
Our Model	75.1	76.9	76.0
wo interaction (Role&Arg)	75.6	76.3	75.9
wo interaction (Frame)	76.1	75.1	75.6
wo interaction (Both)	75.9	74.6	75.3

Table 5: Full structure extraction result of our models considering the effect of the interactions among decoders.

Frame Identification. The metrics of frame identification accuracy includes Ambiguous and All as is shown in Table 3. The ambiguous metrics evaluates targets evoking more than one possible frame in FrameNet and All evaluates all the targets. According to Peng et al. (2018), some previous studies' ambiguous lexical unit sets are not the same as the one from the official frame directory, which makes their results uncomparable. Therefore, it's fairer to use ALL to evaluate the performance of frame identification. Our model outperforms all previous models (0.2 point over SOTA). Our Ambiguous set is the same as Peng et al. (2018)'s and our model outperforms theirs by 0.4 point.

Full Semantic Structure Extraction. Full Semantic Structure Extraction is the metrics that measures the overall performance of Frame Semantic parsing. It requires exact match of arguments' boundaries and jointly evaluates the performance of frame identification, argument identification and role classification. (Baker et al., 2007) shows details of the metrics. Table 4 is the result. The first group contains pipeline models and the second group includes joint models. Our model shows improvement over all previous models (0.7 point over SOTA). We notice that our model greatly outperform state of the art models on Recall. We analyze that it's because the decoders of our model fully interact and make the current decision by considering all previous steps' decisions. Such strategy is likely to predict more complete arguments and roles.

Model	**P**	**R**	**F1**
Our Model	75.1	76.9	76.0
wo pre-train data	72.6	73.1	72.9
wo pre-train data/TAM	72.2	72.9	72.5

Table 6: Full structure extraction result of our models considering the influences of pre-train data and targets-aware attention mechanism (TAM).

Model	**P**	**R**	**F1**
Our Model	75.1	76.9	76.0
wo length constraint	75.1	77.6	76.3

Table 7: Full structure extraction result of our models considering the influence of the length constraint of arguments.

5.2 Ablation study

We train our model in different settings and evaluate them on the metrics of full structure extraction to measure their overall performance. We consider the influences of decoders' interactions, pre-train data (partially-annotated exemplar sentences), targets-aware attention mechanism (TAM) and the length constraint of arguments.

As mentioned before, all the decoders of our model interact with each other and the interactions are reflected in two aspects. To prove the effectiveness of them, we remove the interactive parts of the decoders respectively. Table 5 shows the results of models with following setting:

Without interaction (Arg&Role) means the interaction between argument identification decoder and role classification decoder is deleted.

Without interaction (Frame) means the identified frame information in frame identification decoder is not accessible to the other decoders.

Without interaction (Both) represents that both of the interactions above are removed.

As is shown in Table 5, the performances of the models all drop in varying degrees without any kind of interactive parts. The effect of interaction (Frame) is more pronounced than interaction (Arg&Role) (73.6 to 73.9). The model's performance without both kinds of interactive parts will drop from 76.0 to 75.3. And we noticed that the

Error Type	Description	Proportion(%)	
		Our Model	Peng
Frame error	Frame misprediction	10.5	11.3
Role error	Matching span with incorrect role.	22.0	12.6
Span error	Matching role with incorrect span boundary.	14.1	11.4
Extra predicted arg.	Predicted argument that doesn't overlap any gold argument	20.0	18.6
Missing arg.	Gold argument that doesn't overlap any predicted argument	33.4	43.5

Table 8: Error analysis result on FrameNet development set.

recalling rate of our model is decreasing with the remove of interactions. It verifies the previous analysis that the interactions among decoders make our model predict in a global view and thus our model is likely to predict more complete arguments and roles.

Table 6 shows the performance differences in the influences of pre-train data and targets-aware attention mechanism. The performance of the model without the pre-train step drops by 3.1 points on F1, which demonstrates that adding pre-train data is beneficial to model's performance. The result is consistent with the conclusion of Yang and Mitchell (2017) that models will get a 3-4 points increase on F1 if adding partially-annotated exemplar sentences. Also, we consider the influence of the targets-aware mechanism. The targets-attention mechanism doesn't work at pre-train step because the partially-annotated exemplar sentences only contains one target per sentence. To eliminate the interference of the gap between the train data and the pre-train data, we hold the same setting of skipping the pre-train step. As shown in Table 6, the model without TAM drops from 72.9 to 72.5 on F1. It proves that the targets-aware attention mechanism contributes to overall performance of frame semantic parsing.

As mentioned before, our model has an advantage in terms of computational complexity. We remove the length constraint of spans which is adopted in previous work. Table 7 shows the result. We notice that our model has a slight increase without length constraint of arguments. The result shows that our model get benefit with complete data. Moreover, it proves that the hierarchical pointer network is good at capturing long distance dependency relation. We encourage future work to train and evaluate on complete data if their computational complexity allows.

5.3 Error Analysis

We follow the error analysis method of Peng et al. (2018) and compare our model with theirs. Table 8 shows the proportions of five error types. Though missing arguments is the major error for both of our model and Peng et al. (2018), our model shows a great decrease by 10.1%, which proves that our model prefers to predict more complete arguments and is more likely to overlap gold arguments. However, it correspondingly brings increases on extra predicted arguments, Role error and Span error. We analyze that it's because our model captures the relation between different roles and is able to make current decision by considering all previous steps' action information, it prefers to predict the role which is related to previous predicted roles. Such strategy is more likely to overlap gold arguments. However, it may also predict more arguments and roles than the ground truth.

6 Conclusion

We design a multi-decoder framework to process all the subtasks of frame semantic parsing jointly. The multi-decoder framework strengthens the interactions among these three tasks. Our model works in an alternate way, which predicts the argument and the role by considering all previous decisions. We apply a hierarchical pointer network which achieves the argument identification with linear computational complexity. Experiments show improvement over state of the art models.

Acknowledgments

The authors thank the reviewers for their thorough feedback and their suggestions. This work is supported by the National Science Foundation of China under Grant No.61936012, the National Key R&D Program of China under Grand No.2020AAA0106700, and the National Science Foundation of China under Grant No. 61876004.

References

Collin F Baker, Michael Ellsworth, and Katrin Erk. 2007. Semeval-2007 task 19: Frame semantic structure extraction. In *Proceedings of the Fourth*

International Workshop on Semantic Evaluations (SemEval-2007), pages 99–104.

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90.

Anders Björkelund, Bernd Bohnet, Love Hafdell, and Pierre Nugues. 2010. A high-performance syntactic and semantic dependency parser. In *Coling 2010: Demonstrations*, pages 33–36.

Desai Chen, Nathan Schneider, Dipanjan Das, and Noah A Smith. 2010. Semafor: Frame argument resolution with log-linear models. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 264–267.

James Cross and Liang Huang. 2016. Span-based constituency parsing with a structure-label system and provably optimal dynamic oracles. *arXiv preprint arXiv:1612.06475*.

Dipanjan Das, Desai Chen, André FT Martins, Nathan Schneider, and Noah A Smith. 2014. Frame-semantic parsing. *Computational linguistics*, 40(1):9–56.

Dipanjan Das, Nathan Schneider, Desai Chen, and Noah A Smith. 2010. Probabilistic frame-semantic parsing. In *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics*, pages 948–956.

Dipanjan Das and Noah A Smith. 2011. Semi-supervised frame-semantic parsing for unknown predicates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1435–1444.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Nicholas FitzGerald, Oscar Täckström, Kuzman Ganchev, and Dipanjan Das. 2015. Semantic role labeling with neural network factors. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 960–970.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational linguistics*, 28(3):245–288.

Silvana Hartmann, Ilia Kuznetsov, M Teresa Martín-Valdivia, and Iryna Gurevych. 2017. Out-of-domain framenet semantic role labeling. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 471–482.

Karl Moritz Hermann, Dipanjan Das, Jason Weston, and Kuzman Ganchev. 2014. Semantic frame identification with distributed word representations.

Richard Johansson and Pierre Nugues. 2007. Lth: semantic structure extraction using nonprojective dependency trees. In *Proceedings of the fourth international workshop on semantic evaluations (SemEval-2007)*, pages 227–230.

Hiroki Ouchi, Hiroyuki Shindo, and Yuji Matsumoto. 2018. A span selection model for semantic role labeling. *arXiv preprint arXiv:1810.02245*.

Hao Peng, Sam Thomson, Swabha Swayamdipta, and Noah A Smith. 2018. Learning joint semantic parsers from disjoint data. 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 1492–1502.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Michael Roth and Mirella Lapata. 2015. Context-aware frame-semantic role labeling. *Transactions of the Association for Computational Linguistics*, 3:449–460.

Swabha Swayamdipta, Sam Thomson, Chris Dyer, and Noah A Smith. 2017. Frame-semantic parsing with softmax-margin segmental rnns and a syntactic scaffold. *arXiv preprint arXiv:1706.09528*.

Wenhui Wang and Baobao Chang. 2016. Graph-based dependency parsing with bidirectional lstm. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2306–2315.

Jason Weston, Samy Bengio, and Nicolas Usunier. 2011. Wsabie: Scaling up to large vocabulary image annotation.

Bishan Yang and Tom Mitchell. 2017. A joint sequential and relational model for frame-semantic parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1247–1256.