A Graph-Based Neural Model for End-to-End Frame Semantic Parsing

Zhichao Lin[1], **Yueheng Sun**[2], **Meishan Zhang** [1*]
[1]School of New Media and Communication, Tianjin University, China
[2]College of Intelligence and Computing, Tianjin University, China
{chaosmyth,yhs,zhangmeishan}@tju.edu.cn

Abstract

Frame semantic parsing is a semantic analysis task based on FrameNet which has received great attention recently. The task usually involves three subtasks sequentially: (1) target identification, (2) frame classification and (3) semantic role labeling. The three subtasks are closely related while previous studies model them individually, which ignores their intern connections and meanwhile induces error propagation problem. In this work, we propose an end-to-end neural model to tackle the task jointly. Concretely, we exploit a graph-based method, regarding frame semantic parsing as a graph construction problem. All predicates and roles are treated as graph nodes, and their relations are taken as graph edges. Experiment results on two benchmark datasets of frame semantic parsing show that our method is highly competitive, resulting in better performance than pipeline models.

1 Introduction

Frame semantic parsing (Gildea and Jurafsky, 2002) aims to analyze all sentential predicates as well as their FrameNet roles as a whole, which has received great interest recently. This task can be helpful for a number of tasks, including information extraction (Surdeanu et al., 2003), question answering (Shen and Lapata, 2007), machine translation (Liu and Gildea, 2010) and others (Coyne et al., 2012; Chen et al., 2013; Agarwal et al., 2014). Figure 1 shows an example, where all predicates as well as their semantic frame and roles in the sentence are depicted.

Previous studies (Das et al., 2014; Swayamdipta et al., 2017; Bastianelli et al., 2020) usually divide the task into three subtasks, including target identification, frame classification and semantic role labeling (SRL), respectively. By performing the three subtasks sequentially, the whole frame semantic parsing can be accomplished. The majority of works focus on either one or two of the three subtasks, treating them separately (Yang and Mitchell, 2017; Botschen et al., 2018; Swayamdipta et al., 2018; Peng et al., 2018).

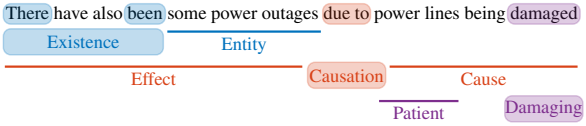


Figure 1: An example involving frame semantic structures, taken from the FrameNet (Baker et al., 1998). Frame-evoking predicates are highlighted in the sentence, and corresponding frames are shown in colored blocks below. The frame-specific roles are underlined with their frames in the same row.

The above formalization has two weaknesses. First, the individual modeling of the three subtasks is inefficient to utilize the relationship among them. Apparently, the earlier subtasks can not exploit the information from their future subtasks. Second, the pipeline strategy can suffer from the error propagation problem, where the errors occurring in the previous subtasks can influence the later subtasks as well. To address the two weaknesses, end-to-end modeling is one promising alternative, which has been widely adopted in natural language processing (NLP) (Cai et al., 2018; He et al., 2018; Sun et al., 2019; Fu et al., 2019; Fei et al., 2020).

In this work, we propose a novel graph-based model to tackle frame semantic parsing in an end-to-end way, using a single model to perform the three subtasks jointly. We organize all predicates and their FrameNet semantic by a graph, and then design an end-to-end neural model to construct the graph incrementally. An encoder-decoder model is presented to achieve the graph building goal, where the encoder is equipped with contextualized BERT representation (Devlin et al., 2019), and the decoder includes node generation and edge building sequentially. Our final model is elegant and easy to

*Corresponding author.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 3864–3874
November 7–11, 2021.

understand as a whole.

We conduct experiments on two benchmark datasets to evaluate the effectiveness of our proposed model. First, we study our graph-based framework in two settings, the end-to-end scenario and the pipeline manner, where the node building and edge building are trained separately. Results show that end-to-end modeling is much better. Besides, we also compare our model with several other pipelines, where the similar findings can be observed. Second, we compare our graph-based framework with previous methods by the three subtasks individually, finding that the graph-based architecture is highly competitive. We can obtain the best performance in the literature, leading to a new state-of-the-art result. Further, we conduct extensive analyses to understand our method in depth.

In summary, we make the following two major contributions in this work:

(1) We propose a novel graph-based model for frame semantic parsing which can achieve competitive results for the end-to-end task as well as the individual subtasks.

(2) To the best of our knowledge, we present the first work of end-to-end frame semantic parsing to solve all included subtasks together in a single model.

We will release our codes as well as experimental setting public available on https://github.com/Ch4osMy7h/FramenetParser to help result reproduction and facilitate future researches.

2 Related Work

Frame-Semantic Parsing Frame-semantic parsing has been received great interest since being released as an evaluation task of SemEval 2007 (Baker et al., 2007). The task attempts to predict semantic frame structures defined in FrameNet (Baker et al., 1998) which are composed of frame-evoking predicates, their corresponding frames and semantic roles. Most of the previous works (Das et al., 2014; Swayamdipta et al., 2017; Bastianelli et al., 2020) focus on a pipeline framework to solve the task, training target identification, frame classification and semantic role labeling models separately. In this work, to the best of our knowledge, we present the first end-to-end model to handle the task jointly.

Among the three subtasks of frame semantic parsing, semantic role labeling has been researched most extensively (Kshirsagar et al., 2015; Yang and Mitchell, 2017; Peng et al., 2018; Swayamdipta et al., 2018; Marcheggiani and Titov, 2020). It is also highly related to the Propbank-style semantic role labeling (Palmer et al., 2005) as while with only differences in the frame definition. Thus the models between the two types of semantic role labeling can be mutually borrowed. There are several end-to-end Propbank-style semantic role labeling models as well (Cai et al., 2018; He et al., 2018; Li et al., 2019; Fu et al., 2019). However, these models are difficult to be applied directly for frame semantic parsing due to the additional frame classification as well as the discontinuous predicates. In this work, we present a totally-different graph construction style model to solve end-to-end frame semantic parsing elegantly.

Graph-Based Methods Recently, graph-based methods have been widely used in a range of other tasks, such as dependency parsing (Dozat and Manning, 2016; Kiperwasser and Goldberg, 2016; Ji et al., 2019), AMR parsing (Flanigan et al., 2014; Lyu and Titov, 2018; Zhang et al., 2019a,b) and relation extraction (Sun et al., 2019; Fu et al., 2019; Dixit and Al-Onaizan, 2019). In this work, we aims for frame semantic parsing, organizing the three included subtasks by a well designed graph, converting it into graph-based parsing task naturally.

3 Method

3.1 Task Formulation

The goal of frame-semantic parsing is to extract semantic predicate-argument structures from texts, where each predicate-argument structure includes a predicate by a span of words, a well-defined semantic frame to express the key roles of the predicate, and the values of these roles by word spans. Formally, given by a sentence X with n words $w_1, w_2, \ldots, w_n$, frame-semantic parsing aims to output a set of tuples $\mathcal{Y} = \{(y_1, y_2, \ldots, y_k)\}_{k=1}^{K}$, where each y_i consists of the following elements:

- $p_i = (p_{i,1}, \ldots, p_{i,d_i})$, where $p_{i,*}$ are word spans in X and d_i indicates the number of pieces of the predicate since it might be discontinuous.
- $f_i \in \mathcal{F}$, where $\mathcal{F}$ is the frame set which is well defined in FrameNet.[1]

[1]The Berkeley FrameNet (Baker et al., 1998) project provides a lexicon of semantic frames. Specifically, the FrameNet 1.5 is with 1020 lexicalized frames, while in 1.7, the number is 1221.

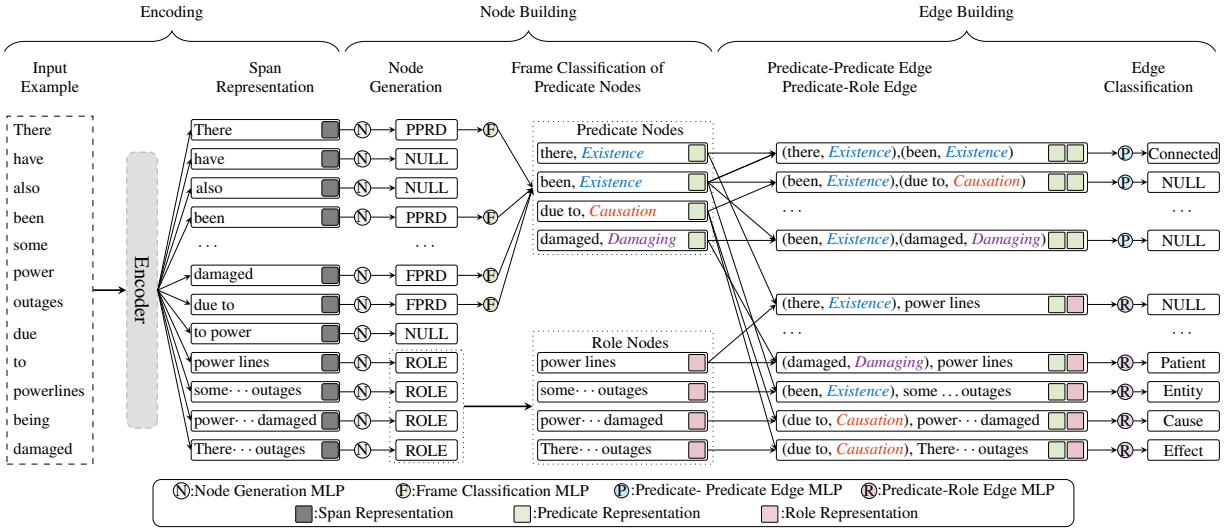


Figure 2: The overall architecture of our graph-based end-to-end model.

- $r_i = ([r_{i,1}, v_{i,1}], \ldots, [r_{i,m_k}, v_{k,m_k}])$, where $r_{i,*}$ are frame roles derived from f_i and $v_{i,*}$ are also word spans in X.

The full frame semantic parsing is usually divided into the following three subtasks:

- **Target Identification** (also known as predicate identification), which is to identify all valid frame-evoking predicates from X, outputting $P = \{(p_1, ..., p_k)\}$.
- **Frame Classification**, which is to predicate the concrete evoking frame f_i of a certain predicate $p_i \in P$.
- **Semantic Role Labeling**, which is to assign concrete values for roles r_i by given a predicate frame pair (p_i, f_i).

Previously, the majority of work of frame semantic parsing performs the three subtasks individually, ignoring their highly-related connections and also being vulnerable to the error propagation problem. Thus, we present an end-to-end graph-based model to accomplish the three subtasks by a single model.

3.2 The Graph-Based Methodology

We formalize the frame-semantic parsing task as a graph constructing problem, and further present an encoder-decoder model to perform the task in an end-to-end way. The encoder aims for representation learning of the frame semantic parsing, and the decoder constructs the semantic graph incrementally. Concretely, for the encoder, we compute the span representations since the basic processing units of our model are word spans, and for the decoder, we first generate all graph nodes, and then build edges among the graph nodes. Figure 2 shows the overall architecture of our method.

3.2.1 Encoding

Due to the strong capability of BERT (Devlin et al., 2019) for represent learning, we adopt it as the backbone of our model. Given a sentence $X = \{w_1, w_2, ..., w_n\}$, BERT converts each word w_i into word pieces, and feed them into deep transformer encoders to get the piece-level representation. To obtain word-level representation, we average all piece vectors of word w_i as its final representation $\mathbf{e}_i$.

For further feature abstraction, we exploit BiHLSTM (Srivastava et al., 2015) to compose high-level features based on word-level output $\mathbf{e}_1, \cdots, \mathbf{e}_n$, following Swayamdipta et al. (2018):

$$\mathbf{h}_1, \cdots, \mathbf{h}_n = \text{BiHLSTM}(\mathbf{e}_1, \cdots, \mathbf{e}_n), \quad (1)$$

where the gated highway connections are applied to BiLSTMs.

Span Representation We enumerate all possible spans $S = \{s_1, s_2, \ldots, s_m\}$ in a sentence and limit the maximum span length to L. Then, each span $s_i \in S$ is represented by:

$$\mathbf{g}_i = [\mathbf{h}_{\text{START}(i)}; \mathbf{h}_{\text{END}(i)}; \mathbf{h}_{\text{ATTN}}; \phi(s_i)], \quad (2)$$

where $\phi(g_i)$ represents the learned embeddings of span width features, $\mathbf{h}_{\text{ATTN}}$ is computed by self-attention mechanism which weights the corresponding vector representations of the words in the span by normalized attention scores, and $\text{START}(i)$ and $\text{END}(i)$ denote start and end indices of s_i.

3.2.2 Node Building

Node Generation We exploit a preliminary classification to achieve the goal of node generation. First, a span can be either a graph node or not. Further, a graph node can be a full or partial predicate node, and the node can also be a role node. Totally, we define four types for a given span:

- `FPRD`: a full predicate span.
- `PPRD`: a partial predicate span.
- `ROLE`: a role span.
- `NULL`: a span that is not a graph node.

The type of a span can be the full permutation of elements in set $\{\texttt{FPRD}, \texttt{PPRD}, \texttt{ROLE}\}$ or `NULL`. Thus, each span can be classified into eight types (i.e., `FPRD`, `PPRD`, `ROLE`, `FPRD-PPRD`, `FPRD-ROLE`, `PPRD-ROLE`, `FPRD-PPRD-ROLE`, `NULL`).

Given an input span s_i with its vectorial representation as $\mathbf{g}_i$, we exploit one MLP layer with softmax to classify the span type:

$$\mathbf{p}_n = \text{softmax}(\text{MLP}_n(\mathbf{g}_i)), \quad (3)$$

where $\mathbf{p}_n$ indicates the probabilities of span types. By this classification, all non-null type spans are graph nodes, reaching the goal of node generation.

Frame Classification of Predicate Nodes Node generation detects all graph nodes roughly, assigning each node with a single label to indicate whether it can be served as a predicate or role. Here we go further to recognize the semantic frames for all predicate nodes, which could be regarded as an in-depth analysis for node attribution. The step is corresponding to the frame classification subtask.

Given an input span s_i of a predicate node (`FPRD` or `PPRD`), assuming its representation being $\mathbf{g}_i$, we use another MLP layer together with softmax to output the probabilities of each candidate frame for the predicate node:

$$\mathbf{p}_c = \text{softmax}(\text{MLP}_c(\mathbf{g}_i)), \quad (4)$$

where $\mathbf{p}_c$ is the output probabilities of semantic frames. Specially, frames are constrained by the lexical units defined in FrameNet. For example, the predicate with the lexical unit "meeting" only evokes frame *Social_event* and *Discussion*.

We also adopt the pseudo strategy following Swayamdipta et al. (2017) to optimize the classification. First, we use spacy lemmatizer (Honnibal et al., 2020) to translate an input sentence into lemmas. Then, if a word span is a predicate node, we treat the corresponding lemma span as the pseudo lexical unit and index the corresponding semantic frame set by it. Finally, we reduce the search space by masking frames outside the set. In our experiments, we find it is practical to apply this strategy.

3.2.3 Edge Building

After graph nodes are ready, we then build edges to accomplish frame semantic parsing accordingly. There are two types of edges in our model.

Predicate-Predicate Edge For extracting discontinuous mentions, we build the edges between nodes which are predicate fragments (i.e., `PPRD` nodes). In detail, we treat it as a binary classification problem considering whether two nodes alongside the edge can form parts of a predicate or not. Formally, given two `PPRD` nodes with the corresponding spans $\mathbf{s}_i^p$ and $\mathbf{s}_j^p$ and their encoding representations $\mathbf{g}_i^p$ and $\mathbf{g}_j^p$, we utilize one MLP layer to classify their edge type:

$$\mathbf{p}_{pe} = \text{softmax}(\text{MLP}_{pe}([\mathbf{g}_i^p, \mathbf{g}_j^p, \mathbf{g}_i^p * \mathbf{g}_j^p])), \quad (5)$$

where $\mathbf{p}_{pe}$ indicates the probabilities of two types, namely `Connected` and `NULL` (i.e., cannot be connected), and the feature representation is borrowed from Zhao et al. (2020).

Predicate-Role Edge For extracting frame-specific roles, we build the edges between predicates nodes (i.e., node type by `FPRD` or `PPRD`) and role nodes (i.e., node type by `ROLE`). Given a predicate node $\mathbf{s}_i^p$ and a role node $\mathbf{s}_j^r$, assuming their neural representations being $\mathbf{g}_i^p$ and $\mathbf{g}_j^r$, respectively, we utilize another MLP layer to determine their edge type by multi-class classification:

$$\mathbf{p}_{re} = \text{softmax}(\text{MLP}_{re}([\mathbf{g}_i^p, \mathbf{g}_j^r, \mathbf{g}_i^p * \mathbf{g}_j^r])), \quad (6)$$

where $\mathbf{p}_{re}$ indicates the probabilities of predicate-role edge types (i.e., frame roles as well as a `NULL` label indicating no relation).

3.3 Joint Training

To train the joint model, we employ the negative log-likelihood loss function for both node building and edge building step:

$$\begin{aligned} \mathcal{L}_n &= -\sum \log \mathbf{p}_n(y_n) - \sum \log \mathbf{p}_c(y_c) \\ \mathcal{L}_e &= -\sum \log \mathbf{p}_{pe}(y_{pe}) - \sum \log \mathbf{p}_{re}(y_{re}), \end{aligned} \quad (7)$$

where y_n and y_c are the gold labels for the text spans and predicate nodes, y_{pe} and y_{re} indicate the gold edge labels for the predicate-predicate and predicate-role node pairs. Further, losses from two steps are summed together, leading to the final training objective of our model:

$$\mathcal{L} = \mathcal{L}_n + \mathcal{L}_e \quad (8)$$

3.4 Decoding

The decoding aims to derive frame semantic parsing results by the graph-based model. Here we describe the concrete process by the three subtasks.

Target Identification The target identification involves both node building and edge building steps. First, all predicate nodes with type `FPRD` are predicates. Second, there is a small percentage of predicates composed of multiple nodes with type `PPRD`. If two or more such nodes are connected with predicate-predicate edges, we regard these nodes as one single valid predicate .

Frame Classification The frame classification decoding is performed straightforwardly for single-node predicates. For multi-node predicates, there may exist conflicts from the frame classification of different nodes. Concretely, given a multi-node predicate composed of two or more nodes, the max-scored frame evoked by them might be different. Thus, to address this issue, we use the maximum operation achieved by first summing up the softmax distributions over all covered nodes and then fetching the max-scored frame.

Semantic Role Labeling The condition of semantic role labeling is similar to frame classification. For the single-node predicates, the semantic role labeling output is determinative. For the multi-node predicates, we assign role values for the candidate roles inside its predicted frame only, and further select the concrete role node, which is the highest-probability to the covered predicate nodes.

4 Experiments

4.1 Setting

Dataset We adopt the FrameNet versions 1.5 and 1.7[2] (denoted by FN1.5 and FN1.7 for short, respectively) as the benchmark datasets to evaluate our models. FN1.5 is the widely used dataset in previous work and FN1.7 is the latest version used recently which involves more semantics. We follow the previous studies (Das et al., 2014; Swayamdipta et al., 2017) to divide the two datasets into the training, validation and test sets, respectively. Table 1 shows the overall data statistics.

Dataset	Type	Train	Dev	Test
FN1.5	# Sentence	2,713	326	982
	# Predicate	16,618	2,282	4,427
	# Role	29,449	4,039	7,146
FN1.7	# Sentence	3,413	326	1,354
	# Predicate	19,384	2,270	6,714
	# Role	34,385	4,024	11,303

Table 1: Statistics of the datasets.

Evaluation We measure the performance of frame semantic parsing by its three subtasks, respectively. For target identification, we treat a predicate as correct only when all its included word spans exactly match with the gold-standard spans of the predicate. For frame classification, we use the joint performance for evaluation, regarding a classification as correct only when the predicate, as well as the frame, are both correct. For semantic role labeling, we also use the joint performance regarding the role as correct when the predicate, role span (exact match), and role type are all correct, which is treated as our major metric.

Derived Models Following previous studies and our graph-based method, we can derive a range of basic models for comparisons:

- **Node**, the node building submodel which is the first step of our decoder module mentioned in section 3.2.2.
- **Edge**: the edge building submodel which is the second step of our decoder module mentioned in section 3.2.3.
- **Predicate**, a graph-based predicate identification model, which is implemented by keeping only the predicate node generation and predicate-predicate edge building in our final graph-based model.
- **Frame**, our final graph-based model with only the frame classification submodel, assuming predicate nodes and their edges are given.
- **Role**, our graph-based model with only role node generation and predicate-role edge building, assuming predicate nodes and their frames are given.
- **Predicate∘Frame**, a joint model of Predicate

[2] https://framenet.icsi.berkeley.edu/fndrupal/

Data	Model	Target			Frame			Role		
		P	R	F1	P	R	F1	P	R	F1
FN1.5	Predicate+Frame+Role	73.17	75.47	74.30	66.08	68.15	67.06	45.91	46.70	46.30
	Predicate∘Frame+Role	74.32	**76.16**	75.23	67.91	68.78	68.34	**46.85**	47.89	47.36
	Predicate+Frame∘Role	73.17	75.47	74.30	67.73	68.56	68.14	46.51	49.01	47.72
	Predicate∘Frame+Semi-CRF	74.32	**76.16**	75.23	67.91	68.78	68.34	46.33	**50.87**	**48.50**
	Node+Edge	**74.99**	75.85	**75.42**	**68.01**	**68.79**	**68.40**	46.64	49.35	47.96
	Ours-Joint	**75.81**	**76.17**	**75.99**	**68.72**	**69.05**	**68.89**	**47.79**	**50.60**	**49.16**
FN1.7	Predicate+Frame+Role	**78.62**	69.92	74.02	**71.05**	63.06	66.82	49.32	45.60	47.38
	Predicate∘Frame+Role	76.79	**72.92**	**74.81**	69.29	66.16	**67.69**	49.21	46.03	47.57
	Predicate+Frame∘Role	**78.62**	69.92	74.02	69.79	65.06	67.34	49.46	46.01	47.67
	Predicate∘Frame+Semi-CRF	76.79	**72.92**	**74.81**	69.29	66.16	**67.69**	49.03	**47.49**	**48.24**
	Node+Edge	76.81	72.54	74.62	68.99	**66.33**	67.63	**49.55**	46.28	47.86
	Ours-Joint	76.16	**74.98**	**75.56**	69.39	**68.30**	**68.84**	49.09	**48.81**	**48.95**

Table 2: Main results of frame-semantic parsing on FN1.5 and FN1.7, where the pipeline and end-to-end methods are compared thoroughly.

and Frame, which is implemented by excluding the role node generation and predicate-role edge building in our final model.

- **Frame∘Role**, a joint model of Frame and Role, which is implemented by excluding the predicate node generation and predicate-predicate edge building in our final model.
- **Semi-CRF**, a span-level semi-Markov CRF (Sarawagi and Cohen, 2005) model for semantic role labeling which is borrowed from Swayamdipta et al. (2018), where the only difference is that we use BERT as the representation layer for fair comparisons.[3]

Note that the above derived models are trained individually. Based on these models, we can build five pipeline systems: (1) **Predicate + Frame + Role**, (2) **Predicate∘Frame + Role**, (3) **Predicate + Frame∘Role**, (4) **Predicate∘Frame + Semi-CRF**, and (5) **Node + Edge**, which are exploited for comparisons with our graph-based end-to-end model.

Hyperparameters All our codes are based on Allennlp Library (Gardner et al., 2017) and trained on a single RTX-2080ti GPU. We choose the BERT-base-cased[4], which consists of 12-layer transformers with the hidden size 768 for all layers. We set all the hidden sizes of BiHLSTM to 200, and the number of layer to 6. The MLP layers are of dimension size by 150 and depth by 1, with ReLU function. We apply dropouts of 0.4 to BiHLSTM and 0.2 to MLP layers. Following Swayamdipta et al. (2018), we also limit the maximum length of spans to 15 for efficiency, resulting in oracle recall of 95% on the development set.

For training, we exploit online batch learning with a batch size of 8 to update the model parameters, and use the BertAdamW algorithm with the learning rate 1×10^{-5} to finetune BERT and 1×10^{-3} to fine-tune other parts of our model. The gradient clipping mechanism by a maximum value of 5.0 is exploited to avoid gradient explosion. The training process are stopped early if the performance does not increase by 20 epochs.

4.2 Main Results

Table 2 shows the main results on the test sets of FN1.5 and FN1.7 datasets respectively, where our end-to-end model is compared with the four strong pipeline methods mentioned in Section 4.1. We can see that the end-to-end joint model can lead to significantly better performance (p-value below 10^{-5} by pair-wise t-test) as a whole on both datasets. Concretely, we can obtain average improvements of $\frac{0.57+0.75}{2} = 0.66$ on target identification, $\frac{0.49+1.15}{2} = 0.82$ on frame classification, and $\frac{0.66+0.71}{2} = 0.69$ on semantic role labeling on the two datasets compared with the best results of the pipeline systems, respectively.

Besides the overall advantage of the end-to-end joint model over the pipelines, we can also find that the joint of two subtasks can also outperform their counterpart baselines. Concretely, as shown in Table 2, Predicate∘Frame is better than Predicate + Frame, and Frame∘Role is better than Frame + Role. The results further indicate the effectiveness

[3] According to the preliminary experiments, we find that the fine-tuned method of BERT usage would hurt the Semi-CRF model performance. Therefore, we freeze the BERT parameters for Semi-CRF here.

[4] https://github.com/google-research/bert

Model	FN1.5	FN1.7
Das et al. (2014)	45.40	-
Swayamdipta et al. (2017)	73.23	**73.25**
Bastianelli et al. (2020) (wo syntax)	74.96	-
Bastianelli et al. (2020) (w syntax)	**76.80**	-
Predicate	76.09	75.34
Predicate∘Frame	76.47	75.88
Ours-Joint	**76.90**	**76.27**

Table 3: Target Identification Results.

Model	FN1.5	FN1.7
Das et al. (2014)	83.60	-
Hermann et al. (2014)	88.41	-
Hartmann et al. (2017)	87.63	-
Yang and Mitchell (2017)	88.20	-
Swayamdipta et al. (2017)	86.40	86.55
Botschen et al. (2018)	88.82	-
Peng et al. (2018)	**90.00**	**89.10**
Bastianelli et al. (2020) (wo syntax)	89.90	-
Bastianelli et al. (2020) (w syntax)	89.83	-
Frame	90.16	90.34
Ours-Joint	**90.62**	**90.64**

Table 4: The accuracy of the Frame Identification task based on the gold targets.

Model	FN1.5	FN1.7
Das et al. (2014)	59.10	-
Kshirsagar et al. (2015)	63.10	-
Yang and Mitchell (2017)	65.50	-
Swayamdipta et al. (2017)	59.48	**61.36**
Swayamdipta et al. (2018)	69.10	-
Marcheggiani and Titov (2020)	69.30	-
Bastianelli et al. (2020) (wo syntax)	72.85	-
Bastianelli et al. (2020) (w syntax)	**75.56**	-
Semi-CRF	**73.56**	**72.22**
Ours-Joint	73.28	72.06

Table 5: Pipeline Semantic Role Labeling results using gold targets and frames.

of joint learning. Further, by comparisons between our graph-based Role model and the Semi-CRF one, we can see that the Semi-CRF is better. The reason could be that the Semi-CRF model can exploit higher-order features among different frame roles which are ignored by our simple edge building module. As our edge building considers all predicates and all roles together, the incorporation of such features is still with great inconveniences.

4.3 Individual Subtask Evaluation

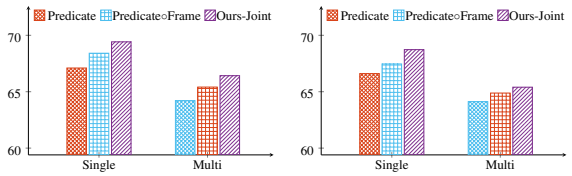
Previous studies commonly focus on only individual subtasks of frame semantic parsing. In order to compare with these studies, we simulate the scenarios by imposing constraints with gold-standard inputs in our joint models. In this way, we show the capability of our models on individual tasks.[5] In particular, Bastianelli et al. (2020) report the best performance of the previous studies in the literature, which is based on BERT representations. They adopt the constituency syntax which can boost the individual model performances significantly. Since our final model uses no other knowledge except BERT, we report their model performances by with syntax (denoted as w syntax) and without syntax (denoted as wo syntax) for careful comparisons.

Target Identification We show the performance of previous studies on target identification in Table 3, and also report the results of three-related models derived from this work. First, by comparing our three models (i.e. predicate only, predicate with frame, and the full graph parsing), the results show that both frame classification and semantic role labeling can help target identification. Second, we can see that our final model can achieve the best performance among the previous work.

Frame Classification Noted that we have Table 4 shows the result of individual frame classification tasks, where all systems assume gold-standard predicates as inputs. Similar to target identification, we can achieve better performance than all previous studies. Peng et al. (2018) did not use BERT, but they use extra datasets from FrameNet (exemplar sentences) and semantic dependency parsing, which can also benefit our task greatly. As for the comparison between our implemented two models, Frame alone and our final joint model, the results show that semantic role labeling can benefit the frame classification, which is reasonable.

Semantic Role Labeling Table 5 shows the results of various models on the semantic role labeling task. By constraining gold-standard predicates and frames to the outputs, our model degenerates to a normal semantic role labeling model. We also give the result by using Semi-CRF. As shown, our final semantic role labeling model is highly competitive in comparison with previous studies, except

[5]The results are obtained by the scripts from Swayamdipta et al. (2017).



(a) FN1.5 (b) FN1.7

Figure 3: F1 scores of frame metric in different predicates. **Single**: single-word predicate. **Multi**: multi-word predicate.

Model	Node	Frame	Edge
Predicate+Frame+Role	64.17	68.39	50.14
Predicate∘Frame+Role	64.32	68.89	50.97
Predicate+Frame∘Role	64.24	68.97	51.09
Node+Edge	**64.56**	**69.30**	**51.43**
Ours-Joint	**65.34**	**69.80**	**52.13**

Table 6: F1 score results of the modules.

Model	Target	Frame	Role
Predicate+Frame+Role	74.86	67.28	47.34
Predicate∘Frame+Role	**75.11**	**67.78**	47.81
Predicate+Frame∘Role	74.86	67.57	47.93
Predicate∘Frame+Semi-CRF	**75.11**	**67.78**	**48.26**
Ours-Joint	**75.72**	**68.86**	**48.89**

Table 7: F1 score results of our proposed method ignoring discontinuous predicates.

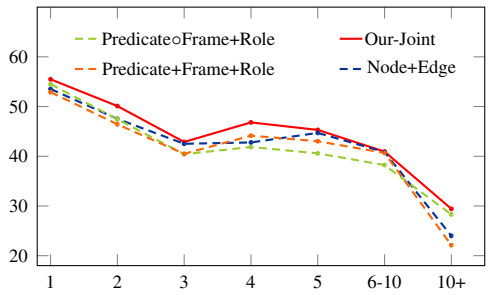


Figure 4: F1 scores of roles in terms of the length.

the model of Bastianelli et al. (2020) with syntax. The exception is expected, since syntax has been demonstrated highly effective before for SRL (Swayamdipta et al., 2018; Peng et al., 2018; Bastianelli et al., 2020). In addition, the Semi-CRF model is better than our method, which is consistent with the results in Table 2.

4.4 Discussion

In this subsection, we conduct detailed experimental analyses for better understanding our graph-based methods. Note that if not specified, the analyses are based on the FN1.7 dataset, which has a larger scale of annotations for exploring.

Effectiveness on recognizing different types of predicates For frame-semantic parsing, extracting correct frame-evoking predicates is the first step that influences the later subtasks directly. Here we performed fine-grained analysis for the predicate identification, splitting the predicates into three categories, i.e., single-word predicates (Single), multi-word predicates (Multi), respectively. As shown in Figure 3, our joint model can achieve consistent improvements over the pipeline models for all kinds of predicates, indicating that the information from the frame and frame-specific roles are both beneficial for target identification. In addition, the multi-word predicates are more difficult than the single-word predicates, leading to significant decreases as a whole.

Performance by the role length Frame-special roles are the core structures that frame-semantic parsing intends to obtain. It is obvious that roles of different lengths would affect the performance, and longer roles would be much more difficult. Here we bucket the roles into seven categories and report the F1-score of our proposed methods on them. Figure 4 shows the results. We can find that the overall curve declines as the length increases, which is consistent with our intuition, and our graph-based end-to-end model is better than the pipeline methods of all lengths.

Performances of node, frame and edge Our graph-based model builds nodes, determines node attributes (frame), and builds edge sequentially, which is different from the standard pipelines based on target identification, frame classification and semantic role labeling. Thus, it is interesting to see the performance based on node building, frame classification[6] and edge building, respectively. Table 6 shows the results, where the joint model as well as four pipeline models are included. As shown, we can see that the full joint model is better than the partial joint models, and the full pipeline model gives the worst results.

Ignoring discontinuous predicates Although in both FN1.5 and FN1.7 datasets, discontinuous

[6]The frame classification is slightly different from that in the pipeline systems, which also includes the frame classification of the `PPRD` nodes.

Model	FN1.5	FN1.7
Semi-CRF	1.94 sent/s	1.72 sent/s
Ours-Joint	**15.72 sent/s**	**16.51 sent/s**

Table 8: Comparison on decoding speed (sentences per second) for the semantic role labeling subtask.

LU Lexicon	Text and Frames
✓	Up Tai Hang [**Road**]$_{Roadways}$ behind Causeway Bay is Aw Boon Haw (Tiger Balm) [**Gardens**]$_{Locale\_by\_use}$.
✗	Up Tai Hang [**Road**]$_{Roadways}$ [**behind**]$_{Locative\_relation}$ Causeway Bay is Aw Boon Haw (Tiger Balm) [**Gardens**]$_{Locale\_by\_use}$.
Ground Truth	Up Tai Hang [**Road**]$_{Roadways}$ behind Causeway [**Bay**]$_{Natural\_features}$ is Aw Boon Haw (Tiger Balm) [**Gardens**]$_{Locale\_by\_use}$.

Table 9: An example for frame suggestion out-the-scope-of the predefined LU lexicon, where the blue indicates the suggested frame outside the dictionary, ✓and ✗represent whether inference with the dictionary, respectively.

predicates are significantly smaller in amount than others, we keep it in this work for a more comprehensive study to demonstrate that our model can process them as well. Here we also add the results which ignore the discontinuous predicates (i.e., removing the predicate- predicate edges) to facilitate future studies. As shown in Table 7, our joint model performs better than the pipeline methods, which is consistent with the main results.

Comparison on decoding speed Table 8 compares the computational efficiency of the strong Semi-CRF baseline and our joint model for semantic role labeling task, which is also an essential measurement of proposed approach. Experimental results are all obtained by running models on a single 2080ti GPU. We could observe that our model can reach an almost ten times faster speed in comparison to Semi-CRF. Even though the Semi-CRF implementation[7] uses dynamic programming to optimize the time complexity, it still needs to iterate over segments of each sentence in the batch one by one, which might not take advantage of the GPU's parallel capabilities to accelerate the process. Nevertheless, our model as a whole adopts batch-based learning, which enables more efficient inference.

[7] https://github.com/swabhs/scaffolding.

Frame classification without dictionary Following Swayamdipta et al. (2017), we also adopt the Lexical Unit (LU) dictionary in our model empirically. However, according to Punyakanok et al. (2008), sometimes the dictionary might be quite limited. Therefore, we offer one eaxmple in Table 9 to illustrate the capability of our model for frames not in the dictionary. As shown, our model could predict the appropriate frame outside the dictionary as well and might additionally enrich the gold-standard annotations (i.e., the blue texts which do not appear in the Ground Truth).

5 Conclusion

In this paper, we proposed a novel graph-based model to address the end-to-end frame semantic parsing task. The full frame semantic parsing result of one sentence is organized as a graph, and then we suggest an end-to-end neural model for the graph building. Our model first encodes the input sentence for span representations with BERT, and then constructs the graph nodes and edges incrementally. To demonstrate the effectiveness of our method, we derived several pipeline methods and used them to conduct the experiments for comparisons. Experimental results showed that our graph-based model achieved significantly better performance than various pipeline methods. In addition, in order to compare our models with previous studies in the literature, we conducted experiments in the scenarios of the individual subtasks. The results showed that our proposed models are highly competitive.

Acknowledgements

We thank all reviewers for their hard work. This work is supported by grants from the National Key Research and Development Program of China (No. 2018YFC0832101) and the National Natural Science Foundation of China (No. 62176180).

References

Apoorv Agarwal, Sriramkumar Balasubramanian, Anup Kotalwar, Jiehan Zheng, and Owen Rambow. 2014. Frame semantic tree kernels for social network extraction from text. In *Proceedings of the EACL*, pages 211–219.

Collin 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 *Proceedings of the ACL-COLING*, pages 86–90.

Emanuele Bastianelli, Andrea Vanzo, and Oliver Lemon. 2020. Encoding syntactic constituency paths for frame-semantic parsing with graph convolutional networks. *CoRR*, abs/2011.13210.

Teresa Botschen, Iryna Gurevych, Jan-Christoph Klie, Hatem Mousselly-Sergieh, and Stefan Roth. 2018. Multimodal frame identification with multilingual evaluation. In *Proceedings of the NAACL-HLT*, pages 1481–1491.

Jiaxun Cai, Shexia He, Zuchao Li, and Hai Zhao. 2018. A full end-to-end semantic role labeler, syntactic-agnostic over syntactic-aware? In *Proceedings of the COLING*, pages 2753–2765.

Y. Chen, W. Y. Wang, and A. I. Rudnicky. 2013. Unsupervised induction and filling of semantic slots for spoken dialogue systems using frame-semantic parsing. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 120–125.

Bob Coyne, Alex Klapheke, Masoud Rouhizadeh, Richard Sproat, and Daniel Bauer. 2012. Annotation tools and knowledge representation for a text-to-scene system. In *Proceedings of the COLING 2012*, pages 679–694.

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

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 NAACL-HLT*.

Kalpit Dixit and Yaser Al-Onaizan. 2019. Span-level model for relation extraction. In *Proceedings of the ACL*, pages 5308–5314.

Timothy Dozat and Christopher D Manning. 2016. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734*.

Hao Fei, Yafeng Ren, and Donghong Ji. 2020. High-order refining for end-to-end Chinese semantic role labeling. In *Proceedings of the AACL*, pages 100–105.

Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. 2014. A discriminative graph-based parser for the Abstract Meaning Representation. In *Proceedings of the ACL*, pages 1426–1436.

Tsu-Jui Fu, Peng-Hsuan Li, and Wei-Yun Ma. 2019. GraphRel: Modeling text as relational graphs for joint entity and relation extraction. In *Proceedings of the ACL*, pages 1409–1418.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. Allennlp: A deep semantic natural language processing platform.

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

Silvana Hartmann, Ilia Kuznetsov, Teresa Martin, and Iryna Gurevych. 2017. Out-of-domain FrameNet semantic role labeling. In *Proceedings of the 15th EACL*, pages 471–482.

Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2018. Jointly predicting predicates and arguments in neural semantic role labeling. In *Proceedings of the ACL*, pages 364–369.

Karl Moritz Hermann, Dipanjan Das, Jason Weston, and Kuzman Ganchev. 2014. Semantic frame identification with distributed word representations. In *Proceedings of the ACL*, pages 1448–1458.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Tao Ji, Yuanbin Wu, and Man Lan. 2019. Graph-based dependency parsing with graph neural networks. In *Proceedings of the ACL*, pages 2475–2485.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

Meghana Kshirsagar, Sam Thomson, Nathan Schneider, Jaime Carbonell, Noah A. Smith, and Chris Dyer. 2015. Frame-semantic role labeling with heterogeneous annotations. In *Proceedings of the ACL-IJCNLP*, pages 218–224.

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 *Proceedings of the AAAI*, volume 33, pages 6730–6737.

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

Chunchuan Lyu and Ivan Titov. 2018. AMR parsing as graph prediction with latent alignment. In *Proceedings of the ACL*, pages 397–407.

Diego Marcheggiani and Ivan Titov. 2020. Graph convolutions over constituent trees for syntax-aware semantic role labeling. In *Proceedings of the EMNLP*, pages 3915–3928.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Hao Peng, Sam Thomson, Swabha Swayamdipta, and Noah A. Smith. 2018. Learning joint semantic parsers from disjoint data. In *Proceedings of the NAACL-HLT*, pages 1492–1502.

Vasin Punyakanok, Dan Roth, and Wen-tau Yih. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2):257–287.

Sunita Sarawagi and William W Cohen. 2005. Semi-markov conditional random fields for information extraction. In *Advances in Neural Information Processing Systems*, volume 17. MIT Press.

Dan Shen and Mirella Lapata. 2007. Using semantic roles to improve question answering. In *Proceedings of the EMNLP-CoNLL*, pages 12–21, Prague, Czech Republic. Association for Computational Linguistics.

Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Training very deep networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Changzhi Sun, Yeyun Gong, Yuanbin Wu, Ming Gong, Daxin Jiang, Man Lan, Shiliang Sun, and Nan Duan. 2019. Joint type inference on entities and relations via graph convolutional networks. In *Proceedings of the ACL*, pages 1361–1370.

Mihai Surdeanu, Sanda Harabagiu, John Williams, and Paul Aarseth. 2003. Using predicate-argument structures for information extraction. In *Proceedings of the ACL*, pages 8–15.

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*.

Swabha Swayamdipta, Sam Thomson, Kenton Lee, Luke Zettlemoyer, Chris Dyer, and Noah A. Smith. 2018. Syntactic scaffolds for semantic structures. In *Proceedings of the EMNLP*, pages 3772–3782.

Bishan Yang and Tom Mitchell. 2017. A joint sequential and relational model for frame-semantic parsing. In *Proceedings of the EMNLP*, pages 1247–1256. Association for Computational Linguistics.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019a. AMR parsing as sequence-to-graph transduction. In *Proceedings of the ACL*, pages 80–94.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019b. Broad-coverage semantic parsing as transduction. In *Proceedings of the EMNLP-IJCNLP*, pages 3786–3798, Hong Kong, China. Association for Computational Linguistics.

He Zhao, Longtao Huang, Rong Zhang, Quan Lu, and Hui Xue. 2020. SpanMlt: A span-based multi-task learning framework for pair-wise aspect and opinion terms extraction. In *Proceedings of the ACL*, pages 3239–3248.