A Span-based Target-aware Relation Model for Frame-Semantic Parsing

XUEFENG SU, School of computer and information technology, Shanxi University; School of Modern Logistics, Shanxi Vocational University of Engineering Science and Technology, China
RU LI*, School of computer and information technology, Shanxi University, China
XIAOLI LI, Institute for Infocomm Research, A*Star, Singapore
BAOBAO CHANG, The MOE Key Laboratory of Computational Linguistics, Peking University, China
ZHIWEI HU, School of computer and information technology, Shanxi University, China
XIAOQI HAN, School of computer and information technology, Shanxi University, China
ZHICHAO YAN, School of computer and information technology, Shanxi University, China

Frame-semantic Parsing (FSP) is a challenging and critical task in Natural Language Processing (NLP). Most of the existing studies decompose the FSP task into frame identification (FI) and frame semantic role labeling (FSRL) subtasks, and adopt a pipeline model architecture that clearly causes error propagation problem. On the other hand, recent jointly learning models aim to address the above problem and generally treat FSP as a span-level structured prediction task, which unfortunately leads to cascading error propagation problem between roles and less efficient solutions due to huge search space of roles. To address these problems, we reformulate the FSRL task into a *target-aware relation classification task*, and propose a novel and lightweight jointly learning framework that simultaneously processes three subtasks of FSP, including frame identification, argument identification and role classification. The novel task formulation and jointly learning with interaction mechanisms among subtasks can help improve the overall system performance, and reduce the search space and time complexity, compared with existing methods. Extensive experimental results demonstrate that our proposed model significantly outperforms ten state-of-the-art models in terms of F1 score across two benchmark datasets.

CCS Concepts: • **Computing methodologies**; • **Artificial intelligence**; • **Natural language processing**;

Additional Key Words and Phrases: FrameNet, Frame-semantic parsing, Frame identification, Relation model

1 INTRODUCTION

Frame-semantic parsing (FSP) is a task of identifying the semantic frames evoked in text along with their semantic roles, formalized in the FrameNet project [2, 8]. In particular, a frame represents an event's *semantic scenario*, and possesses frame elements or *semantic roles* that participate in the event [16], which is grounded on the theory of Frame Semantics [9]. Frame-semantics has shown to be useful in event recognition [19], machine reading comprehension [13, 14], relation extraction [35], and text generation [12], among others.

An example sentence *my brother is locked up in the laundry room* and its frame-semantic annotations are shown in Figure 1. In the example, there are 3 frames, namely, *Immobilization*, *Locative_relation* and

[*] Corresponding author.

Authors' addresses: Xuefeng Su, suexf@163.com, School of computer and information technology, Shanxi University; School of Modern Logistics, Shanxi Vocational University of Engineering Science and Technology, China; Ru Li, School of computer and information technology, Shanxi University, Taiyuan, China, liru@sxu.edu.cn; Xiaoli Li, Institute for Infocomm Research, A*Star, Singapore, xlli@ntu.edu.sg; Baobao Chang, The MOE Key Laboratory of Computational Linguistics, Peking University, China; Zhiwei Hu, School of computer and information technology, Shanxi University, China; Xiaoqi Han, School of computer and information technology, Shanxi University, China; Zhichao Yan, School of computer and information technology, Shanxi University, China.

Building_subparts, evoked by 3 corresponding targets (target words include *locked up*, *in*, and *laundry room*) respectively. In addition, 5 frame-specific semantic roles (e.g. *Patient*, *Place*, *Figure*, etc.) are filled by 5 target-related *arguments* respectively in the sentence, where an argument is a continuous and meaningful text span in the given sentence. For example, "*My brother*" is an *argument* of target *locked up*, and it acts as *Patient* role of frame *Immobilization* in this sentence. In other words, the semantic role of *my brother* is a *patient* as he is locked up or in the semantic scenario of immobilization. Compared with PropBank-style semantic role labeling (SRL), FSP has to handle the thousands [1] of frame-specific semantic roles, while PropBank SRL only uses a small set of 26 syntactically motivated roles. Nevertheless, the more fine-grained semantic roles result in much more rich semantic interpretation of text which is useful and critical for many downstream tasks, although it makes the FSP task more challenging.

Early work adopts pipeline strategy that divides the FSP task into frame identification (FI) subtask and frame-semantic role labeling (FSRL) subtask [4, 17, 27, 28, 33]. In particular, FI subtask aims to find the exact frame evoked by a target word in a given sentence. Through FI step, the thousands of semantic roles can be reduced to a small set of up to 30 frame-specific roles, which makes the subsequent FSRL subtask easier. However, the pipeline strategy usually causes error propagation problem, as the overall performance of FSP is more sensitive to the performance of first FI subtask.

Recently, jointly learning models have drawn much attention for the FSP task. Their models adopt structured prediction paradigm, in which *semantic roles are considered interdependent from each other*, while roles are dependent on the frame. An diagram of structured prediction paradigm is shown in Figure 2(a). The main difference between these models is in how to capture the structural information of frame-semantic. For instance, Peng et al.[21] propose a joint scoring model that learns a joint scoring function for FI and FRSL subtask, and use a linear programming algorithm to search the global optimal solution under the structural constraints of frame-semantic in its inference phase. The maximal searching space reaches to $O(|F||n(n+1)/2||R|)$, where F and R are frames set and roles set in FrameNet knowledge base, and n is the length of the sentence. Chen et al.[3] propose an encoder-decoder model which processes all the subtasks jointly by optimizing them together, and treats FSRL as a *role sequence generation process*. The model uses LSTM to explicitly capture *the dependency between roles* during training phase and predicts the roles subsequently in the inference phase. Overall, existing joint learning models have achieved better performance than pipeline models. However, they have two weaknesses:

(1) For the joint scoring model [21], it does not explicitly capture the structural information in its training phase, and thus the huge searching space affects running efficiency and accuracy in its inference phase.

(2) For the encoder-decoder jointly learning model [3], the way that it predicts role sequence will still cause cascading error propagation problem. As shown in Figure 2(a), if the predicted role of first argument "*my brother*" is wrong, the subsequent prediction will also likely be wrong due to its role dependency assumption.

To address the above weaknesses, we propose a new span-based **t**arget-**a**ware **r**elation classification model for FSP (**TaRFSP**), which jointly processes the FI and FRSL subtasks. In this model, (1) we relax the role dependent assumption of structured prediction paradigm that previous FSP models adopt. In other words, we only assume roles are dependent on frame, while roles are independent from each other. For example in Figure 2(a), the roles between *Patient* and *Null*, and *Null* and *Place* do not have strong dependency at all.

[1] There are 1170 and 1285 roles in FrameNet 1.5 dataset and FrameNet 1.7 dataset respectively.

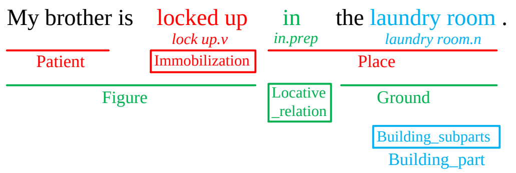
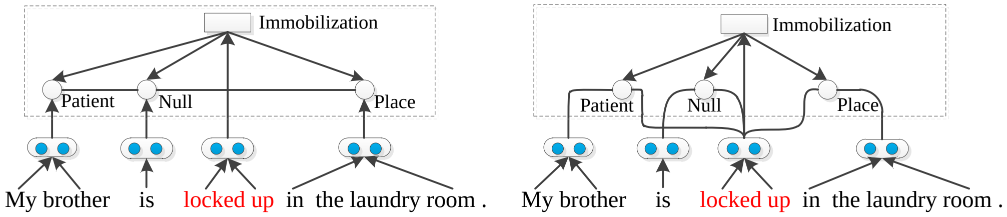


Fig. 1. An example sentence with 3 groups of color-coded frame-semantic annotations below. The three targets are highlighted in the sentence, and their lexical units (original words and corresponding part of speech) are shown italicized below. Frames evoked by targets are shown under the targets in colored blocks, and the corresponding frame semantic roles are shown under horizontal lines alongside the frame. Note that semantic role *Building_part* is placed directly under its frame, which is different from other role annotations, since the argument of this role and its target are totally overlapped.


Immobilization
Patient
Null
Place
My brother is locked up in the laundry room .
Immobilization
Patient
Null
Place
My brother is locked up in the laundry room .

(a) Diagram of structured prediction paradigm for FSP

(b) Diagram of relation classification paradigm for FSP

Fig. 2. The comparison between two schematic diagrams for FSP. The target are highlighted in red color. Frame (Immobilization) and frame-specific roles (e.g. Patient) are shown next to block and circles respectively. In diagram (a), the role of each argument is not only dependent on the current frame but also its neighbor roles. In diagram (b), the role of each argument is only dependent on the current frame, and each role indicates a kind of relation between the argument and its target (the semantic role of *my brother* is a *Patient* as he is *locked up* and in the semantic scenario of *Immobilization*). *Null* denotes the word or span is not an argument of current target. For instance, "*is*" is not an argument of target *locked up*.

(2) We model role classification of each argument as independent target-argument relation classification conditioned on the frame evoked by the target. For instance, in Figure 2(b), target *locked up* evokes the frame *Immobilization* which defines a semantic scenario with two semantic roles *Patient* and *Place*. Target-argument relation between target *locked up* and argument *my brother* helps us to infer *my brother* cannot move in the semantic scenario of *immobilization* due to *locked up* and thus should serve as the semantic role *Patient*. (3) We decompose FSRL into two components: 1) argument identification and 2) role classification. Argument identification is used to identify and filter the impossible arguments, while role classification is used to predict the semantic roles of the most possible arguments.

Compared with previous jointly learning model based on structured prediction paradigm, as shown Figure 2(b), our method predicts the role of each argument independently without considering the role of argument prior to current argument (e.g. we infer *in the laundry room* as a role *Place*, which do not care about arguments *my brother* and *is* and their corresponding roles), which can bring two benefits: overcoming the cascading error propagation problem between roles and enhancing the running efficiency through parallel running of each pair of target-argument classification. As shown Figure 2(b), the role prediction of three possible arguments is processed as three independent target-argument relation classification problem, and each relation is only dependent on the current frame. Furthermore, our method explicitly captures the structural information in training phase, so the frame-roles dependence relations are learned in the trained model. In inference phase the model can identify frames, arguments and argument roles in turn without considering the structural constraints, which makes the maximal searching space be reduced to $O(|F| + 2|n(n+1)/2| + k|R_f|)$, where k is the number of predicted arguments and R_f is the roles set of current predicted frame. In summary, our main contributions can be summarized as follows:

(1) We present a method of converting FSP task, generally regarded as a structured prediction task, into a *target-aware relation classification task*. To the best of our knowledge, this is the first work to tackle FSP based on semantic relation classification. Correspondingly, we develop a new non-structured prediction paradigm for FSP which can significantly mitigate the error propagation problem and reduce the complexity of FSP task comparing with the existing work, without using any optimization search algorithm.

(2) We propose a novel and lightweight model for FSP that jointly processes three subtasks of FSP, including frame identification, argument identification and role-argument classification. Moreover, we also design a group of lightweight interaction mechanisms among subtasks and one post-processing procedure for handling the structure constraint.

(3) We have performed extensive experiments and our experimental results demonstrate that our proposed model outperforms ten state-of-the-art models in terms of $F1$ score across two benchmark datasets.

2 RELATED WORK

FSP task was first proposed in 2002 [11] and has drawn great attention since the SemEval 2007 shared task [1] was released. Early studies adopted a pipeline strategy, which involves FI subtask and FRSL subtask. In general, FI is regarded as a multi-class classification task, while RFSL is regarded as a structured prediction task. Before the popularity of representation learning, the researchers mainly used discrete syntax features and statistical learning methods, such as SVM, CRF, etc., to construct models for FI and FRSL [4, 11]. Das et al. adopted two separate conditional log-linear models to calculate the prediction scores for targets and arguments, and subsequently used linear programming to search the optimal role sequences [4].

With the recent development on deep neural networks and representation learning, many researchers converted the discrete syntax features into distributed representations, and used neural networks to construct FI and FRSL models. For instance, Hermann et al. [17] proposed a FI model using word embedding to represent the context and WSABIE algorithm [31] to train the model. Subsequently, as for FSRL, they utilized the local log-probability to calculate the scores for each argument and performed global inference by applying an integer linear program (ILP) optimization method, subject to hard structural constraints. Michael Roth and Mirella Lapata [23] presented a semantic role labeling system that took into account discourse context and used Glove [22] embedding to represent the features. Täckström et al. [29] proposed a pipeline model for FI and FRSL using distributed context features, and it was the first to use a globally normalized probabilistic model with structural constraints for FRSL. FitzGerald et al. [10] proposed a graphical model for FSRL based on neural networks, which jointly models the assignment of arguments to their semantic roles, subject to linguistic constraints. Yang and Mitchell [33] first presented two separate models for FRSL, i.e., a sequence labeling model (SEQ) based on CRF, a relation model (REL) based on a LSTM encoder and MLP potential function. Subsequently, they proposed an integrated model that incorporated SEQ model into REL model by a unified training objective. They also proposed a FI model using a linear potential function. Finally, they solved the joint inference for assigning frames and roles to all predicates and their arguments by the AD^3 algorithm. Finally, Swayamdipta el al. [27] presented a

FSP model with softmax-margin segmental RNNs, a variant of a semi-Markov CRF, which allows scoring functions that directly model an entire variable-length segment, and used dynamic programming algorithm to predict the labels. All of the above studies adopt pipeline model structure and used optimization algorithm, such as liner programming, dynamic programming, AD^3 etc., to solve the optimal solution for FSP in their inference phase.

Recently, jointly learning models have drawn more attention. Peng et al. [21] proposed a jointly structured prediction model which uses a structured hinge objective for training and a linear programming procedure for inference. Chen et al. [3] presented a novel architecture based on multi-decoder strategy to handle three subtasks of FSP. Overall, jointly learning models have proved more effective than the pipeline models, since these two models achieve the current state-of-the-art performance.

In this study, we also adopt the jointly learning strategy and decomposed the FSP task into three subtasks: frame identification, argument identification and role classification, which is similar to Chen et al.'s work [3]. However, they adopt *structured prediction paradigm* based on its role interdependent assumption, while we proposed a novel *relation classification paradigm.* As discussed in Introduction section, the interdependent assumption may not be valid in practice. In addition, modeling a longer dependency chain can cause cascading error propagation problem. Our work focuses on capturing the dependency between target-argument pair and each role independently, which can avoid their problems and generate more accurate results.

Different from the graph-based neural model for end-to-end FSP recently proposed by Lin et al.[18], the model includes an additional target identification subtask since the target word is assumed to be unknown, while our work and the studies mentioned above focus on FSP task when the target word is given.

3 TASK FORMULATION

Frame-semantic parsing (FSP) consists of two subtasks, namely, FI and FSRL subtask. Specifically, for a given sentence $\mathbf{x}$ with m words $x_0, x_2, ..., x_{m-1}$ and a set T of targets (target words or phrases) in the sentence, FI aims to identify each **frame** f evoked by each target word $t, t \in T$, and FSRL aims to identify a set of non-overlapping **arguments** for each frame f, where each argument $a = (i, j, r)$ has a start index i, end index j ($0 \leq i \leq j \leq m-1$) and role label r. For simplicity, we call the word (or phrase) that can evoke a frame (a semantic scenario) **target** in this paper. Each target is associated with a **lexical unit** (or **LU**) in FrameNet, and a LU consists of the lemma and part-of-speech tag of the target. FrameNet also provides mapping from a LU l to a set of frames it evokes, denoted as $\mathcal{F}_l$, and mapping from a frame f to a set of semantic roles (or frame elements) $\mathcal{R}_f$. Roles could be core roles and non-core roles.

For example, in Figure 1, *lock up.v* is the corresponding LU of target *locked up* that evokes frame *Immobilization.* Its frame semantic roles *Patient* and *Place* are filled by the argument "*My brother*" and "*in the laundry room*" respectively. *Immobilization* has 10 roles in total (including *Agent*, *Anchor*, *Degree*, etc.), although some of them do not occur in this sentence. In this frame, *Agent*, *Patient* and *Place* are its core roles, and other roles (e.g. *Anchor*, *Degree*, *manner*, etc.) are its non-core roles.

In this work, we parse each target (e.g. *locked up*) independently following the previous studies [4, 17, 27, 28]. Specifically, Given the sentence $\mathbf{x}$ and a target t, then the FSP model can be formalized as:

$$f = argmax\, \mathbf{P}_\theta(f \,|\, \mathbf{x}, t), \tag{1}$$

$$Y = argmax\, \mathbf{P}_\Phi(Y \,|\, \mathbf{x}, t, f) \tag{2}$$

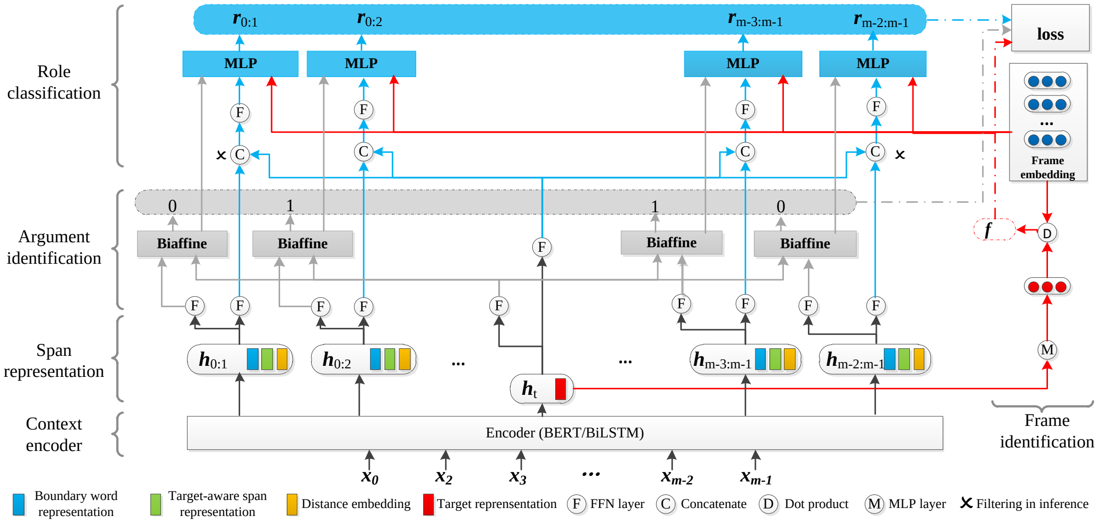


Fig. 3. Overall architecture of our model. Gray arrows, blue arrows and red arrows denote the data flows of arguments identification module, role classification module and frame identification module respectively.

Here, $f \in \mathcal{F}_l$ and $Y = \{(i, j, r)_k\}_{k=1}^{|Y|}$ are predicted frame and possible arguments respectively, where $r \in \mathcal{R}_f \cup \{Null\}$ is the role of each argument, and $Null$ is a special class that denotes the possible argument is not an true argument of the given target. Each possible argument is usually a text span, consisting of continuous words in a sentence, so we also call them argument span in the following. For instance, shown in Figure 1, "*My*", "*My brother*" and "*My brother is*" are three possible arguments of target *locked up* in the sentence.

4 THE PROPOSED METHOD

4.1 Model Architecture

In this paper, we propose a new span-based **t**arget-**a**ware **r**elation classification model for FSP (**TaRFSP**), which jointly processes three subtasks of FSP, i.e., frame identification, argument identification and role classification. As shown in Figure 3, the proposed framework mainly consists of five modules: (1) context encoder, (2) span representation, (3) frame identification, (4) argument identification and (5) role classification.

In particular, context encoder module is responsible for modeling the context representation of the input sentence, and span representation module is used to model the target-aware span representations for all the possible arguments. These two modules are shared by the other three modules. Frame identification module is responsible for identifying the frame evoked by the target and learning the frame embedding simultaneously. Argument identification and role classification module are used to identify arguments and predict the role of each argument respectively. Considering the running efficiency, we design two lightweight and explicit interaction mechanisms. One is frame embedding sharing between frame identification module and role classification module. The other is signal passing from argument identification module to role classification module. Furthermore, These three modules interact implicitly by sharing the parameters of span representation module and context encoder module.

4.2 Context Encoder

Given a sentence $\mathbf{x} = x_0, x_2, ..., x_{m-1}$, we obtain the hidden representation $\mathbf{h}_i$ of each token x_i via a deep contextualized encoder.

$$\mathbf{h}_0, \mathbf{h}_1, ..., \mathbf{h}_{m-1} = Encoder(x_0, x_1, ..., x_{m-1}) \tag{3}$$

where the dimension of $\mathbf{h}_i$ $(0 \leq i \leq m-1)$ is d_h. In this work, we adopt two alternative model architectures for the encoder, i.e. **BiLSTM**-based and **BERT**-based [5]. For the BiLSTM-based model, we initialize the token embedding with Glove [**?**]. For the sake of simplicity and running efficiency, we do not use any other information of tokens, such as POS embedding, lemma embedding and char embedding, which is different from the previous work [3, 21]. Meanwhile, we believe these information has little effect on improving the performance of the model. For the BERT-based model, we conduct finetuning on original BERT model, and use the outputs of last layer as the contextual representation for each token in the sentence.

4.3 Span Representation

In this work, we regard target t as a special span and adopt different representation method to represent it according to its usage. When it acts as a candidate argument span, we use Function (7) to represent it. When it is used to identify the frame, its representation can be calculated by the following function:

$$\mathbf{h}_t = (\mathbf{h}_b + \mathbf{h}_e)/2. \tag{4}$$

Note that, since BERT adopts WordPiece tetanization, we simply use the representations of boundary WordPiece instead of the representations of boundary word.

For argument span representation, we hope the representation can not only represent the semantic of the span but also capture the semantic relations between target and argument. Given the sentence $\mathbf{x} = x_0, x_2, ..., x_{m-1}$, we use $x_{i:j}$ denote a continuous words span from index i to index j $(0 \leq i \leq j \leq m-1)$. In the FSP task, The model aims to distinguish not only which span is *true argument*, but also *the role of the argument*. So, we have to choose the most distinguishing features for argument identification and role classification to represent each span. We believe the boundaries of a span contain rich *syntactic and semantic information* which are beneficial for argument identification and argument role classification. For example, as shown in Figure 1, the boundaries of span "*in the laundry room*" are "*in*" and "*room*". Here "*in*" signifies the beginning of a prepositional phrase, while "*room*" is a noun which acts as a prepositional object of preposition "*in*" and indicates the end of the phrase. In addition, two boundary words together also express the meaning of its rule as *place*. For the information of middle words in a span, we use target-aware attention to weight the word representations in the span. The formulation of attention is shown as follows:

$$\begin{aligned} \mathbf{a}_k &= \frac{exp(h_k^T h_t)}{\sum_{l=i}^{l=j} exp(h_l^T h_t)} \\ \mathbf{h}_{att} &= \sum_{k=i}^{k=j} a_k h_k \end{aligned} \tag{5}$$

where $\mathbf{a}_k$ is the attention value of k-th word in the span from index i to index j in a sentence, and $\mathbf{h}_{att}$ is the target-aware representation of the span.

Furthermore, the position relation between target and its argument span is also important to distinguish their roles. In particular, the core-roles are usually closer to the target than noncore-roles in a sentence. For instance, the argument "*My brother*" and "*in the laundry room*" act as core-role: *Patient* and *Place*, and they are all near to the target "*locked up*". In addition, the *patient* is usually occured after the verb target in an active sentence, while it is usually occured before the target in an passive sentence. As shown in Figure 1, "*My brother*" occurs before the target "*locked up*", acting as *patient* in the passive sentence. Given a span $x_{i:j}$, we use the relative distance to model the position relation between target and the span, and the relative distance can be calculated using the following function:

$$D(t \rightarrow x_{i,j}) = \begin{cases} b - j, & j < b \\ e - i, & i > e \\ 0, & i = b, j = e \end{cases} \tag{6}$$

where t is the target in sentence $\mathbf{x}$, and b and e are indices of t in the sentence ($0 \leq b \leq e \leq m - 1$). Here, we regard target as a special span. This function is only used to measure the position from the target to its candidate argument spans. The spans partially overlapping with the target are filtered in our method which will be discussed in the following part of the paper.

In addition, the length of a span is one of the widely used features in span-based models [7, 20, 21, 36]. Thus, in the work, our span representations are computed based on three components, i.e. (1) boundary word representations, (2)target-aware span representation, (3) the relative distance from the span to the target, and (4) the length of span. Particularly, the representation of span $x_{i:j}$ can be denoted as:

$$\mathbf{h}_{i:j} = [\mathbf{h}_i; \mathbf{h}_j; \mathbf{h}_{att}; E(D(t \rightarrow x_{i,j})); E(|x_{i,j}|)] \tag{7}$$

where $\boldsymbol{E}(.)$ denotes an embedding function which converts the discrete value into distributed representation, and [.;.] denotes vector concatenation.

4.4 Frame Identification

Frame identification is usually regarded as a classification task in the previous work [4, 15, 17], which aims to identify the frame evoked by a given target in a sentence. In this work, however, the frame identification module is not only responsible for frame classification, but also responsible for learning the frame representation which is used to interact with role classification module. Thus, inspired by KGFI model[26], we simply adopt a bi-encoder structure. One encoder is used to represent target, while the other is used to represent frames. The bi-encoder maps the target and frames into the same space which facilitates predicting the frame by calculating the similarity, along with obtaining the frame representation. As shown in Figure 3, the frame embedding block denotes the representation of all the frames and the red vector denotes the target representation. The detailed methods of obtaining the representations of frames and target are described in the following.

Given the frame set $\mathcal{F} = \{f_1, f_2, ..., f_{|\mathcal{F}|}\}$, the frame encoder maps the discrete frame labels into distributed representation. The mapping function is defined as:

$$\mathbf{M}^{(\mathcal{F})} = [E(f_1); E(f_2); ...; E(f_{|\mathcal{F}|})]^T \mathbf{W}^{(\mathcal{F})} \tag{8}$$

where $E(.)$ denotes an embedding function, $\mathbf{W}^{(\mathcal{F})}$ is a learnable matrix, and $\mathbf{M}^{(\mathcal{F})}$ is the frame matrix in which each row represents a frame embedding. The dimension of matrix $\mathbf{M}^{(\mathcal{F})}$ is $|\mathcal{F}| \times d_f$.

For the target t, we first use a deep contextualized encoder, such as BERT or BiLSTM, to obtain its hidden representation $\mathbf{h}_t$, and subsequently use the target encoder maps $\mathbf{h}_t$ into the same space as frames. The mapping function is defined as:

$$\mathbf{g}_f(t) = \mathbf{W}_f \mathbf{h}_t + \mathbf{b}_f \tag{9}$$

where $\mathbf{W}_f$ and $\mathbf{b}_f$ are learnable parameters, and the demension of $\mathbf{W}_f$ is $d_f \times d_h$.

After obtaining the representations of target and frames, we adopt dot product to calculate their similarity scores, and then use a *softmax* function to normalize the scores into probability distribution:

$$P(f \,|\, \mathbf{x}, t) = softmax(\mathbf{M}^{(\mathcal{F})}.\mathbf{g}_f(t)). \tag{10}$$

In the inference phase, we choose the best score from the possible frame subset, and predict the frame with the highest score:

$$\hat{f} = argmax_{f \in \mathcal{F}_l} P(f \,|\, \mathbf{x}, t) \tag{11}$$

where l is the LU of target t, and $\mathcal{F}_l$ is the frame subset which can be evoked by the target t. The mapping from LU to frames are defined in FrameNet.

4.5 Argument Identification

Argument identification module identifies the true arguments of the target from all the possible continuous word spans of a given sentence. We divide argument identification into two steps: candidate argument generation and argument classification.

Suppose a sentence $\mathbf{x}$ with $|\mathbf{x}|$ words, the amount of possible argument spans is $|\mathbf{x}|(|\mathbf{x}| + 1)/2$, which could be very big given a long sentence, especially compared with the ground truth, i.e., an average of about up to 3 arguments for a target in a sentence[2]. To address this problem, we adopt three pruning strategies. 1) We set the maximum length of a span following the previous work [3, 21], and a argument span should not exceed the maximum length. 2) We set up a window with fixed size centered on the target word and generate the candicate arguments within the window, as most of the arguments are locally located around the target. 3) An argument span can't partially overlap with the target, while an argument may sometimes totally overlap with the target. We call the span that meets these three criteria candidate argument. As shown in Figure 1, the span "*laundry room*" is not only a target that evokes frame *Building_subparts*, but also a argument that acts as role *Building_part* of frame *Building_subparts*. In this case, the argument and the target are totally overlapped, so the span totally overlapping with the target could be a candidate argument. In addition, the "*is locked up in*" is partially overlapped with target *locked up*. So, it is not a candidate argument of target *locked up* according to our criteria.

The detailed generating processes are illustrated in Algorithm 1. Specifically, first, the algorithm obtains the start and end index of target in the sentence x, as illustrated in line 5. Second, set up the window boundary according the target position, as illustrate in line 6-7. Third, as illustrated in line 8-13, the algorithm enumerates all the possible spans of sentence constrained on maximum length max_len, and filters the spans that partially overlapped with target. Meanwhile, the algorithm obtains the indexes of boundary

[2] The statistical result is from FrameNet 1.5 and FrameNet 1.7.

Algorithm 1 Generating the candidate arguments

```
1:  Input: sentence $\mathbf{x}$ and a target $t$ in $\mathbf{x}$,
2:          the max length of a span $max\_len$.
3:          the fixed window size $win\_size$.
4:  $\mathcal{A} \leftarrow \emptyset$
5:  $b, e \leftarrow getTargetIndex(t, \mathbf{x})$
6:  $win\_l$=max($b - win\_size/2$,0)
7:  $win\_r$=min($e + win\_size/2$,$|\mathbf{x}|$))
8:  for $i$ in range($win\_l$,$win\_r$):
9:      for $j$ in range($i + 1$, min($win\_r$,i+$max\_len$)):
10:         if $overlap((i, j), (b, e))$ ==False:
11:             $distance = D(t, x_{i:j})$
12:             $width = j - i$
13:             $\mathcal{A}.add((i, j, distance, width))$
14: $\mathcal{A} = \mathcal{A} \cup \{(b, e, 0, b - e)\}$
15: return $\mathcal{A}$
```

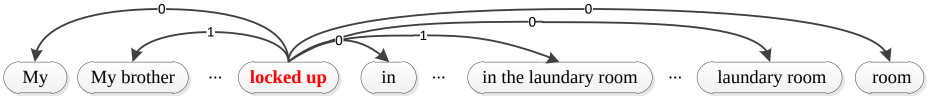


Fig. 4. An example for argument identification based on span-level dependency relations.

words and calculates the distance and width of each span, and subsequent adds each span into candidate argument set $\mathcal{A}$. Finally, the target as a special candidate argument is added into $\mathcal{A}$, as shown in line 14.

Inspired by word-level dependency parsing [25, 34], we formulate argument classification as a span-level dependency parsing. In particularly, we only model whether there is a dependency between the target and each candidate argument, regardless of the type of dependency. As shown in Figure 4, it is an example for argument identification based on span-level dependency relations, where "*locked up*" is the target in sentence "*My brother is locked up in the laundry room*". So, here, argument classification is a binary classification task. We use biaffine network [6, 30] to model the span-level dependencies between the target and candidate arguments.

Let $\mathcal{A} = \{a_1, ..., a_{|\mathcal{A}|}\}$ denote the candidate arguments of target t, the relation scores between target t and k-th candidate argument a_k can be calculated by:

$$\begin{aligned}
&\mathbf{h}_a(a_k) = FFN_a(\mathbf{h}_{a_k(i):a_k(j)}) \\
&\mathbf{h}_s(t) = FFN_s(\mathbf{h}_t) \\
&\mathbf{g}_a(a_k) = \mathbf{h}_a(a_k)^T \mathbf{U}_a \mathbf{h}_s(t) + \mathbf{V}_a^T [\mathbf{h}_s(t); \mathbf{h}_a(a_k)] + \mathbf{b}_a
\end{aligned} \tag{12}$$

where FFN_a and FFN_s are two separate one-layer feedforward neural network (FFN) with *relu* activation function, and $a_k(i)$ and $a_k(j)$ denote the start and end indices of argument span a_k in the sentence. $\mathbf{U}_a$ is a $d_a \times 2 \times d_a$ tensor, $\mathbf{V}_a$ is a $2d_a \times 2$ matrix, and b_a is the bias vector. $\mathbf{g}_a(a_k)$ is a 2-dimension vector. Finally, we apply a softmax function to normalize the scores into probability distribution:

$$P(y(a_k)\,|\,\mathbf{x}, t, a_k) = softmax(\mathbf{g}_a(a_k)) \tag{13}$$

where $y(a_k) \in \{0, 1\}$ denotes wether the candidate argument a_k is a false or true argument of target t. In the inference phase, we identify a_k as an argument of target t when $P(y(a_k) = 1\,|\,\mathbf{x}, t, a_k) > P(y(a_k) = 0\,|\,\mathbf{x}, t, a_k)$ is true.

4.6 Role Classification

Role classification module is responsible for predicting the roles of a set of arguments of target t in a sentence. We formulate role classification as multi-class relation classification task, where each class is a specific role that denotes a semantic relation between the target t and one of the arguments. Meanwhile, we observe both *arguments of a given target* and *their corresponding roles* are independent from each other. For example, as shown in Figure 1, the appearance of argument "*My brother*" will not necessarily lead to the appearance of argument "*in the laundry room*". Similarly, in a frame (semantic scenario) *Immobilization*, the appearance of role *Patient* will not necessarily lead to the appearance of role *Place*. Thus, we independently model the relation of each (target, argument) pair in a sentence, which is significantly different from existing approach. Specifically, given a (target, argument) pair (t, $a_k \in \mathcal{A}$), we calculate the relation score of the pair using following functions:

$$\begin{aligned}
\mathbf{h}_r(a_k) &= FFN_r(\mathbf{h}_{a_k(i):a_k(j)}) \\
\mathbf{h}_o(t) &= FFN_o(\mathbf{h}_t) \\
\mathbf{g}_o(a_k) &= \mathbf{W}_o^T[\mathbf{h}_o(t); \mathbf{h}_r(a_k)] + \mathbf{b}_o
\end{aligned} \tag{14}$$

FFN_a and FFN_s are two separate one-layer FFN with *relu* activation function. $\mathbf{h}_{a_k(i):a_k(j)}$ and $\mathbf{h}_t$ denote the hidden representation of argument span a_k and target t respectively. $\mathbf{W}_o$ is a $2d_o \times (|\mathcal{R}|)$ learnable matrix, and b_o is the bias vector. Different from argument identification module, we use linear network instead of biaffine network to construct the score function due to the following consideration. First, we introduce the diversity between the two modules which can focus on different aspects of the relations between the target and its arguments respectively. Second, the complexity of biaffine network is higher than that of linear network, especially in our our role classification subtask, as there are many semantic roles ($|\mathcal{R}| \gg 2$) that need to be taken into account.

In addition, since the roles are constrained on frames, we design a explicitly interactive mechanism that captures the semantic constraints using a simple linear network, which is beneficial for predicting frame-specific roles. Meanwhile, to explicitly model the interaction between argument identification module and role classification module, we simply pass the argument identification signal to role classification module, which has been empirically proved useful for both modules. Thus, We directly replace the function $\mathbf{g}_o(a_k)$ with following two functions:

$$\begin{aligned}
\mathbf{h}_c(a_k) &= FFN_c([\mathbf{h}_o(t); \mathbf{h}_r(a_k)]) \\
\mathbf{g}_o(a_k) &= \mathbf{W}_r^T[\mathbf{M}^{(\mathcal{F})}[\hat{f}]; \mathbf{h}_c(a_k); \mathbf{g}_a(a_k)] + \mathbf{b}_r
\end{aligned} \tag{15}$$

where $\mathbf{M}^{(\mathcal{F})}$ is the frame embedding matrix learned by frame identification module, $\hat{f}$ is the predicted frame of target t by frame identification module, and $\mathbf{M}^{(\mathcal{F})}[\hat{f}]$ is the embedding of frame $\hat{f}$. $\mathbf{g}_a(a_k)$ is the argument

Algorithm 2 Inference with Non-overlap Constraint

```
Input: sentence $\mathbf{x}$ and a target $t$ in $\mathbf{x}$,
        the trained model $P(.)$,
        candidate arguments $\mathcal{A} = \{a_1, ..., a_{|\mathcal{A}|}\}$.
$cur \leftarrow 0$
$\mathcal{C} = \emptyset$
$\hat{f} \leftarrow argmax_{f \in \mathcal{F}_l} P(f \,|\, \mathbf{x}, t)$
$\mathcal{A}^+ \leftarrow \{(a_k, score) \,|\, a_k \in \mathcal{A}, y(a_k) = 1$
        $and\ score = P(y(a_k) = 1 \,|\, \mathbf{x}, t, a_k)\}$
$\mathcal{A}^+ \leftarrow$descending_sort_by_score($\mathcal{A}^+$)
while $|\mathcal{A}^+| - cur > 2$ :
    $flag \leftarrow cur$
    for $idx$ in range($cur$, $|\mathcal{A}^+|$):
        if $cur! = idx$
          and is_overlap($\mathcal{A}^+[cur], \mathcal{A}^+[idx]$):
            $flag = idx$
    $cur = cur + 1$
    if $flag > cur$:
        $\mathcal{A}^+$.remove_by_index($flag$)
for $a_k$ in $\mathcal{A}^+$:
    $\mathcal{C}.add\{(a_k, label) \,|\, label = \hat{r}(a_k)\ and$
    $\hat{r}(a_k) = argmax_{r(a_k) \in \mathcal{R}_{\hat{f}}} P(r(a_k) \,|\, \mathbf{x}, t, a_k, \hat{f})\}$
return $\hat{f}$, $\mathcal{C}$
```

score from argument identification module. $\mathbf{W}_r$ is a $(d_o + d_f + 2) \times (|\mathcal{R}|)$ learnable matrix, and b_r is the bias vector.

After obtaining the relation scores, we apply a softmax function to normalize the scores into probability distribution:

$$P(r(a_k) \,|\, \mathbf{x}, t, a_k, \hat{f}) = softmax(\mathbf{g}_o(a_k)). \tag{16}$$

In our inference phase, since the golden frame f is unknown, we can't predict the roles from role subset $\mathcal{R}_f$. However, if we directly use the predicted frame $\hat{f}$ to reduce the candidate roles to $\mathcal{R}_{\hat{f}}$, which may cause error propagation. Since we consider the model can automatically focus on the most frame-specific roles through interaction with frame and these roles can obtain higher scores, we have to predict the roles from the full role set $\mathcal{R}$:

$$\hat{r}(a_k) = argmax_{r(a_k) \in \mathcal{R}} P(r(a_k) \,|\, \mathbf{x}, t, a_k, \hat{f}). \tag{17}$$

In our training phase, we will calculate the relation scores for all of the (target, candidate argument) pairs, just like argument identification module. However, in prediction phase, we only need to calculate the relation scores of (target, argument) pairs that are predicted as arguments by argument identification module. This two-steps roles labeling strategy is beneficial for both improving prediction accuracy and running efficiency.

4.7 Training and Inference

During our training phase, we utilize cross-entropy loss to maximize the probability of the golden frame type, argument type and role class:

$$\begin{aligned}
\mathscr{L}_f &= log(P(f \,|\, \mathbf{x}, t)) \\
\mathscr{L}_a &= \sum_{k=1}^{k=|\mathcal{A}|} log(P(y(a_k) \,|\, \mathbf{x}, t, a_k)) \\
\mathscr{L}_r &= \sum_{k=1}^{k=|\mathcal{A}|} log(P(r(a_k) \,|\, \mathbf{x}, t, a_k, \hat{f}))
\end{aligned} \tag{18}$$

The total loss is defined as:

$$\mathscr{L} = \mathscr{L}_f + \mathscr{L}_a + \mathscr{L}_r \tag{19}$$

where α, β and γ are hyper-parameters that adjust the weights of each loss, and each function is the loss for one training sample. Finally, we jointly optimize the losses of three subtasks over all the training data.

In our inference phase, the argument identification module first identifies the arguments of the target from all the candidate arguments, and then the role classification module predicts the roles for the identified arguments instead all the candidate arguments, which can reduce the searching space of roles from $O(|\mathcal{A}|R_f|)$ to $O(k|R_f|)$, where k ($k \ll |\mathcal{A}|$) is the amount of identified arguments and $|R_f|$ is the amount of roles of current predicted frame f. Taking our proposed pruning strategies into account, the overall searching space of all the subtasks, including frame identification, argument identification and role classification, is less than $O(|F| + 2|n(n+1)/2| + k|R_f|)$, since the window size used in arguments pruning stage is always less than or equal to the sentence length n.

Most of the previous studies [4, 10, 17, 27, 28] have enforced non-overlap constraint on the arguments. The constraint requires that the argument spans of the target do not overlap. The argument identification module can efficiently identify the arguments and their boundaries, but it can not guarantee that the arguments meet the non-overlap constraint, since the argument identification module identifies each argument independently in this work. To deal with this issue, we adopt a post-processing strategy that greedily select higher scoring arguments subject to the constraint in the overall inference process.

Given one testing sample x, the detailed inference process with non-overlap constraint is described in Algorithm 2. Firstly, the inference process predicts frame of the target t in the sentence x, as shown in line 6. Secondly, the process obtains the predicted argument set $\mathcal{A}^+$ and sorts them using their scores in descending order, as illustrated in lines 7-8. Thirdly. the process searches the overlap arguments, and then reserves the argument with highest score and removes other arguments overlapped with this argument from the set $\mathcal{A}^+$, as illustrated in lines 9-16. Finally, the inference process predicts the role for each predicted argument in $\mathcal{A}^+$, as shown in line 18.

Table 1. Numbers of instances in two datasets

	#exemplar	#train	#dev	#test
FN 1.5	153946	16621	2284	4428
FN 1.7	192431	19391	2272	6714

5 EXPERIMENTS

5.1 Datasets

We employ two full-text annotation datasets[3] from both standard FrameNet 1.5 [2] and FrameNet 1.7 [24] respectively, and follow the same train/development/test split as the existing work [3, 21, 27]. In addition, we also utilize the partially-annotated exemplar sentences (each exemplar sentence has only one annotated frame and its roles.) in FrameNet as training data, following the previous studies [3, 3, 21]. For the two full-text annotation datasets, we treat one annotated sentence for one target as one training sample. So, both the annotated sentence from full-text and exemplar sentence are processed into the same format. Table 1 shows the numbers of instances in two datasets. Note we have employed the standard evaluation script that measures the performance of full structure extraction, including three evalution metrics: *precision*, *recall* and *F1 score*[4].

5.2 Baselines for Performance Comparison

We have conducted extensive experiments to compare our proposed model with following ten models. Specifically, **Semafor** [4] is a well-known FSP system which uses a variety of syntactic features to construct two separate probabilistic models, i.e., frame identification model and role labeling model. The system applies the AD algorithm to conduct collective prediction of a target's arguments, incorporating declarative linguistic knowledge as constraints. **Hermann-14** [17] is a pipeline model that uses feature representation based on dependency path embedding, and utilizs WSABIE algorithm[31] to learn FI model and linear programming to search the optimal roles sequence. **Framat+context** [23] is an extension version of Framat[5], which adds extra context features in discourse and frame structure information to the model. **Täckström** uses dynamic programming algorithm to conduct inference with constraints based on a globally-normalized log-linear model using syntactic features [29]. **FitzGerald** extends the model of Täckström et al. by replacing its linear potential functions with a multi-layer neural network which maps arguments and roles of a given target into a shared embedding space [10]. **Yang and Mitchell** (SEQ) [33] is a sequence tagging model based on LSTM and CRF. **Yang and Mitchell** (REL) [33] is a relation model that uses the dependency path and dependency to represent the relations between target and its arguments. **Open-SESAME** [27] is a model based on Semi-Markov CRF that models a conditional distribution over labeled segmentations of an input sequence. **Peng et al**. (Basic) [21] is a jointly learning model using a structured hinge objective, and it is the **current SOTA model on FrameNet 1.7 dataset** for FSP task. **Chen et al**. [3] is a jointly learning model that adopts multi-decoder to handle the subtasks of FSP together, and it is the **current SOTA model on FrameNet 1.5 dataset** for FSP task.

[3] http://framenet.icsi.berkeley.edu
[4] http://www.cs.cmu.edu/ ark/SEMAFOR/eval/
[5] An open-source semantic role labeling tool proposed by Björkelund et al. (2010)

Table 2. Details of hyper-parameter settings with respect to different encoders (BERT/BiLSTM).

Parameters	**Values**
Encoder type	BERT-Base/BiLSTM
BiLSTM layers	- /2
Word embedding	- /300-d Glove[22]
Embedding dropout	- /0.2
Hidden size d_h,d_a,d_o	768/300
Frame embedding d_f	150
Span length embedding	150
Span width embedding	150
max_len	20
win_size	40
Dropout rate for FFNs	0.2
Pre-train epoch	50
Train epoch	150
Learning rate	1e-3
Optimizer	AdamW
Warmup proportion	0.1

Note that some of the existing systems may also report the results of ensemble learning models based on different types of ensemble learning methods, and some of the systems adopt multi-task learning framework using additional training data except for FrameNet dataset. In addition, the graph-based FSP model recently proposed by Lin et al.[18] is based on the assumption that the target word is unknown. The basic setting is different from our work, so we did not compare the performance since the comparison is unfair. For fair comparison, we only report the performance of models that were trained on FrameNet data only (without leveraging additional training data) and did not use an ensemble learning strategy. Furthermore, we will pay much attention on the performance comparison between our models and two jointly learning models [21] [3], since our model also adopts jointly learning strategy and these two models achieve the SOTA performance.

5.3 Parameter Settings

We adopt two alternative encoders, i.e., BERT and BiLSTM. Thus we have two groups of different parameter settings. In particular, BiLSTM adopts 2-layer structure and the word embeddings are initialized with 300-dimension Glove embeddings [22], while BERT uses the default settings of BERT-base. We set win_size=40 since the performance of the model on dev dataset remains stable after it is grater than 40. Following the previous work[3, 21, 27], the max_len is set to 20 for the fair comparison. We fine tune BERT encoder in our training phase. The detailed hyper-parameter settings are shown in Table 2.

Note the way that utilizes standard train samples and exemplar samples together to train our model has important effect on its performance. A straightforward way is to simply mix two kinds of samples together and the model is trained on this mixed data. However, this way may cause semantic drift since two kinds of samples are from different domains. We have proposed a different way, that is to conduct pre-training on partially-annotated exemplar samples first and then followed by continually training on standard training samples [3]. The pre-training epoch and training epoch are set to 50 and 150 respectively. We evaluate our model on the development test and select the best model for test purpose.

Table 3. Frame identification accuracy on FrameNet test dataset. '**ALL**' and '**Amb**' denote testing on test data and on ambiguous data respectively. '*****' denotes these models are trained on both the standard train data and exemplar data . The best performance among the models based on BERT encoder are denoted in bold face, and the best performance among the models based on BiLSTM or other non-BERT encoders are denoted with underline.

	FN 1.7		**FN 1.5**	
Models	**All**	**Amb**	**All**	**Amb**
Semafor	-	-	83.60	69.19
Hermann-14	-	-	88.41	73.10
SimpleFrameId	83.00	71.70	87.63	73.80
Open-SESAME	86.55	72.40	86.40	72.80
*Yang and Mitchell	-	-	88.20	75.70
*Peng et al.+BiLSTM	88.60	76.60	89.20	76.30
*Chen et al.+BiLSTM	88.65	76.70	89.40	76.70
*Chen et al.+BERT	90.10	78.90	90.50	79.10
***TaRFSP**				
+BiLSTM	88.71	76.80	88.78	76.20
+BERT	**91.71**	**82.45**	**92.07**	**81.20**

5.4 Overall Results

Following the existing work, we first evaluate the model performance on frame identification in terms of accuracy, and then evaluate the comprehensive performance with full structure extraction using precision, recall and F1 score. The results of frame identification include accuracy on both ambiguous data and all data. In particular, ambiguous data refers to the sentences in which the given target may evoke more than one possible frame in FrameNet. All data means all the test data including both ambiguous data and non-ambiguous data.

The results of frame identification are shown in table 3. All methods are partitioned into three groups: (1) pipeline models (first block), (2) jointly learning models (second block), and (3) our proposed models (third block) respectively. In particular, pipeline models are separately trained for frame identification, while jointly learning models refer to those models trained simultaneously for frame identification and frame-semantic labeling. Note that we do not list the FI accuarcy for some pipeline models mentioned in baselines, as these models directly use the results of FI of other models and did not report their results in the original papers. Overall, due to the error propagation problem, jointly learning models perform better than pipeline models. For fair comparison, we focus on the comparison between our models and three joint learning models, since they are all jointly learning models and adopt similar encoders. We observe that our TaRFSP+BERT achieves the best performance and outperforms Chen et al.+BERT and all of the other models by a relatively large margin. Compared with those baseline models based on BiLSTM, our TaRFSP+BiLSTM slightly outperforms Peng et al.+BERT on FrameNet 1.7, and slightly lags behind Chen et al.+BiLSTM in terms of accuracy. Considering BERT is now the state-of-the-art encoder with better representation capabilities, our method with BERT achieves the best performance for both benchmark datasets consistently, indicating it can be better used for FSP task than all the existing methods.

Table 4. Full structure extraction result on FrameNet 1.5 test data. Bold font indicates the best performance of model with BERT encoder, and the best performance of model with BiLSTM or other non-BERT encoder are denoted in underline. '*' denotes these models are trained on both the standard train data and exemplar data.

Models	**P**	**R**	**F1**
Semafor	69.2	65.1	67.1
Hermann-14	72.8	64.9	68.6
Framat+context	71.1	64.8	67.8
Täckström	75.4	65.8	70.3
FitzGerald et al.	74.8	65.5	69.9
Open-SESAME	71.0	67.8	69.4
*Yang and Mitchell (SEQ)	69.6	70.9	70.2
*Yang and Mitchell(REL)	77.1	68.7	72.7
*Peng et al.+BiLSTM	<u>79.2</u>	71.7	75.3
*Chen et al.+BiLSTM	75.1	<u>76.9</u>	76.0
***TaRFSP**+BiLSTM	78.1	75.4	<u>76.7</u>
*Chen et al.+BERT	78.2	**82.4**	80.2
***TaRFSP**+BERT	**82.5**	**82.4**	**82.5**

Table 5. Full structure extraction result on FrameNet 1.7 tset data. Bold font indicates best performance of model with BERT encoder, and the best performance of model with BiLSTM encoder are denoted in underline. † denote the result is reproduced according the original paper.

Models	**P**	**R**	**F1**
Peng et al.+BiLSTM	78.0	72.1	75.0
† Chen et al.+BiLSTM	74.0	77.1	75.5
TaRFSP+BiLSTM	<u>79.2</u>	<u>74.0</u>	<u>76.5</u>
† Chen et al.+BERT	80.0	81.8	80.9
TaRFSP+BERT	**83.8**	**82.1**	**82.9**

Full semantic structure extraction is a task, concerning the overall performance of FSP, which jointly evaluates the performance of frame identification and role classification that require exact match the boundaries of each argument as a role filler.

The results of full semantic structure extraction are shown in Table 4. All methods are partitioned into three groups: (1) pipeline models (first block), (2) jointly learning models with BiLSTM encoder(second block), and (3) jointly learning models with BERT encoder (third block) respectively. The results demonstrate that our proposed models achieve the best F1 score compared with the existing models on FrameNet 1.5 test data. Specifically, Our TaRFSP+BiLSTM model outperforms the current SOTA model with BiLSTM (i.e., Chen et al.+BiLSTM) by 0.7%, and our TaRFSP+BERT model outperforms the current SOTA model (i.e., Chen et al.+BERT) with BERT encoder by 2.3 %, indicating our method is able to perform frame-semantic parsing more accurately. For precision and recall, comparing with Peng et al.+BiLSTM model, our TaRFSP+BiLSTM model performs 1.1% worse in terms of precision but manages to achieve 3.7% high recall. Comparing with Chen et al.+BiLSTM model, our TaRFSP+BiLSTM model performs 1.5% worse in terms of recall but manages to perform 3.0% better in terms of precision. Our TaRFSP+BERT model achieves the improvement over the current SOTA BERT-based (Chen et al.+BERT) model by 5.7%,

Table 6. Full structure extraction results of ablation study on FrameNet 1.5 and FrameNet 1.7 tset data. The sign 'w/o' denotes the model without corresponding module. Each Δ represents the difference between our full model and the corresponding ablation model.

TaRFSP (our model)	**FN 1.5**		**FN1.7**	
	F1	Δ	F1	Δ
+BiLSTM	76.7	-	76.5	-
w/o frame interaction	74.4	***2.3***	74.5	***2.0***
w/o argument interaction	76.1	***0.6***	76.1	***0.4***
w/o non-overlap constraint	76.6	***0.1***	76.4	***0.1***
w/o relative distance	75.7	***1.0***	75.7	***0.8***
+BERT	82.5	-	82.9	-
w/o frame interaction	81.3	***1.2***	81.8	***1.1***
w/o argument interaction	82.2	***0.3***	82.7	***0.2***
w/o non-overlap constraint	82.4	***0.1***	82.8	***0.1***
w/o relative distance	81.7	***0.8***	82.3	***0.6***

with the comparable recall score, leading to 2.3% overall improvement. This is more important given that BERT is better than BiLSTM in many existing research [7, 30, 32] and our results also demonstrate that BERT based methods are around 4% better than BiLSTM based methods.

Note Chen et al. [3] do not conduct experiments on FrameNet 1.7 in their original paper. We reproduce the experiment on FrameNet1.7. For the performance on FrameNet 1.7 dataset, as shown in Table 5, our models outperform the current SOTA model in terms of all of the metrics. Overall, our models achieve the best score in terms of F1 consistently across both FrameNet 1.5 and FrameNet 1.7 test data.

To further analyze the features of our proposed model based on independency assumption, we compare it with the SOTA model proposed by Chen et al., which adopts structured prediction paradigm based on role interdependent assumption, and incremental identify arguments and roles so as to explicitly model the interdependency relations between roles. Modeling interdependency relations in the model is beneficial to achieve higher recall, but it seems to be unhelpful for improving the precision. We believe this phenomenon is caused by the error prorogation problem between roles, because the previously identified roles directly affect the subsequently identified roles. If the previous identified role is wrong, the subsequent identified roles may be also wrong. By contrast, our proposed TaRFSP model obtains better balance between recall and precision and achieves the better performance than Chen et al.'s model as well as other models. We attribute this mainly to our target-aware relation classification model based on role independency assumption. In other word, the model predicts the role of each argument independently without taking other roles of arguments of current frame into account, which can eliminate the error prorogation problems between roles.

6 ANALYSIS

To evaluate the performance of our models in different settings as well as the function of each model component, we conduct detailed ablation experiments on the two datasets.

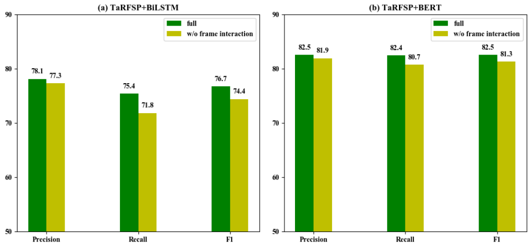


Fig. 5. All of the metrics comparison between full model and model without frame interaction on FrameNet 1.5 test data.

6.1 Effect of Frame Interaction

In FrameNet, the fine-grained roles are frame-specific roles, so the role and frame interaction is beneficial for the model to more precisely identify its roles in a given sentence. As shown in Table 6, no matter the BiLSTM-based model or BERT-based model, their F1 scores have declined in some degree without the frame interaction, although there are less effect on BERT-based model than BiLSTM-based model, possible due to BERT's strong representation capacity. For the BiLSTM-based model, the improvement over F1 score achieves 2.0%, with the help of frame interaction. So we can conclude that our proposed lightweight frame interaction mechanism is effective and feasible for improving the performance of FSP models, indicating that semantic roles dependent on its frame that defines the semantic scenarios is reasonable.

How does the frame interaction affect the performance of our models? We answer this question through analyzing the other specific metrics thoroughly which influence the F1 metric. As shown in Figure 5, both the recall and precision of the BiLSTM-based model and BERT-based model have decreased significantly without the frame interaction, and we also observe that the influence on recall is much higher that on precision. So, with frame interaction, the models can not only more accurately predict the roles for the predicted arguments, but also identify more arguments to fill the roles. The models without frame interaction tend to predict those predicted arguments whose roles are hard to identify as *Null* role. One reason is that the imbalance role distribution of training samples in which the *Null* role labeled samples account for the vast majority. The other reason is that models need to predict the role for a certain predicted argument from all of the roles of FrameNet. On the other hand, with the frame interaction, the model tends to predict the role for a certain predicted argument from a small set of frame-specific roles, which makes the prediction of role more easier, and the model tends to assign the potential argument a exact frame-specific role instead of *Null* role.

6.2 Effect of Argument Interaction

In this study, the argument identification module is mainly used to identify the arguments of the given target. Meanwhile, we also design a lightweight explicit interaction mechanism with the role classification module through simply passing the output of the biaffine layer of the argument identification module to the role classification module, as described in Section 3.7.

As shown in Table 6, no matter with the BiLSTM-based model or BERT-based model, their F1 scores have declined in some degree without the argument interaction, although there are less effect on BERT-based model than BiLSTM-based model, possible due to BERT's strong representation capacity. Overall, the

Table 7. Full structure extraction result and frame identification accuracy considering the influences of different way using the exemplar data. Each Δ represents the metric difference between two models in corresponding settings. The signs *pre*, *mix* and *w/o exe* denote pre-training, mixture and without using exemplar data respectively.

TaRFSP	**FN 1.5**		**FN 1.7**	
	F1	Acc.	F1	Acc.
+BiLSTM (w/o exe)	73.7	88.4	74.2	88.0
+BiLSTM (mix)	75.3	87.9	75.7	87.9
+BiLSTM (pre)	**76.7**	**88.8**	**76.5**	**88.7**
(mix)-(w/o exe) Δ	1.6	-0.5	1.5	-0.1
(pre)-(w/o exe) Δ	3.0	0.4	2.3	0.7
+BERT (w/o exe)	81.2	91.4	81.4	91.3
+BERT (mix)	81.9	91.2	82.5	91.6
+BERT (pre)	**82.5**	**92.1**	**82.9**	**91.7**
(mix)-(w/o exe) Δ	0.7	-0.2	1.1	0.3
(pre)-(w/o exe) Δ	1.3	0.7	1.5	0.4

interaction between frame identification and role classification module is beneficial for improving model's overall performance.

6.3 Effect of Non-overlap Constraint

Table 6 shows that the F1 scores of both BiLSTM-based models and BERT-based models have dropped without the non-overlap constraint in general. Although the impact of the non-overlap constraint on the overall performance seems to be small, it is still necessary for keeping the well-formed argument structure of the given target.

In addition, the experimental results also demonstrate that: (1) our proposed method based on target-aware relation classification for argument identification has already eliminated the overlap arguments to some extend, since there are very few overlapping arguments. (2) Our proposed inference algorithm with non-overlap constraint is effective to handle the overlap arguments, since it can not only eliminate the overlap arguments but also improve the model's performance in terms of F1 score.

6.4 Effect of Relative Distance

The relative distance represent the span's position and distance relation corresponding to the target. To test its effect, we separately train the models without using the relative distance information. As show in Table 6, no matter with the BiLSTM-based model or BERT-based model, their F1 scores have declined in some degree without incorporating relative distance into the span representation. The experimental results show that the relative distance information is beneficial for improving the model's overall performance.

6.5 Effect of Exemplar Data and Pre-training

The previous studies have shown that the exemplar data included in FrameNet is useful for enhancing the FSP model's performance, although these exemplars as a part of FrameNet are initially used to illustrate the meaning of frames and collected from a variety of domains. It is worth discussing how to use the exemplar data. We simply adopt two types of way in using exemplar data: (1) mixture: mixing the exemplar with

Table 8. The simulation results of inference speed. "tgt/s" denotes the FSP speed for one target per second.

Model	**FN1.5**	**FN1.7**
Peng et al.+BiLSTM	20.24 tgt/s	20.78 tgt/s
Chen et al.+BiLSTM	28.15 tgt/s	28.56 tgt/s
TaRFSP+BiLSTM	36.12 tgt/s	36.68 tgt/s

Table 9. Percentage of errors made by BERT-based and BiLSTM-based models on the FN1.5 development dataset.

Error type	**Description**	**TaRFSP+**	
		BiLSTM	BERT
Frame error	Frame misprediction	10.3	8.1
Role error	Matching span with incorrect role.	14.4	11.3
Span error	Matching role with incorrect span boundary	12.3	7.0
Extra predicted arguments	Predicted argument that does not overlap any gold argument	19.8	21.2
Missing arguments	Gold argument that does not overlap any predicted argument	35.0	22.6

train data; (2) pre-training: pre-train on exemplar data followed by training on our training data. As shown in Table 7, compared with the model trained without using exemplar data, no matter which way (mixture or pre-training) we adopt, the F1 scores of both the models trained using extra exemplar data have been improved significantly. Specifically, for the BiLSTM-based model trained with the way of mixture, the improvement over F1 score achieves 1.6% on FrameNet 1.5 test dataset, while the improvement over F1 score achieves 3.0% on FrameNet 1.5 test dataset for the BiLSTM-based model trained with pre-training. We also notice that the mixture strategy may be harmful to FI in terms of accuracy, although it is beneficial for overall performance in terms of F1 score. Therefore, we can conclude that pre-training is more effective for improving the model's overall performance than mixture. It is probably because pre-training strategy is beneficial to handle the semantic drift problem, and the mixture strategy can not effectively handle this problem well which can produce negative effect on the performance of FI.

6.6 Inference speed analysis

Table 8 shows the running speed of our model and two strong baseline models. Experimental results are all obtained by running models on a single V100 GPU. For Peng et al.+BiLSTM model, we simulate the inference process by replacing the original score function for each span with our span-based score function and use linear programs to search the optimal solution just like the original paper. Although we can not accurately assess the inference speed of each model, we can find the tendency that our proposed method has the advantage in inference speed. Furthermore, we find that the pruning modules used in argument identification are effective since the inference speed has dropped to 32 tgt/s when they are removed from the model.

6.7 Error analysis

To further analyze the performance of our proposed method, we conduct the error analysis and categorize the error type into for categories [3, 21]. We conduct the error analysis on development dataset of FrameNet 1.5. As show in Table 9, no matter with the BiLSTM-based model or BERT-based model, frame error, role

Table 10. Case analysis for four sentences. In each block, the first two lines are sentences with golden labels and predicted labels respectively, and the third line describes the error types and causes of the predicted results. The span in bracket are arguments and the bold words in bracket are targets. The roles of arguments and frames evoked by targets are denoted as subscript of their corresponding right bracket.

1	• [Steve, who was sitting next to John,]$_{Traveller}$ [**got**]$_{Disembarking}$ down [in Rome]$_{Place}$. • [Steve, who was sitting next to John,]$_{Theme}$ [**got**]$_{Arriving}$ down [in Rome]$_{Goal}$. • **Frame error**:*Arriving*, **Role error**: *Theme*, *Goal*
2	• All parties have agreed that they seek a non-unclear Korean peninsula, but it [**remains**]$_{State\_continue}$ [unclear]$_{State}$ [how this objective will be achieved]$_{Entity}$. • All parties have agreed that they seek a non-unclear Korean peninsula, but it [**remains**]$_{State\_continue}$ [unclear how this objective will be achieved]$_{State}$. • **Span error**: [unclear how this objective will be achieved]$_{State}$
3	• Since April 2003, multilateral talks have been held in Beijing to [**resolve**]$_{Resolve\_problem}$ [the nuclear crisis]$_{problem}$. • Since April 2003, [multilateral talks]$_{Cause}$ have been held in Beijing to [**resolve**]$_{Resolve\_problem}$ [the nuclear crisis]$_{problem}$. • **Extra predicted argument**: [multilateral talks]$_{Cause}$
4	• He also sold [the XYZ-11]$_{Part}$, the [key]$_{Part\_Prop}$ [**part**]$_{Part\_whole}$ necessary for the trigger. • He also sold the XYZ-11, the [key]$_{Part\_Prop}$ [**part**]$_{Part\_whole}$ necessary for the trigger. • **Missing argument**: [the XYZ-11]$_{Part}$

error and span error account for a relatively small fractions, which missing arguments and extra predicted arguments account for most of the errors. In combination with the error case studies, as shown in Table 10, we observe that these errors are closely related. Firstly, The error of frame identification directly affects the errors of role classification since roles are dependent on frame. For example, although the span boundaries of two arguments are correctly predicted, the roles of these arguments are incorrectly predicted as *Theme* and *Goal* due to the wrong prediction of frame evoked by target ***got*** in case 1. Secondly, span error can directly causes the errors of missing arguments and extra predicted arguments. For example, in case 2, due to the incorrect boundary of argument '*unclear how this objective will be achieved*' predicted by the model, this argument becomes extra predicted argument, and the argument '*unclear*' and '*how this objective will be achieved*' are missing arguments. Thus it can be seen that frame prediction and span boundary identification of arguments are two bottlenecks of improving the performance of FSP.

Due to the large number of frames and frame-specific roles in FrameNet knowledge base, we observe that frames and frame-specific roles have serious long tail distribution problem that considerable frames and roles appear in train dataset very few times. Therefore, the learned model tends to predict frames, argument spans and corresponding roles as these labels with high frequency in train dataset. For example, the frame *Disembarking* only appears one time in train dataset, while the frame *Arriving* appears many times. So the model incorrectly predicts the frame of target ***got*** in case 1 as *Arriving* instead of *Disembarking*. The role *Entity* of frame *State_continue* usually precedes the target and role *State* is after the target in a sentence in train dataset. Thus the model incorrectly predicts the whole span '*unclear how this objective will be achieved*' as argument for role *State*. In addition, some uncommon words or phrases make the model unable to accurately obtain its semantic representations, so that they are predicted as *null* arguments (i.e. missing arguments) or wrong roles. For example, the argument '*the XYZ-11*' in case 4 is missing, since it is an uncommon phrase. How to solve the long tail distribution problem will be a challenge work for FSP.

7 CONCLUSION

In this work, we propose a innovative relation-based paradigm for FSP, which relaxes the role interdependent assumption in structured learning paradigm, and regards the frame semantic role labeling as a relation classification task. Based on this paradigm, we present a novel and lightweight jointly learning framework for FSP, which decomposes the FSP task into three subtasks: frame identification, argument identification and role classification, which incorporates lightweight interaction mechanism among the three subtasks. Our detailed experimental analysis shows that interaction with predicted frame is very important to improve the overall performance of FSP models, and interaction with argument identification is useful for enhancing the model's precision and reducing the inference time. The exemplar data contained in FrameNet is also useful for improving our model's performance, especially for BiLSTM-based models. Extensive experimental results demonstrate our relation-based jointly learning model outperforms ten state-of-the-art FSP models by a large margin in F1 score. As such, relation-based method for FSP is a promising direction given our proposed jointly learning framework is both efficient and accurate.

For our further work, we will explore a more sophisticated method to handle the long-tail distribution problem so as to improve the performance of FSP, especially in enhancing the performance of two bottleneck subtasks (i.e. frame identification and span boundary identification of arguments). For example, we will try to utilize more extra data and linguistic knowledge and adopt prompt learning paradigm to construct the model.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their helpful comments and suggestions. This work was supported by the National Natural Science Foundation of China (No.61936012), the Research Fund of Guangxi Key Lab of Multi-source Information Mining & Security (NO. MIMS22-10) and the Foundation of Shanxi Vocational University of Engineering Science and Technology NO.KJ202203).

REFERENCES

[1] 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)*. Association for Computational Linguistics, Prague, Czech Republic, 99–104.

[2] 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*. Association for Computational Linguistics, Montreal, Quebec, Canada, 86–90. https://doi.org/10.3115/980845.980860

[3] Xudong Chen, Ce Zheng, and Baobao Chang. 2021. Joint Multi-Decoder Framework with Hierarchical Pointer Network for Frame Semantic Parsing. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, Online, 2570–2578. https://doi.org/10.18653/v1/2021.findings-acl.227

[4] Dipanjan Das, Desai Chen, André F. T. Martins, Nathan Schneider, and Noah A. Smith. 2014. Frame-Semantic Parsing. *Comput. Linguistics* 40, 1 (2014), 9–56. https://doi.org/10.1162/COLI_a_00163

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[6] Timothy Dozat and Christopher D. Manning. 2017. Deep Biaffine Attention for Neural Dependency Parsing. *ArXiv* abs/1611.01734 (2017).

[7] Markus Eberts and Adrian Ulges. 2019. Span-based Joint Entity and Relation Extraction with Transformer Pre-training. *CoRR* abs/1909.07755 (2019). arXiv:1909.07755

[8] C. Fillmore and Collin F. Baker. 2001. Frame semantics for text understanding. In *Proceedings of the NAACL WordNet and Other Lexical Resources Workshop*. Association for Computational Linguistics, Pittsburgh, Pennsylvania.

[9] C. J. Fillmore. 1985. Frames and the semantics of understanding. *Quaderni di Semantica* 6, 2 (1985), 222–254.

[10] 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*. Association for Computational Linguistics, Lisbon, Portugal, 960–970. https://doi.org/10.18653/v1/D15-1112

[11] Daniel Gildea and Daniel Jurafsky. 2002. Automatic Labeling of Semantic Roles. *Comput. Linguist.* 28, 3 (Sept. 2002), 245288. https://doi.org/10.1162/089120102760275983

[12] Yong Guan, Shaoru Guo, Ru Li, Xiaoli Li, and Hu Zhang. 2021. Frame Semantics guided network for Abstractive Sentence Summarization. *Knowledge-Based Systems* 221 (2021), 106973. https://doi.org/10.1016/j.knosys.2021.106973

[13] Shaoru Guo, Yong Guan, Ru Li, Xiaoli Li, and Hongye Tan. 2020. Incorporating Syntax and Frame Semantics in Neural Network for Machine Reading Comprehension. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Barcelona, Spain (Online), 2635–2641. https://doi.org/10.18653/v1/2020.coling-main.237

[14] Shaoru Guo, Ru Li, Hongye Tan, Xiaoli Li, Yong Guan, Hongyan Zhao, and Yueping Zhang. 2020. A Frame-based Sentence Representation for Machine Reading Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 891–896. https://doi.org/10.18653/v1/2020.acl-main.83

[15] Silvana Hartmann, Ilia Kuznetsov, Teresa Martin, 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*. Association for Computational Linguistics, Valencia, Spain, 471–482.

[16] Karl Moritz Hermann, Dipanjan Das, Jason Weston, and Kuzman Ganchev. 2014. Semantic Frame Identification with Distributed Word Representations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, 1448–1458. https://doi.org/10.3115/v1/P14-1136

[17] Karl Moritz Hermann, Dipanjan Das, Jason Weston, and Kuzman Ganchev. 2014. Semantic Frame Identification with Distributed Word Representations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*. The Association for Computer Linguistics, 1448–1458. https://doi.org/10.3115/v1/p14-1136

[18] ZhiChao Lin, Yueheng Sun, and Meishan Zhang. 2021. A Graph-Based Neural Model for End-to-End Frame Semantic Parsing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 3864–3874. https://doi.org/10.18653/v1/2021.emnlp-main.314

[19] Shulin Liu, Yubo Chen, Shizhu He, Kang Liu, and Jun Zhao. 2016. Leveraging FrameNet to Improve Automatic Event Detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, 2134–2143. https://doi.org/10.18653/v1/P16-1201

[20] Hiroki Ouchi, Hiroyuki Shindo, and Yuji Matsumoto. 2018. A Span Selection Model for Semantic Role Labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 1630–1642. https://doi.org/10.18653/v1/D18-1191

[21] 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)*. Association for Computational Linguistics, New Orleans, Louisiana, 1492–1502. https://doi.org/10.18653/v1/N18-1135

[22] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1532–1543. https://doi.org/10.3115/v1/D14-1162

[23] Michael Roth and Mirella Lapata. 2015. Context-aware Frame-Semantic Role Labeling. *Transactions of the Association for Computational Linguistics* 3 (2015), 449–460. https://doi.org/10.1162/tacl_a_00150

[24] Josef Ruppenhofer, Michael Ellsworth, Miriam R.L.Petruck, Christopher R. Johnson, Collin F. Baker, and Jan Scheffczyk. 2016. *Framenet ii: extended theory and practice.* https://framenet2.icsi.berkeley.edu/docs/r1.7/book.pdf.

[25] Tianze Shi, Igor Malioutov, and Ozan Irsoy. 2020. Semantic Role Labeling as Syntactic Dependency Parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 7551–7571. https://doi.org/10.18653/v1/2020.emnlp-main.610

[26] Xuefeng Su, Ru Li, Xiaoli Li, Jeff Z. Pan, Hu Zhang, Qinghua Chai, and Xiaoqi Han. 2021. A Knowledge-Guided Framework for Frame Identification. In *Proceedings of the 59th Annual Meeting of the Association for Computational*

Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Association for Computational Linguistics, Online, 5230–5240. https://doi.org/10.18653/v1/2021.acl-long.407

[27] Swabha Swayamdipta, Sam Thomson, Chris Dyer, and Noah A. Smith. 2017. Frame-Semantic Parsing with Softmax-Margin Segmental RNNs and a Syntactic Scaffold. *CoRR* abs/1706.09528 (2017). arXiv:1706.09528

[28] Oscar Täckström, Kuzman Ganchev, and Dipanjan Das. 2015. Efficient Inference and Structured Learning for Semantic Role Labeling. *Trans. Assoc. Comput. Linguistics* 3 (2015), 29–41.

[29] Oscar Täckström, Kuzman Ganchev, and Dipanjan Das. 2015. Efficient Inference and Structured Learning for Semantic Role Labeling. *Transactions of the Association for Computational Linguistics* 3 (2015), 29–41. https://doi.org/10.1162/tacl_a_00120

[30] Yijun Wang, Changzhi Sun, Yuanbin Wu, Hao Zhou, Lei Li, and Junchi Yan. 2021. UniRE: A Unified Label Space for Entity Relation Extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Online, 220–231. https://doi.org/10.18653/v1/2021.acl-long.19

[31] Jason Weston, Samy Bengio, and Nicolas Usunier. 2011. N.: Wsabie: Scaling up to large vocabulary image annotation. In *International Joint Conference on Artificial Intelligence*.

[32] Lu Xu, Yew Ken Chia, and Lidong Bing. 2021. Learning Span-Level Interactions for Aspect Sentiment Triplet Extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Online, 4755–4766. https://doi.org/10.18653/v1/2021.acl-long.367

[33] 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*. 1247–1256.

[34] Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Named Entity Recognition as Dependency Parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 6470–6476. https://doi.org/10.18653/v1/2020.acl-main.577

[35] Hongyan Zhao, Ru Li, Xiaoli Li, and Hongye Tan. 2020. CFSRE: Context-aware based on frame-semantics for distantly supervised relation extraction. *Knowledge-Based Systems* 210 (2020), 106480. https://doi.org/10.1016/j.knosys.2020.106480

[36] Zexuan Zhong and Danqi Chen. 2021. A Frustratingly Easy Approach for Entity and Relation Extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 50–61. https://doi.org/10.18653/v1/2021.naacl-main.5