

A Unified Framework for Frame-Semantic Parsing Based on Marker Attention

Xuefeng Su^{1,2}, Ru Li^{2*}, Xiaoli Li³, Zhichao Yan²

¹School of Modern Logistics, Shanxi Vocational University of Engineering Science and Technology, Jinzhong 030619, China

²School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China

³Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore 138632, Singapore

Keywords: Frame-semantic parsing; Marker attention; Graph based pruning; FrameNet

Citation: Su X.F., Li R., Li X.L., et al.: A Unified Framework for Frame-Semantic Parsing Based on Marker Attention. Data Intelligence, 2026, Vol. 8, Art. No.: 20240053. DOI: <https://doi.org/10.3724/2096-7004.di.2024.0053>

ABSTRACT

Frame-semantic Parsing (FSP) aims to extract frame-semantic structures from text. The task involves three subtasks: target identification (TI), frame identification (FI) and frame-semantic role labeling (FSRL). Most of the previous studies predominantly focus on one or two of the three subtasks, and limited studies have focus on constructing a unified framework to systematically solve the whole FSP task. In addition, how to efficiently utilize knowledge of lexical units and improve the span representations for FSP is still an open question. Therefore, this work presents a *novel unified framework for FSP based on marker attention*, in which special markers are strategically inserted into the input sequence of the encoder to incorporate the lexical units knowledge and frame information, as well as to enhance the span representation through attention mechanisms, without using any additional complicated module. Experiments on two public benchmarks demonstrate the effectiveness of our proposed framework.

1. INTRODUCTION

Frame-semantic Parsing (FSP) is a semantic analysis task based on FrameNet [1-2], aiming to extract frame-semantic structures from text. Figure 1 shows an example, where all targets, their associated semantic frames, arguments and respective roles in the sentence are depicted. The salient feature of FSP is that roles are frame specific, and the relationships between frames and roles are defined in FrameNet by cognitive linguists. The parsing results of FSP yield finegrained semantic information embedded within

*Corresponding author: Ru Li (E-mail: liru@sxu.edu.cn; ORCID: 0000-0003-4507-8736).

the text, which has shown to be useful in event recognition [3], machine reading comprehension [4-5], relation extraction [6], and text generation [7-9] etc.

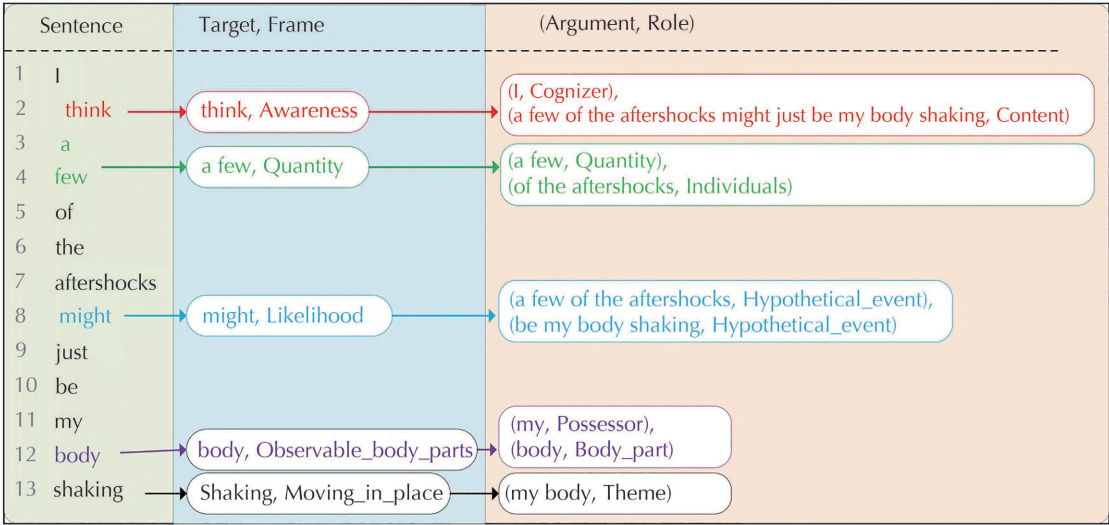


Figure 1. An example for FSP. The first block is the given sentence, in which there are five groups of frame-semantic structures (denoted in five different colors) expressing five different semantic scenarios that need to be extracted by FSP model. Each group includes a target, a frame and a (argument, role) pair set. Take the first group (in red) as an example, in which Awareness is the evoked frame by target “think”, while “I” and “a few of the aftershocks might just be my body shaking” are arguments acting as *Cognizer* and *Content* roles in the frame, respectively.

FSP is usually divided into three subtasks [10]: 1) target identification (TI), in which frameevoking targets are predicted in the sentence, 2) frame identification (FI), in which the evoked frame is selected for each target, 3) frame semantic role labeling (FSRL), in which arguments of each frame are identified and labeled with a role defined in that frame. Some studies [10-15] focus on individual subtasks, whereas others [16-19] simultaneously handle FC and FSRL subtasks, assuming the availability of all targets in advance. Recently, Lin [20] proposes a graph-based end-to-end FSP model addressing all three subtasks jointly.

While the existing studies have achieved notable progress in FSP, they still have three weaknesses. Firstly, only a limited number of studies [10, 15, 20] investigate the TI subtask, while most of the studies assume that all the targets are given in advance which is unrealistic in some real application scenarios. Furthermore, the knowledge about lexical units and frames in FrameNet is not fully exploited. Secondly, how to accomplish FSP automatically in a unified framework is still under-investigated. Although Lin [20] uses one model to tackle all the subtasks jointly, the model still suffers from span missing problem as it fails to handle overlapping spans between targets and arguments. Finally, how to represent spans in FSP poses a challenge. The widely used boundary-based span representation method is inadequate to model the comprehensive meaning of an argument, especially for those longer arguments.

To address the above limitations, firstly, we reformulate the FSP task as two subtasks: the joint TI and FI task and the FSRL task (see Sec. 4). Because TI and FI are both span-level classification problem while FSRL is a relation classification problem, it is not suitable to model them together due to their different structures. Secondly, based on the formulation, we propose a two-stage method to accomplish FSP with a unified model framework. The framework integrates a marker-based method [21-23] to incorporate lexical units knowledge, thereby enhancing performance in joint TI and FI. Additionally, it employs a marker-based method to integrate frame information, thereby enriching span representation and improving FSRL performance. The two-stage approach can also address the span missing issue since targets and arguments are identified in different stage.

The contributions of our work are as follows:

- (1) We reformulate the FSP task and propose a novel two-stage method to accomplish FSP with a unified model framework.
- (2) We propose a model that jointly handles TI and FI, leveraging lexical unit knowledge and enriching the span representation through paired levitated markers and marker-based attention.
- (3) We propose a one-stage FSRL model that incorporates frame information and enhances the span representation through single levitated markers. Additionally, we present a graph-based pruning method to reduce the potential argument spans, facilitating efficient one-stage FSRL.
- (4) Our proposed model achieves state-of-the-art performance on two benchmark datasets in an end-to-end setting, and it also attains state-of-the-art performance in two of the three individual subtasks.

2. BACKGROUND

2.1 FrameNet

FrameNet is a linguistic resource grounded on the theory of Frame Semantics [24], serving as a comprehensive linguistic repository, housing rich information on frames, frame elements, and lexical units.

A **frame** is a schematic representation of a situation or scenario involving diverse participants, props, and other conceptual roles, each of which is a frame element. These elements are meticulously defined by linguists within FrameNet. Essentially, a **frame element** is the framespecific semantic role which is the basic unit of a frame. A frame in FrameNet usually includes a list of **lexical units** (e.g. *express.v*, *put into words.v*, *expression.n*, etc.), which are associated with words and phrases that can potentially evoke the frame in a natural language utterance. Table 1 shows the descriptions of frames *Expressing publicly* and *Awareness* within FrameNet. Each description consists of a textual frame definition, a set of frame elements, a set of lexical units, and a group of annotated exemplars.

Table 1. Two examples for frame description in FrameNet. The bold words in the annotated exemplars are targets that can evoke frame, i.e., *aired* and *aware*. The colored blocks represent the arguments of the given frame, with their respective roles indicated by subscripts. Due to space constraints, lexical units, frame elements, and annotated exemplars are partially provided.

Frame: Expressing publicly
Definition: A <i>Communicator</i> publicly communicates some difficult-to-express <i>Content</i> which they (or a social group of which they are a member) have had for some time.
Frame elements: <i>Communicator</i> , <i>Content</i> , <i>Medium</i> , <i>Addressee</i> , <i>Circumstances</i> , <i>Depictive</i> , etc.
Lexical units: <i>air.v</i> , <i>express.v</i> , <i>expression.n</i> , <i>articulate.v</i> , <i>voice.v</i> , <i>put into words.v</i> , <i>give voice.v</i> , etc.
Annotated exemplar: Members _{Communicator} aired their grievances _{Content} and resented the way higher education was being dealt with.
Frame: Awareness
Definition: A <i>Cognizer</i> has a piece of <i>Content</i> in their model of the world.
Frame elements: <i>Cognizer</i> , <i>Content</i> , <i>Expressor</i> , <i>Degree</i> , <i>Evidence</i> , <i>Explanation</i> , etc.
Lexical units: <i>aware.a</i> , <i>awareness.n</i> , <i>believe.v</i> , <i>think.v</i> , <i>conceive.v</i> , <i>comprehend.v</i> , <i>conception.v</i> , etc.
Annotated exemplar: Your boss _{Cognizer} is aware of your commitment _{Content} .

There are two versions of FrameNet: version 1.5 and 1.7. In particular, version 1.5 encompasses 1,019 frames involving 1,170 frame elements, while version 1.7 features 1,221 frames involving 1,285 frame elements.

2.2 Frame-semantic Structure

The goal of FSP task is to extract frame-semantic structures from text. A frame-semantic structure signifies a specific situation, event or concept, which consists of a target, a frame, a set of arguments and their corresponding frame-specific roles.

A **target** typically refers to a word or phrase that strongly implies the type of situation, event, or concept (referred to as a frame) within a given sentence. In simpler terms, a target can evoke or trigger a specific frame. For instance, in the sentence illustrated in Figure 1, the word “think” serves as a target, suggesting the presence of a situation where someone is contemplating ideas. This scenario closely aligns with the definition of the *Awareness* frame, leading us to identify the word “think” as the target evoking the *Awareness* frame in the FSP task. For each target in the sentence, leveraging lemmatization and part-of-speech tagging tools allows us to derive its corresponding form as lexical unit. This form enables us to establish connections between the target and one of the mappings in the form of (lexical unit, frames). For example, in the sentence depicted in Figure 1, the word “think” transforms into “think.v”, thereby associating the target “think” with the mapping (*express.v*, [*Expressing publicly*, *Encoding*, *Sending*]).

An **argument** typically refers to a word, phrase, or clause that acts as a specific role within the current situation. For instance, in Figure 1, “I” serves as an argument of the target *think*, embodying the role of *Cognizer* within the *Awareness* frame in this sentence. Similarly, “a few of the aftershocks might just be my body shaking” acts as another argument of the target *think*, fulfilling the *Content* role within the *Awareness* frame in the sentence.

For the sentence “I think a few of the aftershocks might just be my body shaking”, depicted in Figure 1, it comprises five distinct groups of frame-semantic structures. Each structure consists of a (target, frame) pair, alongside a series of a collection of (argument, role) pairs.

3. RELATED WORK

3.1 Frame-Semantic Parsing (FSP)

FSP has drawn great attention since the SemEval 2007 shared task [25] was introduced. It encompasses three subtasks: TI, FC and FSRL. Most of the previous studies [10-15] focus on the FI or FSRL subtasks individually, whereas others [16-19] simultaneously handle FC and FSRL subtasks, assuming the availability of all targets in advance. Limited works focus on the TI and the end-to-end FSP. Das [10] proposes a TI model based on a featureengineering based method, while Emanuele [15] proposes a sequence labeling model for TI. Lin [20] proposes a graph-based end-to-end FSP model addressing all three subtasks jointly.

Early studies [10-12] employ feature-engineering based methods construct models. Features usually include morphology (such as POS, lemma of the target), syntax (such as dependencies, child words, parent words of targets) and semantics (such as synonyms and collocations of targets) features. Recent studies [14-15, 17-18] adopt pre-trained models such as BERT [26] and RoBERTa [27], or deep neural networks such as LSTM and biLSTM, as the backbone to construct FSP models.

As for FI subtask, the previous studies can be divided into two categories according to whether frame knowledge is utilized, such as frame identification, frame relations. The methods without using frame knowledge [10-11, 28] usually model FI as a span-based classification problem, while the methods using frame knowledge [29-31] usually model FI as a semantic similarity problem.

The existing methods of FSRL are roughly divided into three categories: chart-based methods, sequence-labeling based methods, and graph-based methods. Chart-based methods [10, 13, 16] adopt local scoring functions to model each span (candidate arguments) of every segmentation of the sentence given a target, and finally search a global optimal solution of segmentation scores using optimization algorithm. Sequence-labeling based methods can be divided into two categories, i.e., token-level sequence labeling and span-label sequence labeling. Token-label sequence labeling methods [14] adopt traditional BIO tagging schemes. Span-label sequence labeling [15, 17] methods usually consist of two stages, i.e., argument identification (AI) and argument classification (AC). Graph-based methods [18-20] regard FSRL task as a process of graph construction, in which the model incrementally identifies the nodes and their

relations, where targets and arguments are regarded as nodes, and roles are regarded as relations between them. Most recently, Zheng [32] proposes a query-based method to conduct FSRL.

Since both targets and arguments are spans, how to represent spans is a basic problem for FSP. Boundary-based representation method, utilizing the concatenation of the representations of two boundary words of the span as the span representation, is the widely used method. In addition, Swayamdipta [13] and Lin [20] use LSTM to enhance the boundary-based span representations. Su [18] proposes a target-aware span representation method.

Overall, the existing studies have achieved notable progress in FSP. The majority of research efforts focus on one or two of the three subtasks, achieving the extraction of partial information of frame-semantic structure. The research on how to achieve all three subtasks and extract complete frame-semantic structures within a unified framework is still insufficient. Furthermore, some fundamental issues, such as knowledge incorporation and span representation still require further in-depth research.

3.2 Markers-based representation

Inserting special markers (e.g. [MASK]) into text to achieve special function has been very common since the Transformer [33] was released, while levitated Marker was first proposed by Zhong [21] for solving an efficient batch computation by inserting many markers simultaneously. To be specific, if multiple markers are inserted into a sentence simultaneously, the semantic continuity of the sentence would be disrupted. However, by attaching the markers to the end of the sentence, more markers can be seamlessly integrated into the sentence. Ye [23] uses levitated markers to represent entities in relation classification task.

In this work, we construct a unified framework for FSP based on levitated markers, in which the (lexical unit, frame) knowledge can be conveniently incorporated and the span representations can be enhanced. In addition, the framework can efficiently conduct all of the three subtasks of FSP and extract complete frame-semantic structures through carefully designing the input of the encoder without using additional complicated modules.

4. PROBLEM FORMULATION

Formally, given a sentence X with n words $w_1, w_2, \dots, w_n$, FSP aims to extract a set of framesemantic structures $Y = \{y_1, y_2, \dots\}$, where each structure $y_i = (TF_i, TAR_i)$, $1 \leq i \leq |Y|$:

In particular, $TF_i = (t_i, f_i)$, where each t_i is a frame-evoking target that could be either a word or phrase in sentence X , and $f_i \in F$ is the frame evoked by t_i . F is the frame set derived from Frame-semantic knowledge base, such as FrameNet.

$TAR_i = \{(s_{i,1}, r_{i,1}), \dots, (s_{i,rdi}, r_{i,rdi})\}$, where each $s_{i,*}$ represents a span serving as an *argument* of the target t_i , and $r_{i,*} \in R_{fi}$ denotes the *associated frame-semantic roles*. Here, R encompasses the complete set of *roles* extracted from a frame-semantic knowledge base, while $R_{fi} \subset R$ specifies the frame-specific roles related to frame f_i .

For instance, $y_1 = (TF_1, TAR_1) = ((\text{think}, \text{Awareness}), \{(I, \text{Cognizer}), (\text{a few of the aftershocks might just be my body shaking}, \text{Content})\})$ is one of the five frame-semantic structures in the sentence from Figure 1 that we aim to extract automatically.

5. THE PROPOSED APPROACH

We propose a two-stage frame-semantic parsing method based on marker attention to respectively extract (target, frame) pairs TF_i ($1 \leq i \leq m$) and (argument, roles) pairs set TAR_i for each TF_i . We construct two separate models for each stage, i.e., joint target and frame identification model and frame-semantic role labeling model.

5.1 First stage: Joint Target and Frame Identification Based on Marker Attention

We propose a joint TI and FI model based on marker attention (**TFIMA**), which incorporates (lexical unit, frames) mapping knowledge to levitated markers and further combines markers to potential targets through marker based attention. The diagram of TFIMA is shown in Figure 2.

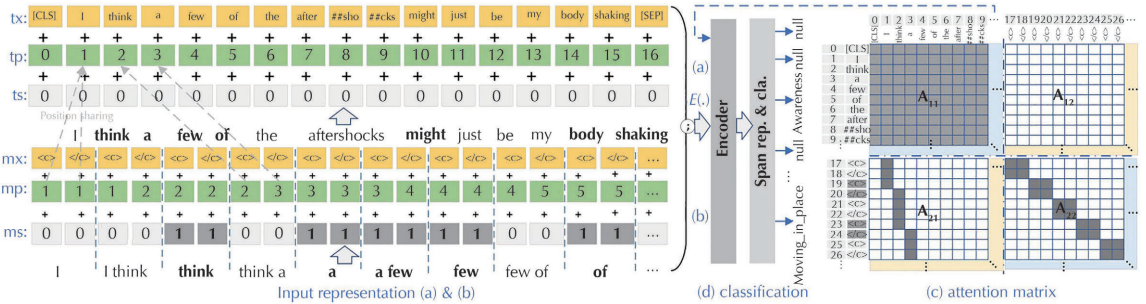


Figure 2. Diagram of TFIMA. The input sequence includes tokens (tx) and levitated markers (mx), the input positions include token positions (tp) and marker positions (mp), and the input saliency vector includes two parts: full zero vector (ts) for tokens and saliency vector (ms) for levitated markers. Paired levitated markers `< >` and `</ >` (yellow blocks) are associated with two boundary tokens of potential targets through position sharing, i.e., identical position code (green blocks). The saliency vector (gray blocks) highlights potential targets linked to lexical units using value 1. Part (c) displays the attention matrix, comprising four sub-matrices: token-token matrix A_{11} , token-marker matrix A_{12} , marker-token matrix A_{21} , and marker-marker matrix A_{22} . Dark-colored cell signifies a value of 1, while white cell denotes a value of 0. Part (d) displays the encoder followed by the span representation and classification module.

5.1.1 Potential Targets Generation

TI is the problem of deciding which words or phrases evoke frames in a given sentence. In other semantic role labeling schemes (e.g., PropBank), simple part-of-speech criteria typically distinguish targets from non-targets. However, in FSP, verbs, nouns, adjectives, and even prepositions can evoke frames under certain conditions [10].

To get the potential targets, TFIMA first enumerates all spans up to a specified length $tarLen$ in the given sentence, and then filters out those impossible spans. Specifically, for the spans containing one word, TFIMA only chooses verbs, nouns, prepositions, adverbs, adjectives, and conjunctions as potential targets. As for spans containing more than one word, TFIMA only chooses the spans that two boundary-words has direct semantic dependency relations as potential targets. Therefore, suppose n is the sentence length, the number of potential targets is approximately $O(50\% \times tarLen \times n)$. Particularly, when $tarLen = 2$ is true, the number of potential targets is approximately $O(n)$.

5.1.2 Saliency Vector Construction

FrameNet provides valuable insights into the relationships between lexical units and frames, such as (*think.v*, [*Awareness*, *Opinion*, etc.]), where the word “think” is associated with a list of potential frames (e.g., *Awareness*, *Opinion*). Identifying a word or phrase within these mappings suggests that it has the potential as a target capable of evoking frames in a given context. However, as mentioned above, the reality may be quite complicated. The inclusion of a lexical unit in FrameNet does not imply that it can evoke a frame in all sentences. Whether it can evoke a specific frame also depends on the specific context in which the word is used. For instance, semantically limited support predicates like the verb “make” in “make a request” may not evoke frames when coupled with a frame-evoking, syntactically dependent noun (such as *request*). Nevertheless, despite these nuances, leveraging mapping knowledge serves as a valuable indicator to guide the model in identifying the most probable targets within a given sentence, albeit without absolute certainty.

We propose a *word-based saliency vector* inspired by Liu [34] to leverage mapping knowledge. This vector, sized to match the length of the sentence, serves to indicate which words are included in the (lexical unit, frame) mappings. Specifically, given a sentence, we first convert each word into its corresponding lexical unit and then search for them within the mappings. When a word is found in the mappings, the corresponding position in the vector is set to 1; otherwise, it is set to 0. This saliency vector is subsequently integrated into the encoder architecture as an additional input. Nevertheless, the word-based saliency vector has shortcomings in highlighting *phrase targets* and discerning the associations between words or phrases and lexical units. Consider the sentence “I think a few of ...”: “think”, “a”, “few”, and “of” can match lexical units in the mappings, resulting in a saliency vector like [0, 1, 1, 1, 1, ...]. However, the phrase “a few” is also present in the mappings. Thus, the word-based saliency vector lacks clarity in distinguishing between words and phrases.

To overcome these limitations, we further propose the *span-based saliency vector*, augmented with levitated markers, to precisely identify words or phrases associated with lexical units in a given sentence. Each potential target span is affixed with a pair of levitated markers, denoted as $\langle c \rangle$ and $\langle /c \rangle$, via positional alignment. The saliency vector is then constructed to have a size equal to the total number of levitated markers. Specifically, if a potential target span corresponds to a mapping, the saliency values for the paired markers are both set to 1; otherwise, they are set to 0. Illustrated in Figure 2, the saliency values for “think”, “a”, “a few”, “few”, and “of” are all [1, 1], distinctly delineating which word or phrase is associated with lexical units.

5.1.3 Input Representation

The input of TFIMA consists of three parts: token sequence, position sequence and saliency vector. As illustrated in Figure 2, the input sequence includes token sequence (tx) for words in the sentence and levitated-marker sequence (mx) for potential targets, the input positions include token positions (tp) and marker positions (mp), and the input saliency vector includes two parts: a full zero vector (ts) for tokens and a saliency vector (ms) for levitated markers. The levitated-marker sequence (mx) consists of a series of paired levitated-markers, and each of them, comprising the marker $\langle c \rangle$ and $\langle /c \rangle$, is associated with a potential target. The position codes of two markers in each paired levitated-markers share the identical positions of their associated two boundary tokens of the potential target.

Suppose *Tokenize(.)* is the default function of BERT used in generating token sequence tx and its corresponding position code tp for a given sentence, *MarkerGen(.)* is the function generating potential targets and their corresponding mx and mp , and *Saliency(.)* is the function generating ts and ms . The processing details of *MarkerGen(.)* and *Saliency(.)* are described in the above sections. Formally, give a sentence X , the process of input representation consists of two sub-process. One is the preparing sub-process, which can be expressed as: $tx, tp = \text{Tokenize}(X)$

$$\begin{aligned} tx, tp &= \text{Tokenize}(X) \\ mx, mp &= \text{MarkerGen}(X) \\ ts, ms &= \text{Saliency}(X) \end{aligned} \tag{1}$$

The other is the embedding sub-process, which can be expressed as:

$$\text{Emb}(tx, mx, tp, mp, ts, ms) = TE([tx; mx]) + PE([tp; mp]) + SE([ts; ms]) \tag{2}$$

where $[.; .]$ is concatenation operation, $TE(.)$, $PE(.)$, and $SE(.)$ are embedding functions for input sequence, position codes and saliency vector, respectively. The results of embedding subprocess consists of two parts: token embedding and marker embedding, as shown in top part (a) and bottom part (b) of Figure 2 respectively.

5.1.4 Marker based Attention

To promote the interaction between markers, tokens, and saliency information, *our proposed TFIMA reformulates the traditional bidirectional attention of the transformer into a hybrid attention through reformulating the attention matrix.*

As shown in Figure 2(c), A_{11} is a full ones matrix, indicating bidirectional attention between tokens, while A_{12} is a full zero matrix, signaling no attention from tokens to markers. A_{21} enforces *directional attention* from levitated markers to their corresponding tokens, enabling the integration of token information into markers carrying saliency details. For instance, cells (19, 1) and (20, 2) in matrix A_{21} are filled with a value of 1, indicating attention from marker $\langle c \rangle$ to the 1st token, i.e., *I*, and attention from marker $\langle /c \rangle$ to the 2nd token, i.e., *think*. We assume each pair of markers operates independently of the others, allowing a marker to attend to its partner within the pair. Thus, in A_{22} , each 2×2 matrix along the main diagonal is populated with a value of 1, while all other cells contain a value of 0. Formally, the attention matrix A is composed of the above-mentioned four submatrices, which can be expressed as:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \quad (3)$$

Throughout the modeling process, the attention matrix A plays a pivotal role in conducting mixed attention modeling on sequences. Thereby generating contextually rich semantic representations. This modeling process is formally represented as follows:

$$H = \text{Encoder}(\text{Emb}(tx, mx, tp, mp, ts, ms), A) \quad (4)$$

where $H \in \mathbb{R}^{(n+2m) \times d}$ denotes the output representation of the encoder, comprising two distinct components: $H^{(a)} \in \mathbb{R}^{n \times d}$ captures the token sequence's representation, and $H^{(b)} \in \mathbb{R}^{2m \times d}$ encodes the marker sequence's representation. The function $\text{Encoder}(\cdot)$ refers to the context encoder, exemplified by BERT. BERT is a Transformer based model, and it uses multi-head attention mechanism from the Transformer, which allows the model to attend to different parts of the input when computing representations for a given token. The attention matrix A is used to constrain the scope and direction of attention in multi-head attention mechanism of Transformer. Specifically, suppose Q , K and V are query matrix, key matrix and value matrix of one attention head, the output of this head can be expressed as:

$$\text{attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} + \tilde{A}\right)V \quad (5)$$

where $\tilde{A}$ is a mask matrix converted from matrix A , and d_k is the dimensionality of the key. The values 0 and 1 in matrix A are converted into 0 and -inf (i.e. negative infinity), respectively, thereby forming the mask matrix $\tilde{A}$. After applying the softmax function, the corresponding value at the -inf position of the output matrix of softmax function becomes 0, indicating that the corresponding attention weight is 0. When the output matrix is multiplied by V , each row of the output matrix is multiplied with each column

of V , by which the result of the multiplication aggregates the information of columns of V with different weights contained in rows of $\tilde{A}$. Value 0 in a row of $\tilde{A}$ signifies the result of the multiplication does not aggregate the information of corresponding column in V .

5.1.5 Span Representation and Classification

Given the contextual representation H , for any potential target s_{ij} with start index i and end index j in the token sequence, along with its associated paired markers indexed as $m(i)$ and $m(j)$ in the marker sequence, the representation of the potential target can be expressed as:

$$r(s_{ij}) = [h_i^{(a)}; h_j^{(a)}; h_{m(i)}^{(b)}; h_{m(j)}^{(b)}] \quad (6)$$

where $h_i^{(a)}, h_j^{(a)} \in H^{(a)}$ and $h_{m(i)}^{(b)}, h_{m(j)}^{(b)} \in H^{(b)}$. Finally, $r(s_{ij})$ is fed into a *Linear* classifier, which can be represented as:

$$p(s_{ij}) = \text{Linear}(r(s_{ij})) \quad (7)$$

5.2 Second stage: Frame-semantic Role Labeling Based on Marker Attention

The existing two-stage FSRL methods, involving argument identification (AI) and argument classification (AC), often result in the issue of error propagation. To address this issue, we adopt a one-stage method that identifies arguments and their roles in a single run, and propose a graphbased method to prune impossible spans in advance so as to ensure the running efficiency of the model. Moreover, representing spans poses a fundamental challenge. The conventional boundary-based span representation faces limitations in effectively capturing complete semantic information of a given span, particularly for longer spans. Thus, we propose a novel one-stage FSRL model based on marker attention (named FSRLMA), which leverages levitated markers to enhance span representations by capturing internal information. Marker-based representation and boundary-based representation provide different views for span representation, achieving complementary information. The combination of them is beneficial to forming a complete and semantic rich representation for spans.

5.2.1 Potential Argument Generation

FSRLMA operates as a span-based method, necessitating the enumeration of all potential spans to identify arguments and their corresponding roles. However, for a text with n words, the sheer number of possible spans can escalate to $O(n^2)$, resulting in considerable computational overhead. To tackle this challenge, we design two pre-processing strategies: *window-based span generation* [18] and *graph-based span pruning*. Unlike conventional methods for argument identification, our approach is not solely focused on precisely identifying arguments. Instead, it aims to selectively prune the least relevant spans while retaining as many genuine arguments as possible.

Given arguments typically cluster around the target, window-based span generation confines the enumerated spans within a specific window around the target word. Let the window size be $winSize$ ($winSize \leq n$), and the maximum span length be $maxLen$ ($maxLen < winSize$). Employing the window-based strategy ensures the number of generated spans is, at most, $O(winSize \times maxLen)$.

A potential argument should convey complete semantics, such as a concept, event, or entity. Thus, the corresponding sub-graph derived from the dependency graph of the sentence must either satisfy (1) a weakly connected graph, as depicted in Figure 3 (a), (b), and (c); or (2) disconnected sub-graphs sharing a common parent node, exemplified in Figure 3 (d). Otherwise, the graph should be pruned. Following the pruning process, the number of spans can be reduced by 50%, resulting in the maximum number of *potential arguments* being at most $O(winSize \times maxLen/2)$.

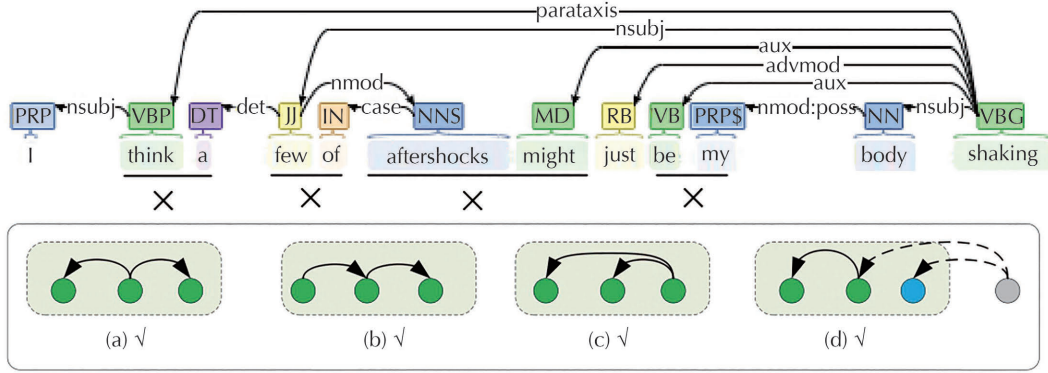


Figure 3. Graph-based potential arguments pruning.

This pre-processing method offers two advantages: firstly, it enhances the operational efficiency of the FSRLMA model; secondly, it mitigates bias concerns stemming from an excessive number of irrelevant spans by reducing their quantity.

5.2.2 Input and Encoder Representation

FSRLMA adopts a similar architecture to TFIMA, albeit with some differences in the input representation. The diagram of FSRLMA is illustrated in Figure 4.

Firstly, the saliency vector is omitted. Secondly, [fname] and [/T] markers are inserted into token sequence tx to highlight the target and incorporate frame information into the encoder. The incorporated frame information can be regarded as a kind of “soft constraint”, enforcing the model to predict frame-specific roles for arguments. Thirdly, FSRLMA associates each potential argument to a single levitated-marker [c] in marker sequence mx , instead of paired levitated-markers. The aim of each single levitatedmarker is to capture the internal information of the corresponding potential argument through directional attention.

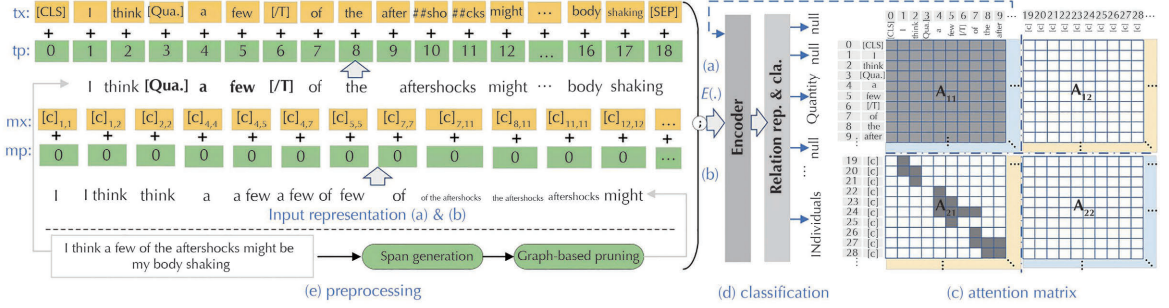


Figure 4. Diagram of FSRLMA. The input tokens include token sequence (tx) and marker sequence (mx), while the input positions include token positions (tp) and marker positions (mp). $[Qua.]$ (abbreviation for [Quantity]) and $[T]$ are markers to highlight frame evoked by target “a few” and separate target from other words. $[c]_{i,j}$ denotes the sigle marker $[c]$ attending the potential argument span (i, j) . Part (c) illustrates the attention matrix, and part (d) shows the encoder followed by the relation representation module and classification module. The classification outputs: $null$, $Quantity$, and $Individuals$ are role labels for potential arguments. Part (e) displays two pre-processing modules.

In addition, the attention matrix A_{22} is transformed into a complete zero matrix, signifying no attention between markers, while the values of A_{21} are adjusted using the following function:

$$a_{ij} = \begin{cases} 1 & b \leq j \leq e \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

where (b, e) denotes the i th potential argument, and $a_{ij} = 0$ shows no attention exists from the i th marker to its associated potential arguments (b, e) .

Suppose $Tokenize(.)$ is the default function of BERT used in generating token sequence tx and its corresponding position code tp for a given sentence, $SmarkerGen(.)$ is the function generating potential targets and their corresponding mx and mp . The processing details of $SmarkerGen(.)$ are described in the above sections. Formally, given a sentence X , the process of input representation consists of two sub-process. One is the preparing sub-process, which can be expressed as:

$$\begin{aligned} tx, tp &= Tokenize(X) \\ mx, mp &= SmarkerGen(X) \end{aligned} \quad (9)$$

The other is the embedding sub-process, which can be expressed as:

$$Emb(tx, mx, tp, mp) = TE([tx; mx]) + PE([tp; mp]) \quad (10)$$

where $[.;.]$ is the concatenation operation, $TE(.)$ and $PE(.)$ are embedding functions for the input sequence and position codes, respectively.

After obtaining the input representations, they are fed into a contextual encoder to get contextually rich semantic representations. This modeling process is formally represented as:

$$H = \text{Encoder}(\text{Emb}(tx, mx, tp, mp), A) \quad (11)$$

and $H \in \mathbb{R}^{((n+2)+m) \times d}$ consists of $H^{(a)} \in \mathbb{R}^{(n+2) \times d}$ and $H^{(b)} \in \mathbb{R}^{m \times d}$ is the encoder output. The function $\text{Encoder}(\cdot)$ refers to the context encoder, exemplified by BERT.

5.2.3 Role Classification

We view frame-semantic roles of each frame as a group of semantic relationships between the target and its arguments. Given the contextual representation H , target t_{ij} and its evoked frame f_{ij} , for any potential argument s_{be} , the relation representation is defined as: $r(t_{ij}, s_{be}) = [h_b^{(a)}; h_e^{(a)}; \phi(t_{ij}, s_{be}); h_{m(b,e)}^{(a)}; h_{j-1}^{(a)}; h_{j+1}^{(a)}]$, where $h_{i-1}^{(a)}, h_{j+1}^{(a)}, h_b^{(a)}, h_e^{(a)} \in H^{(a)}$ are boundary representations for target t_{ij} and potential arguments s_{be} , while $h_{m(b,e)}^{(b)} \in H^{(b)}$ is the levitated marker representation associated with s_{be} , where $m(b, e)$ is its index in the marker sequence. For span (1, 1) and (1, 2) in Figure 4, their corresponding marker index are $m(1, 1) = 1$ and $m(1, 2) = 2$ respectively. Building on prior research [18], $\phi(t_{ij}, s_{be})$ is the concatenated embeddings for the relative distance (from t_{ij} and s_{be}) and the width of s_{be} .

Finally, $r(t_{ij}, s_{be})$ is fed into a *Linear* classifier for relation classification, which is represented as:

$$p_r(t_{ij}, s_{be} | f_{ij}) = \text{Linear}(r(t_{ij}, s_{be})) \quad (12)$$

5.3 Training and Prediction

We train two separate models using cross-entropy loss. Given a sentence, let S represent its potential target (or argument) set. The golden label for span $s_{ij} \in S$ is denoted as $y_{ij} \in FU\{null\}$ (or $y_{ij} \in RU\{null\}$), where *null* indicates that the potential target (argument) is not an actual target (argument). The cross-entropy loss is calculated on S in training phase.

For TFIMA model, which jointly predicts targets and their evoked frames, the training loss can be represented as follows:

$$\mathcal{L}_r = - \sum_{s_{ij} \in S} y_{ij} \log(p_r(s_{ij})) \quad (13)$$

During prediction, we employ a *frame filtering* strategy similar to previous studies [18, 20]. Let $\tilde{\mathbf{F}} = FU\{null\}$ denote the frame classes. Then, we construct a masking vector $v_{s_{ij}} \in [0, -\infty^{1 \times |\tilde{\mathbf{F}}|}]$ based on the association between the potential target s_{ij} and the lexical knowledge base. Here, $|\tilde{\mathbf{F}}|$ represents the number of elements in $\tilde{\mathbf{F}}$. Each element in the vector is defined as $v_{s_{ij}}[0] = 0$, signifying that the *null* frame could be triggered by any potential target when no suitable frame is identified. For the k -th element ($k \geq 1$) in $v_{s_{ij}}$, its value is determined as follows:

$$v_{s_{ij}}[i] = \begin{cases} 0 & \text{if } \tilde{\mathbf{F}}[k] \in F_{lex(s_{ij})} \\ -\infty & \text{otherwise} \end{cases} \quad (14)$$

where $lex(s_{ij})$ denotes the lemmatized form of the potential target, and $\mathbf{F}_{lex(s_{ij})} \subset \mathbf{F}$ represents the set of frames in the (lexical unit, frames) mappings corresponding to $lex(s_{ij})$. Ultimately, the masking vector is utilized for filtering during prediction, allowing the prediction function to be denoted as follows:

$$\hat{f} = \arg\max_{f \in \tilde{\mathbf{F}}} (\text{softmax}(p_f(s_{ij}) + v_{s_{ij}})) \quad (15)$$

For FSRLMA model which jointly predicts arguments and their corresponding roles, the training loss can be represented as follows:

$$\mathcal{L}_r = - \sum_{s_{be} \in S} y_{be} \log(p_r(t_{ij}, s_{be} | f_{t_{ij}})) \quad (16)$$

During role prediction, in line with previous research [18, 20], we enforce a “hard constraint” on the predicted roles to belong to the current frame through *role filtering*. Specifically, for each frame f , a mask vector $v_f \in 0, -\infty^{1 \times |\tilde{\mathbf{R}}|}$ is constructed based on the semantic relationship between the frame and roles. Here, $v_f[0] = 0$ indicates that each potential argument can possibly be predicted to have a *null* role. For the i -th element ($i \geq 1$) in v_f , its value is defined as:

$$v_f[i] = \begin{cases} 0 & \tilde{\mathbf{R}}[i] \in R_f \\ -\infty & \text{otherwise.} \end{cases} \quad (17)$$

Subsequently, the predicted roles undergo filtering using the mask to confine them within the current frame. For a given target word t_{ij} and its associated framework $f_{t_{ij}}$, the role of the potential argument s_{be} can be predicted using the following function:

$$\hat{r}_{s_{be}} = \arg\max_{r \in \tilde{\mathbf{R}}_f} (\text{softmax}(p_r(t_{ij}, s_{be} | f_{t_{ij}}) + v_{f_{t_{ij}}})) \quad (18)$$

6. EXPERIMENTS AND ANALYSIS

6.1 Datasets and Experimental Setup

We evaluate our models using two benchmark datasets: FrameNet annotation datasets version 1.5 and 1.7, denoted as FN1.5 and FN1.7 respectively. FN1.5 is a widely utilized full-text annotation dataset in previous research, while FN1.7 [35] is the latest version, incorporating a broader range of frames and roles. We follow the same train/development/test split as previous studies [13, 16-17, 20]. Table 2 presents the statistics of the two datasets. The first line of each block displays the total number of sentences in the train, development, and test datasets, while the second and third lines respectively show the total number of targets and arguments in each block.

Table 2. Statistics of the datasets.

Dataset	Type	Train	Development	Test
FN1.5	#sentence	2,713	326	982
	#target	16,621	2,284	4,427
	#argument	29,426	4,044	7,148
FN1.7	#sentence	3,413	326	1,354
	#target	19,391	2,272	6,714
	#argument	34,219	4,029	11,302

We utilize the Python package Stanza [36]^① to obtain the dependency graph of the sentences. Stanza, developed by the natural language processing team at Stanford University, provides robust tools for linguistic analysis. Additionally, for lemmatization and part-of-speech tagging, we employ the NLTK^② (Natural Language Toolkit), developed by Steven Bird and Edward Loper at the Department of Computer and Information Science, University of Pennsylvania, offering comprehensive resources for natural language processing tasks.

We choose the BERT-base-uncased architecture as the backbone of our context encoder, finetuning all parameters during training. Our models are trained using GPU A100 and implemented with PyTorch. Hyperparameters are tuned based on the F1 metric on the development dataset. Further details about the parameter settings are illustrated in Tables 3 and 4.

6.2 Overall Comparison Results

In this section, we compare the performance of end-to-end FSP models. Here, end-to-end model refers to the model that can predict the complete frame-semantic structures given a sentence.

To the best of our knowledge, **FParser** [20] is currently the only publicly available end-to-end FSP model. It models FSP as a process of graph construction problem, in which targets and arguments are nodes, frames are node types for targets, and roles are edges between target and argument nodes. TaRFSP [18] is a SOTA span-based model that jointly performs FI and FSRL when the golden target is given in a sentence. We adapt it into an end-to-end FSP model, called **TaRFSP(Joint)**, in which the potential targets generating method is the same as TFIMA, and the model needs to enumerate all the potential targets to predict each complete frame-semantic structure from a given sentence. Furthermore, to deeply analyze our proposed two-stage model (TFIMA + FSRLMA) for FSP, we also design two variants: **TFIMA(no know.) + FSRLMA** and **TFI & FSRLMA(Joint)**. TFIMA (no know.) signifies TFIMA excludes the mapping knowledge by removing the saliency vector from the model. TFI & FSRLMA (Joint) is an end-to-end FSP model based

^① <http://nlp.stanford.edu/software/stanza/1.0.0/en/default.zip>.

^② <https://www.nltk.org/>.

Table 3. Hyper-parameters for TFIMA.

Parameters	Values
Maximum length of potential target <i>tarLen</i>	2
Maximum length of token sequence (tx)	160
Maximum length of marker sequence (mx)	320
Training epochs	20
Batch size	16
Learning rate	5e-5
Optimizer	AdamW
Warmup proportion	0.1

Table 4. Hyper-parameters for FSLRMA.

Parameters	Values
Maximum length of potential argument <i>maxLen</i>	20
Window size for argument generation <i>winSize</i>	40
Maximum length of token sequence (tx)	160
Maximum length of marker sequence (mx)	100
Training epoch	20
Batch size	6
Learning rate	2e-5
Optimizer	AdamW
Warmup proportion	0.1

on FSRLMA, wherein the marker [fname] and [/T], highlighting the target, are removed from the token sequence as the frame is unknown in advance for each potential target in joint model, and word-based saliency vector is used to incorporating (lexical unit, frame) mapping knowledge instead of span-based saliency vector since it is impossible to simultaneously append single markers and paired levitated-markers into the sequence. All the comparison models and their performances on three subtasks of FSP are illustrated in Table 5.

Table 5. The performance of different end-to-end FSP models in all of the three subtasks. * indicates that our first-order model achieves significantly stronger F1-score than FParser and TaRFSP(Joint) with $p < 0.05$ on Wilcoxon test.

Dataset	Model	TI			FI			FSRL		
		P	R	F1	P	R	F1	P	R	F1
FN1.5	FParser (pipeline) [20]	73.17	75.47	74.30	66.08	68.15	67.06	45.91	46.70	46.30
	FParser [20]	75.81	76.17	75.99	68.72	69.05	68.89	47.79	50.60	49.16
	TaRFSP (Joint) [18]	73.49	76.18	74.81	64.70	67.00	65.82	52.03	51.20	51.61
	Ours (TFIMA (<i>no know.</i>) + FSRLMA)	76.50	76.51	76.50	69.59	69.60	69.59	53.51	55.99	54.72
	Ours (TFI & FSRLMA (Joint))	74.58	82.60	78.38	67.59	73.55	70.40	49.91	56.76	53.11
	Ours (TFIMA + FSRLMA)	78.37	78.72	78.55*	70.12	72.83	71.45*	53.74	57.46	55.53*
FN1.7	FParser (pipeline) [20]	78.62	69.92	74.02	71.05	63.06	66.82	49.32	45.60	47.38
	FParser [20]	76.16	74.98	75.56	69.39	68.30	68.84	49.09	48.81	48.95
	TaRFSP (Joint) [18]	75.12	75.18	75.15	65.70	65.81	65.75	53.13	51.30	52.20
	Ours (TFIMA (<i>no know.</i>) + FSRLMA)	75.76	79.64	77.65	69.33	72.88	71.06	54.58	55.44	55.01
	Ours (TFI & FSRLMA (Joint))	78.15	78.82	78.49	70.14	71.13	70.64	53.85	54.31	54.02
	Ours (TFIMA + FSRLMA)	78.50	79.89	79.19*	69.34	74.62	71.88*	55.52	56.24	55.88*

Table 5 shows that our proposed two-stage model TFIMA + FSRLMA achieves significant superiority over the current SOTA end-to-end FSP model FParser across all three subtasks on two datasets. Since FParser does not use (lexical units, frame) mapping knowledge during its encoding phase, we compare it with TFIMA (no know.) + FSRLMA to ensure a fairer comparison. Remarkably, even in this adjusted scenario, our model still outperforms FParser by a substantial margin.

Through deep analysis, we have found that FParser fails to recognize overlapping spans, resulting in span missing issue. Specifically, FParser uses a classification module to classify each possible span into one of the three kinds of nodes, i.e., target node, argument node, or null node. In other words, given a possible span, it can only be classified into one of the three kinds of nodes. When a span acts both a target or an argument, it will be missing in the argument set or target set. For instance, in the sentence “I think a few of the aftershocks might just be my body shaking”, the phrase “a few” is a target that evokes frame Quantity, and it is also an argument that acts as the role Quantity. If FParser recognizes it as a target, it

means an argument is missing; conversely, if it is recognized as an argument, it means a target is missing. TaRFSP (Joint) uses two separate classification modules to identify targets and arguments, so it can address span missing problem. As shown in Table 5, TaRFSP (Joint) achieves substantial improvements over FParser in FSRL subtask, outperforming FParser by absolute values of 2.54 and 3.25, respectively, in terms of F1 metric across two datasets. The improvements can be mainly attributed to the elimination of span missing problem as the overlapping of arguments and targets is common in FSP.

Compared to FParser, our two-stage model TFIMA+FSRLMA overcomes the issue of span missing by identifying targets in the first stage and arguments in the second stage, and achieves substantial improvements in all of the three subtasks in terms of the F1 metric, outperforming FParser across two datasets by absolute values of 2.56 and 3.63 on TI, 2.56 and 3.04 on FI, 6.73 and 6.93 on FSRL, respectively. Compared with TaRFSP (Joint), our TFIMA+FSRLMA model still has a significant advantage on all three subtasks. Specifically, the average improvements across two datasets on TI, FI and FSRL in terms of the F1 metric are 3.89, 5.88 and 3.8 respectively. As shown in Table 5, our two variant models (TFI&FSRLMA (Joint) and TFIMA (no know.) + FSRLMA) are also substantially superior to TaRFSP (Joint) in terms of the F1 metric. The results suggest that our proposed method has a substantial advantage in addressing the end-to-end FSP task. The results also reveal that two joint models fall short compared to the twostage models. This suggests that sharing an encoder might not be conducive for the end-to-end FSP task, which can be attributed to the following reasons. Firstly, given that TI and FI entail span-level classification tasks, while FSRL involves two-span relation classification. Secondly, the incorporation of (lexical unit, frame) mapping knowledge with the word-based saliency vector is less effective than the span-based saliency vector through marker-based attention. Thirdly, the joint models are not convenient to explicitly model the dependency relation between frame and its roles.

Overall, these results underscore the suitability of our two-stage method for addressing the end-to-end FSP task, and highlight the overall advantage of our approach. Further insights into the specific contributions of each module will be elaborated in the subsequent section.

6.3 Ablation Studies

To further analyze the impact of each component of our two-stage model on the performance of FSP, we conduct ablation experiments on two models across two datasets.

6.3.1 Ablation Studies on TFIMA

Table 6 illustrates that all three modules are useful in improving the performance of TFIMA, since the overall performance has dropped substantially when one of them is removed, and the model achieves the worst performance in terms of F1 when all of the three modules (i.e., directional attention, saliency embedding, and paired markers) are omitted from TFIMA. Specifically, the F1 metric on TI decreased from 78.55 to 75.75 on the FN1.5 dataset, and from 79.19 to 75.73 on the FN1.7 dataset. The F1 metric on FI decreased from 71.45 to 69.13 on the FN1.5 dataset, and from 71.88 to 68.60 on the FN1.7 dataset.

Table 6. Ablation results for TFIMA.

Dataset	Model	TI			FI		
		P	R	F1	P	R	F1
FN1.5	TFIMA	78.38	78.72	78.55	70.12	72.83	71.45
	w/o directional attention	77.34	77.71	77.53	70.65	70.98	70.81
	w/o saliency embedding	76.50	76.51	76.50	69.59	69.60	69.59
	w/o paired markers	78.30	74.23	76.21	73.60	66.87	70.07
	w/o all of the three	77.12	74.44	75.75	69.87	68.41	69.13
FN1.7	TFIMA	78.50	79.89	79.19	69.34	74.62	71.88
	w/o directional attention	78.89	77.42	78.15	72.23	70.88	71.55
	w/o saliency embedding	76.94	76.08	76.51	69.89	69.11	69.50
	w/o paired markers	77.88	76.82	77.35	70.66	69.71	70.18
	w/o all of the three	77.64	73.91	75.73	70.33	66.95	68.60

TFIMA (w/o directional attention) refers to each paired levitated-marker sharing the same position code with their corresponding boundary words of the potential target, without using the attention from markers to words, namely the attention sub-matrix A_{21} being a full zero matrix. The ablation results show that binding levitated markers to the potential targets through direction attention is necessary, which is beneficial to promote the interaction between targets and their corresponding markers. When directional attention is removed, the performance of TI and FI has dropped substantially, as shown in Table 6. Specifically, the F1 metric on TI decreased from 78.55 to 77.53 on FN1.5 dataset, and from 79.19 to 78.15 on FN1.7 dataset. The F1 metric on FI decreased from 71.45 to 70.81 on FN1.5 dataset, and from 71.88 to 71.55 on FN1.7 dataset.

TFIMA (w/o saliency embedding) refers to the removal of the saliency vector from the input of TFIMA. The ablation results show that saliency embedding plays a significant role in improving model's performance, as the performance of TI and FI has dropped substantially when it is removed, as shown in Table 6. Specifically, the F1 metric on TI decreased from 78.55 to 76.50 on the FN1.5 dataset, and from 79.19 to 76.51 on the FN1.7 dataset. The F1 metric on FI decreased from 71.45 to 69.59 on the FN1.5 dataset, and from 71.88 to 69.50 on the FN1.7 dataset.

Note that TFIMA (w/o paired markers) replaces the span-based saliency vector with the word-based saliency vector and attaches the saliency embedding directly to the token sequence embedding, while removing the marker sequence from the TFIMA model input. This adjustment leads to a significant decrease in TI and FI performance, particularly in Recall. As shown in Table 6, the Recall (R) metric on TI decreased by 4.49 and 3.07 on two datasets respectively, while the Recall (R) metric on FI decreased

by 5.96 and 4.91 on two datasets respectively. This suggests that the span-based saliency vector is more effective in identifying additional targets compared to the word-based saliency vector, emphasizing the importance of incorporating saliency information with levitated markers.

These findings support our assumption that incorporating (lexical unit, frame) mapping knowledge yields benefits, as this information provides useful background knowledge for target identification. These findings also reveal that our proposed knowledge incorporating method based on the paired levitated-markers and span-based saliency vector is more effective than the method that directly incorporate mapping knowledge into contextual encoder using word-based saliency vector.

6.3.2 Ablation Studies on FSRLMA

Table 7 illustrates the importance of both frame markers and levitated markers in enhancing the performance of FSRLMA. Removing either of these components results in a significant decrease in overall performance, with the model achieving its lowest F1 score when both are omitted from FSRLMA. Specifically, the F1 metric decreased from 75.00 to 74.41 on the FN1.5 dataset, and from 75.38 to 74.42 on the FN1.7 dataset.

Table 7. Ablation results for FSRLMA.

Dataset	Model	P	R	F1
FN1.5	FSRLMA	74.00	76.02	75.00
	w/o frame markers	72.45	75.61	73.99
	w/o levitated markers	72.61	75.39	73.97
	w/o both of them	72.30	74.56	73.41
	w/o graph-based pruning	73.64	75.10	74.36
FN1.7	FSRLMA	76.25	74.52	75.38
	w/o frame markers	74.59	74.59	74.59
	w/o levitated markers	75.33	74.13	74.73
	w/o both of them	75.17	73.68	74.42
	w/o graph-based pruning	76.18	73.35	74.74

FSRLMA (w/o frame markers) refers to the removal of the markers inserted into the token sequence tx . As shown in Table 7, the F1 metric dropped by 1.01 and 0.79 on two datasets respectively, when the frame markers are removed from FSRLMA. The ablation results support our assumption that incorporating frame information with markers yields benefits, as this information provides background knowledge for argument identification and role classification, thereby constraining predicted roles within the current frame.

FSRLMA (w/o levitated markers) refers to the removal of the marker sequence mx from the input sequence. As shown in Table 7, the F1 metric dropped by 1.03 and 0.65 on two datasets respectively, when the levitated markers are removed from FSRLMA. The ablation results demonstrate the effectiveness of utilizing levitated markers to enhance the representation of argument spans. The traditional boundary-based span representation method only uses the information from the boundary word of each span. It is effective for representing shorter spans. For some longer argument spans, it is necessary to capture internal information of spans to enhance their representations. Therefore, our proposed marker-based method provides a new approach to span representation.

The proposed graph-based pruning method serves a dual purpose: it enhances running efficiency by reducing potential argument spans and contributes to the improvement in FSRLMA's performance. This is evident from Table 7, where FSRLMA's overall performance notably declines when the graph-based pruning module is removed. Specifically, the F1 metric decreased from 75.00 to 74.36 on the FN1.5 dataset, and from 75.38 to 74.74 on the FN1.7 dataset. The observed performance boost can be attributed to the significant reduction in irrelevant spans—approximately 50%—which helps the model mitigate bias against false arguments.

6.4 Individual Task Comparison Results

To test the scalability and adaptability of our proposed method, we conduct comparison experiments with currently existing individual TI, FI and FSRL models. For fair comparison, the experiments only choose models adopting BERT as the encoder.

6.4.1 TI and FI subtasks

As depicted in Table 8, the first block shows the dedicated TI models, while the second shows the joint TI and FI models. FParser (target & frame) and FParser (target) are two variants of the end-to-end FSP model FParser [20], and Emanuele (target) is a BIO-label based sequence labeling model for TI proposed by Emanuele et al. [15]. The results show that both TFIMA (target) (a variant of TFIMA designed for TI, converting TFIMA into a binary classification model) and TFIMA outperform the two baselines (Emanuele

Table 8. TI performance in terms of F1.

Model	FN1.5	FN1.7
Emanuele (target) [15]	74.96	-
FParser (target) [20]	76.09	75.34
TFIMA (target)	78.13	78.18
FParser (target & frame) [20]	76.47	75.88
TFIMA	78.55	79.19

and FParser (target)) by a considerable margin, underscoring the effectiveness of our approach. Moreover, the results reveal that TFIMA achieves superior performance compared to TFIMA (target), while FParser (target & frame) also achieves superior performance compared to FParser (target), suggesting that jointly modeling TI and FI is beneficial for improving the performance of TI. The findings reveal that jointly modeling TI and FI subtasks in our proposed framework is reasonable.

As illustrated in Table 9, all the models in the first block are span-based classification model for FI. The differences among these span-based models mainly lie in the different approaches for representing spans. TFIMA (frame) is a variant of TFIMA for FI, which is trained in the condition that the golden targets are given. It surpasses all the span-based baselines (shown in the first block) in terms of Accuracy, without employing frame knowledge, and slightly trails behind the dedicated FI models (shown in the second block) using frame knowledge, such as frame definition and frame relations. The results not only reveal the superiority of TFIMA (frame) among span-based models but also the importance of frame knowledge in FI. Therefore, how to incorporate frame knowledge into our marker-based method will be our further work.

Table 9. FI accuracy with golden target.

Model	FN1.5	FN1.7
PAFIBERT [28]	91.65	91.11
Emanuele [15]	89.90	-
Chen [17]	90.50	90.10
Fparser (frame) [20]	90.62	90.64
TaRFSP (frame) [18]	91.40	91.30
TFIMA (frame)	91.89	91.78
FIDO [29]	91.30	92.10
KGFI [30]	92.13	92.40
COFFTEA [31]	92.55	92.64

6.4.2 FSRL subtask

As depicted in Table 10, Kalyanpur is a generative model for FSRL proposed by Kalyanpur et al. [37], which adopts an encoder-decoder architecture to generate arguments and roles for a given sentence when the target and frame are provided. Emanuele (role) is a FSRL model proposed by Emanuele et al. [15], comprising an argument identification (AI) module and an argument classification (AC) module. The AI is responsible for arguments identification, while the AC module is used to predict their roles. FParser (role) [20] is a variant of FParser, which first identifies the argument nodes and then predicts the relations between argument nodes and targets nodes as the roles for arguments. FParser (role) adopts LSTM to

represent the potential argument spans. TaRFSP (role) is a variant of joint FI and FSRL model TaRFSP [18], which utilizes a Biaffine network to identify argument spans and the predict the roles for these arguments. TaRFSP adopts a target-aware attention network to represent argument spans. Except for Kalyanpur, all the other three FSRL models adopt two-stage (i.e., argument identification and role classification.) approach to perform argument identification and role classification sequentially.

Our FSRLMA model consistently outperforms the above-mentioned models by large margins across two dataset, as shown in Table 10. This improvement primarily stems from two key factors: marker-enhanced span representation and one-stage role classification. Specifically, FParser employs LSTM to augment boundary-based span representations. TaRFSP (role) uses a target-aware attention network to augment boundary-based span representations. The experimental results demonstrate the superiority of marker-based methods in enhancing the boundarybased span representations, since it is more effective to capture the internal information of spans than LSTM-basd and target-aware based methods. Furthermore, FSRLMA directly conducts role classification on potential arguments. This streamlined approach helps mitigate the error propagation problem inherent in two-stage FSRL systems like FParser.

Table 10. FSRL performance with golden targets and frames.

Model	FN1.5			FN1.7		
	P	R	F1	P	R	F1
Kalyanpur [37]	-	-	-	71.00	73.00	72.00
Emanuele (role) [15]	71.19	74.26	72.69	-	-	-
FParser (role) [20]	-	-	73.28	-	-	72.06
TaRFSP (role) [18]	-	-	73.89	-	-	73.16
AGED [32]	71.93	76.78	74.28	74.02	75.46	74.73
FSRLMA	74.00	76.02	75.00	76.25	74.52	75.38

Our FSRLMA model demonstrates superior performance compared to AGED in terms of Precision and F1. The higher Precision of FSRLMA indicates its stronger capabilities in semantic representations, enabling more precise distinction of roles for arguments. AGED utilizes (sentence, frame definition) pairs as input for BERT-based encoder in order to promote interactions between role labels within frame definition and words in the sentence, so as to obtain semantic-rich role representations. It then utilizes role representations as query terms to retrieve corresponding arguments from the sentence. Note that both FSRLMA and AGED adopt onestage method, and their performances are superior to two-stage models, such as FParser (role) and TaRFSP (role). The findings support our assumption that simultaneous argument and role identification yields benefits for FSRL.

The individual task experiments demonstrate our proposed framework for FSP has strong scalability and adaptability.

6.5 Discussion

6.5.1 Running Efficiency Analysis

A previous study [38] demonstrates that the computation of Pre-trained Language Models (PLMs) such as BERT rises almost linearly with the increase in sequence length. However, as the sequence length continues to grow, the computation escalates quadratically due to the self-attention module [33]. It is also discussed in [39]. In other words, if the increased number of levitated markers remains within a reasonable range, the increased computation is limited.

To analyze the effects of marker length on computation. We choose the TFIMA model to conduct running efficiency experiment for quantitative analysis. For TFIMA, the optimal value for the maximum potential targets $tarLen$ is set to 2, so the number of potential targets is approximately $O(n)$ given a sentence with the length of n . Therefore, the length of the input sequence of TFIMA model is increased by $O(2n)$ compared to that of the TFIMA (w/o marker). The results (see Table 11) show that the runtime of TFIMA increases by 6.4% compared to TFIMA (w/o marker). The increased running time caused by markers insertion. The results show that the increased running time is limited and acceptable.

Table 11. Efficiency analysis of TFIMA on the FN1.5 test dataset with varying marker lengths. ‘sec.’ denotes ‘second’ and ‘sen./sec.’ refers to ‘sentences per second’.

Model	Sequence length	Time (sec.)	Speed (sen./sec.)
TFIMA (w/o markers)	$O(n)$	10.22	97.88
TFIMA	$O(3n)$	10.87 (↑6.4%)	91.93

As for FSRLMA, we employ window-based span generation and graph-based pruning to reduce the potential argument spans in advance, ensuring that the maximum number of potential arguments is at most $O(winSize \times maxLen/2)$. Additionally, we utilize a single marker to attend to the potential target spans in the sentence, resulting in the marker sequence length being equivalent to the number of potential argument spans. To be specific, when $winSize$ and $maxLen$ are set to 40 and 20 respectively, the average number of potential arguments for each target is approximately 100, while the average length of the sentence is approximately 50. In other words, the average length of marker sequence is 2 times of that of token sequence, which is similar to that of TFIMA. Therefore, we can roughly conclude that the increased runtime of the model FSRLMA is also limited.

Indeed, several factors can influence the running efficiency of the model beyond sequence length, including network architecture, running settings, and etc. Our proposed framework employs BERT as the encoder followed by a one-layer linear network as the classifier, without any additional complexity

modules, thus ensuring a concise network architecture. Furthermore, our proposed framework is applicable to all three subtasks of FSP.

In conclusion, although direct comparisons of running efficiency with other models may not be feasible, based on the above analysis, we can confidently assert that our proposed model achieves substantial performance improvements while incurring only minor increases in overall complexity and running efficiency compared to previous models.

6.5.2 Analysis of Performance of TFIMA on out-of-domain Datasets

To further investigate the performance of TFIMA in out-of-domain datasets, we use MASC [40-41] as an additional experimental dataset. The annotated texts in MASC dataset involves five sources, i.e., newspaper, letter, email, travel-guide & journal, and the statistics of MASC are illustrated in Table 12. We regard each source as one domain to evaluate the TFIMA's performance on out-of-domain datasets. Specifically, we train a model based on four of the five domains, and test the performance on the other domain. To effectively demonstrate the performance of our proposed methods on out-of-domain dataset, we train our proposed TFIMA model and its baseline version in order to compare their performance differences. The baseline version of TFIMA, denoted as TFIMA (base), does not use the following modules: directional attention, saliency embedding, and paired markers. In other words, TFIMA (base) treats joint target and frame identification as a span-based classification task and uses boundary-based method to represent spans without using additional modules, just like FParser.

Table 12. Statistics of MASC dataset involving five domains.

Domains	Newspaper	Letter	Email	Travel-guide	Journal	Total
#sentence	106	71	32	243	62	514
#target	772	399	113	1549	360	3193
#frame	430	229	63	426	182	1330

Table 13 demonstrates the experimental results on out-of-domain datasets. Each block of the table shows the performance of TFIMA and TFIMA (base) in the TI and FI task, and line Δ illustrates the performance differences between two models. There are two groups (TI & FI) of metrics (P, R and F1) in each block. The largest difference and the second largest difference are respectively shown in bold and bold-italics in each group. The results show our proposed TFIMA model based on marker attention has substantial performance advantage than TFIMA (base) on all of the five out-of-domain testings in terms of comprehensive F1 metric. For instance, the F1 metric of TFIMA respectively outperforms TFIMA (base) by 11.27 and 4.98 in terms of TI and FI on Newspaper domain. The results also show the F1 improvements can be mainly attributed to the improvements in the Recall (R) metric, as the differences between TFIMA and TFIMA (base) in Recall attain the largest value in each group. Therefore, we believe that even on

Table 13. Performance comparison between TFIMA and TFIMA(base) on out-of-domain datasets. TFIMA (base) is the base version of TFIMA model, without using direction attention, saliency embedding and paired markers modules. Δ line in each block refers to the performance differences between TFIMA and TFIMA (base) in the same block. There are two groups (TI & FI) of metrics (P, R and F1) in each block. The largest difference and the second largest difference are respectively shown in bold and bold-italics in each group.

Domains	Models	TI			FI		
		P	R	F1	P	R	F1
Newspaper	TFIMA	81.85	64.25	71.99	67.49	52.98	59.36
	TFIMA(base)	85.25	47.15	60.72	76.35	42.23	54.38
	Δ	-3.40	17.10	11.27	-8.85	10.75	4.98
Travel-guide	TFIMA	100.00	99.03	99.51	97.98	97.03	97.50
	TFIMA(base)	100.00	96.32	98.13	98.46	94.84	96.61
	Δ	0.00	2.71	1.39	-0.48	2.19	0.89
Letter	TFIMA	91.09	69.17	78.63	72.61	55.14	62.68
	TFIMA(base)	87.26	46.37	60.56	74.06	39.35	51.39
	Δ	3.82	22.81	18.08	-1.45	15.79	11.29
Journal	TFIMA	81.73	68.33	74.43	70.43	58.89	64.15
	TFIMA(base)	80.97	50.83	62.46	71.24	44.72	54.95
	Δ	0.75	17.50	11.98	-0.81	14.17	9.20
Email	TFIMA	74.49	64.60	69.19	64.29	55.75	59.72
	TFIMA(base)	69.51	50.44	58.46	59.76	43.36	50.26
	Δ	4.98	14.16	10.73	4.53	12.39	9.46

out-of-domain datasets, our proposed method can still achieve better performance. The conclusions are consistent with those on in-domain datasets.

6.5.3 Analysis of Performance of FSRLMA on Varying Length of Arguments

For the FSP task, the length of arguments varies greatly. To analyze the recognition capability of the FSRLMA model for arguments of different lengths, as well as the impact of the marker-based span representation method on model performance, we compare the performance of FSRLMA with its variant model that does not utilize frame markers and single levitated markers.

We have observed significant improvement in the model's ability to identify moderate-length arguments when compared with the FSRLMA model that does not utilize frame markers and single levitated markers. Table 14 illustrates the absolute improvements achieved by FSRLMA in terms of F1 score, varying with

argument length, compared to the counterpart model without markers. The frame marker provides valuable frame information, enabling the model to identify arguments more accurately, while the single levitated markers enhance longer span representations by capturing internal span information through directional attention. Although identification of moderate-length arguments (3-10) has notably improved, we noticed that improvements are subtle for arguments longer than 11.

Table 14. Improvements of FSRLMA in F1 score with varying argument lengths across two datasets.

Argument length	1	2	3-5	6-10	≥ 11
FN1.5	1.1	1.1	1.8	1.4	0.1
FN1.7	1.2	1.1	1.9	1.6	0.1

The above analysis suggests that although our marker-based method has achieved substantial performance improvements in moderate-length arguments identification, the identification of even longer arguments remains challenging.

6.5.4 Case Analysis for TFIMA and FSRLMA

To visually demonstrate the characteristics of our proposed two-stage methods, we select specific cases for intuitive illustration for two learned models.

For the TFIMA model, we have observed that it is much more helpful in improving the model's identification ability for unseen or rarely seen target words compared to the model TFIMA without using (lexical unit, frames) mapping knowledge and paired markers. As shown in Table 15, although the target “apprehended” in sentence 1 and “deceptive” in sentence 2 are unseen words in the training dataset, the TFIMA model can identify them correctly. Even for words appearing multiple times in the training data, our proposed method is also useful. For targets “get”, “by”, and “operated” in sentence 3, which appear many times in the training data, TFIMA identifies them correctly, while its counterpart cannot. These cases demonstrate that the incorporated knowledge prompts the TFIMA model to have a higher bias toward favoring words that appear in the knowledge base.

Additionally, the current datasets, including FN1.5 and FN1.7 datasets, have annotation incompleteness issues. As shown in sentence 2, the word “forced” should be a target that can evoke the frame *Causation*, while it is not annotated in the dataset.

It is commonly believed that two-stage methods may encounter issues with error propagation, i.e., incorrect prediction in the first stage inevitably leading to incorrect prediction in the second stage as well. However, for two-stage FSP, the problem of error propagation is not as severe, largely due to the frame-to-frame relations among frames. In essence, even if the frame associated with a particular target is predicted incorrectly, the assigned argument roles may not necessarily be incorrect. For instance, as depicted in

Figure 5, although the predicted frames evoked by the targets “thought” and “move” are incorrect, the predicted arguments and their roles are mostly correct, except for the role “Content” in case 1. This phenomenon can be attributed to the frame-to-frame relations among frames, as closely related frames often share some roles.

Table 15. Case studies for TFIMA. The golden targets are marked in bold, while line (a) and (b) are predicted targets by TFIMA and TFIMA without using (lexical unit, frames) mapping knowledge and paired markers.

ID	sentence
1	Immediately after the interview, he was apprehended again by the Jordanian authority. (a) after, <i>apprehended</i> , authority (b) after, authority
2	This behavior and not the edict itself forced them to be deceptive . (a) behavior, <i>forced</i> , <i>deceptive</i> (b) behavior, <i>forced</i>
3	The easiest way to get to Macau is by jetfoil, operated by TurboJet. (a) easiest, way, <i>get</i> , <i>by</i> , <i>operated</i> (b) easiest, way

Case 1: I thought i heard a holler there in the background but i was n’t sure.

- (a) (thought, *Cogitation*){
(I, *Cognizer*),
(i head a holler there in the background, *Topic*)}
- (b) (thought, *Awareness*){
(I, *Cognizer*),
(i head a holler there in the background, *Content*)}

Case 2: They would have to move it to Singapore for people not to know about it practically.

- (a) (move, *Cause_motion*){
(it, *Theme*),
(to Singapore, *Goal*)}
- (b) (move, *Motion*){
(it, *Theme*),
(to Singapore, *Goal*)}

Figure 5. Two cases for two-state FSP using TFIMA and FSRLMA. Targets are marked in bold with underline in two sentences. In each example, (a) is the golden frame-semantic structure, while (b) is the predicted result.

For example, in case 1, the frame Mental activity is inherited by the frame Awareness, and the frame Cogitation uses the frame Mental activity. Here, “inherited by” and “use” represent two types of frame-to-frame relations. Since the frame Awareness and Cogitation share the same role *Cognizer*, the role *Cognizer* for the argument “I” is correctly predicted even if the predicted frame is incorrect. Similarly, in case 2, because the frame Cause motion is causative of the frame Motion, where “be causative of” is a kind of frame-to-frame relations, they share some roles, such as *Theme* and *Goal*. Thus, even if the predicted frame is wrong, the roles for two arguments “it” and “to Singapore” are still correctly predicted.

Given the prevalence of frame relationships within frames, two-stage methods do not introduce a significant issue of error propagation. Therefore, based on the experimental analysis and case studies, we can conclude that our proposed two-stage method may be more suitable for solving the FSP task compared to joint learning models, such as FParser and Our(Joint FSRLMA).

6.5.5 FSP Performance using LLMs

Recent advancements in large language models (LLMs) have revolutionized natural language processing tasks, often achieving remarkable success comparable to or even surpassing humanlevel performance [42]. To evaluate their performance in Frame-Semantic Parsing (FSP) task, we employ LLaMA-2-13B-chat [43] as the base model. Since FSRL subtask is more challenging, we mainly focus on the FSRL subtask for preliminary testing and analysis, where the target and frame are provided in advance.

The input prompt for the LLMs is structured as follows: “Given the frame ‘[fname]’ evoked by the target ‘[target]’, it includes the following roles: [role1], [role2], ..., [rolen] and the Frame ‘[fname]’ refers to [frame definition]. [role1] refers to ...; [role2] refers to ...; ...; [rolen] refers to Here are several annotation examples: [demonstration]. Please extract the frame semantic roles around the target word ‘[target]’ from the sentence: ‘[sent]’, in the form of a. sentence: b. annotations:”, where [fname], [target], [role1], [role2], ..., [rolen], [sent], and [demonstration] are placeholders for the frame name, target, role names, sentence, and demonstration examples, respectively. Typically, a prompt comprises three components: background knowledge, demonstration, and instruction. An example prompt is illustrated in Figure 6 for the sentence “The only thing I did know for sure was here’s a chance to change things for me and my children”, where the target is “know” and the evoked frame is “Awareness”.

The background knowledge section provides information about the current target, the evoked frame, and relevant details about the frame and its roles, such as frame definition and role definitions. The demonstration part showcases several annotated examples in a specific format. The instruction part outlines the requirements for the FSRL task.

As depicted in Table 16, a notable performance gap is evident between FSRLMA and the models based on LLaMA-2-13B-chat. LLaMA-2-13B-chat utilizes annotated examples with the same target as the testing sentence in the training dataset as demonstrations, whereas LLaMA-2-13B-chat (w/o demonstration) does not incorporate demonstrations. Although the comparison may not be entirely equitable, these preliminary tests underscore the substantial difficulty of the FSP task, even when LLMs (such as LLaMA-2-13B-chat)

<i>#background knowledge</i> Given the frame ‘Awareness’ evoked by the target ‘know’, it includes the following roles: [Cognizer], [Content], [Evidence], [Topic], [Degree], [Manner], [Expressor], [Role], [Paradigm], [Time], [Explanation], Frame ‘Awareness’ refers to A [Cognizer] has a piece of [Content] in their model of the world. [Cognizer] refers to [Content] refers to [Explanation] refers to
<i>#Demonstration</i> Here are several annotation examples: a. sentence: But we all know how that story turned out. b. annotations: [Cognizer]: we all; [Content]: how that story turned out; a. sentence: Step out of the hotel and your senses will be bombarded by a range of sights, sounds, smells, tastes, and textures that let you know that you could only be in Jamaica. b. annotations: [Cognizer]: you; [Content]: that you could only be in Jamaica;
<i>#Instruction</i> please extract the frame-semantic roles around the target word ‘know’ from the sentence: ‘The only thing I did know for sure was here’s a chance to change things for me and my children’, in the form of a. sentence: b. annotations: like the examples.

Figure 6. Prompt example for FSRL using LLaMA-2-13B-chat. A prompt consists of three components: background knowledge, demonstration, and instruction. Please note that the definition for roles is not fully listed due to space constraints.

Table 16. The FSRL performance comparison between FSRLMA and LLaMA-2-13B-chat.

Data	Model	P	R	F1
FN1.5	FSRLMA	74.00	76.02	75.00
	LLaMA-2-13B-chat	20.64	37.11	26.56
	LLaMA-2-13B-chat (w/o demonstration)	10.94	21.12	14.41
FN1.7	FSRLMA	76.25	74.52	75.38
	LLaMA-2-13B-chat	22.18	38.34	28.10
	LLaMA-2-13B-chat (w/o demonstration)	10.84	21.20	14.35

are provided with the target word, frame definition, role names, and role definitions. In other words, FSP is a challenging task for LLMs, which can be attributed to the following reasons. Firstly, the concepts of frames and roles are derived from cognitive linguistics, expressing highly abstract semantics. Secondly, the datasets annotated with frame and role information are very scarce. Therefore, LLMs, trained on vast amounts of data but without explicit supervision signals about frame-semantics, find it difficult to capture abstract frame-semantic information.

In future work, we plan to explore instruction fine-tuning to delve deeper into the capabilities of LLMs for FSP. Additionally, we aim to investigate methods for integrating our proposed marker-based approaches

with LLM models, thereby harnessing the strengths of both approaches and facilitating greater performance enhancements in FSP.

7. CONCLUSION AND FUTURE WORK

This work proposes a novel two-stage framework for FSP utilizing marker-based attention, which can efficiently extract all the frame-semantic structures from a given sentence. The first stage achieves joint TI and FI identification, which adopts paired levitated-markers and markerbased attention to incorporate (lexical unit, frame) mappings into the potential target representations. The second stage achieves FSRL, which uses solid markers to incorporate frame information into the encoder, and employs single levitated markers and marker-based attention to enhance the representations of potential arguments. Comprehensive experiments validate the effectiveness of our approach. The experimental results reveal that: (1) our proposed two-stage method is more suitable for solving FSP task, and it overcomes the shortcomings of the previous proposed end-to-end FSP models; (2) the combination of paired levitated-markers and marker-based attention is effective in incorporating (lexical unit, frame) mappings into target representations, which provides a new method for incorporating span-level mapping knowledge; (3) the combination of single levitated markers and marker-based attention is efficient to enhance argument representations, which provides a new way to address the shortcomings of the previous studies, adopting boundary-based span representation method.

The proposed method achieves state-of-the-art performance in an end-to-end manner and highly competitive performance in individual subtasks. Nevertheless, there is still room for investigation into incorporating more knowledge into the proposed framework, such as constructional grammar knowledge for argument, frame definitions and frame relations knowledge. In addition, how to identify even longer arguments is still challenging. Future research could focus on integrating more knowledge into our proposed framework, as well as addressing the challenge of identifying roles for extremely long arguments, aided by LLMs-based methods, such as instruction tuning [44], retrieval augmented generation [45], etc..

AUTHOR CONTRIBUTIONS

Xuefeng Su (suexf@163.com) was responsible for designing the research framework, performing the research and manuscript writing. Ru Li (liru@sxu.edu.cn) proposed the research problem. Xiaoli Li (xlli@ntu.edu.sg) made contributions to writing structure optimization and manuscript revisions. Zhichao Yan (751824801@qq.com) contributed to collect and analyze data.

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China (No. 61936012), the Basic Research Program of Shanxi Province (No. 202203021211286), the Science and Technology

Cooperation and Exchange Special Project of ShanXi Province (No. 202204041101016), and the Research Fund of Guangxi Key Lab of Multi-source Information Mining & Security. (No. MIMS22-10)

REFERENCES

- [1] C. Baker, C. Fillmore, and J. Lowe, "The Berkeley FrameNet Project," in *Proc. 36th Annu. Meeting Assoc. Computational Linguistics and 17th Int. Conf. Computational Linguistics*, vol. 1, 1998, pp. 86–90.
- [2] C. Fillmore and C. Baker, "Frame semantics for text understanding," in *Proc. NAACL WordNet and Other Lexical Resources Workshop*, 2001, pp. 1–10.
- [3] S. Liu et al., "Leveraging FrameNet to improve automatic event detection," in *Proc. 54th Annu. Meeting Assoc. Computational Linguistics*, vol. 1, 2016, pp. 2134–2143.
- [4] S. Guo et al., "Incorporating syntax and frame semantics in neural network for machine reading comprehension," in *Proc. 28th Int. Conf. Computational Linguistics*, 2020, pp. 2635–2641.
- [5] S. Guo et al., "A frame-based sentence representation for machine reading comprehension," in *Proc. 58th Annu. Meeting Assoc. Computational Linguistics*, 2020, pp. 891–896.
- [6] H. Zhao et al., "CFSRE: Context-aware based on frame-semantics for distantly supervised relation extraction," *Knowl.-Based Syst.*, vol. 210, no. 12, Art. no. 106480, 2020.
- [7] Y. Guan et al., "Frame semantic-enhanced sentence modeling for sentence-level extractive text summarization," in *Proc. 2021 Conf. Empirical Methods Natural Language Processing*, 2021, pp. 4045–4052.
- [8] Y. Guan et al., "Frame semantics guided network for abstractive sentence summarization," *Knowl.-Based Syst.*, vol. 221, no. 6, Art. no. 106973, 2021.
- [9] Y. Guan et al., "Integrating semantic scenario and word relations for abstractive sentence summarization," in *Proc. 2021 Conf. Empirical Methods Natural Language Processing*, 2021, pp. 2522–2529.
- [10] D. Das et al., "Frame-semantic parsing," *Comput. Linguistics*, vol. 40, no. 1, pp. 9–56, 2014.
- [11] K. Hermann et al., "Semantic frame identification with distributed word representations," in *Proc. 52nd Annu. Meeting Assoc. Computational Linguistics*, 2014, pp. 1448–1458.
- [12] O. Täckström, K. Ganchev, and D. Das, "Efficient inference and structured learning for semantic role labeling," *Trans. Assoc. Computational Linguistics*, vol. 3, pp. 29–41, 2015.
- [13] S. Swayamdipta et al., "Frame-semantic parsing with softmax-margin segmental RNNs and a syntactic scaffold," 2017. [Online]. Available: arXiv:1706.09528.
- [14] B. Hart and T. Mitchell, "A joint sequential and relational model for frame-semantic parsing," in *Proc. 2017 Conf. Empirical Methods Natural Language Processing*, 2017, pp. 1247–1256.
- [15] E. Bastianelli, A. Vanzo, and O. Lemon, "Encoding syntactic constituency paths for frame-semantic parsing with graph convolutional networks," 2020. [Online]. Available: arXiv:2011.13210.
- [16] H. Peng et al., "Learning joint semantic parsers from disjoint data," in *Proc. 2018 Conf. North American Chapter Assoc. Computational Linguistics: Human Language Technologies*, vol. 1, 2018, pp. 1492–1502.
- [17] X. Chen, C. Zheng, and B. Chang, "Joint multi-decoder framework with hierarchical pointer network for frame semantic parsing," in *Findings Assoc. Computational Linguistics: ACL-IJCNLP 2021*, 2021, pp. 2570–2578.
- [18] X. Su et al., "A span-based target-aware relation model for frame-semantic parsing," *ACM Trans. Asian Low-Resource Language Information Processing*, vol. 22, pp. 1–24, 2022.

- [19] C. Zheng et al., “A double-graph based framework for frame semantic parsing,” in *Proc. 2022 Conf. North American Chapter Assoc. Computational Linguistics: Human Language Technologies*, 2022, pp. 4998–5011.
- [20] Z. Lin, Y. Sun, and M. Zhang, “A graph-based neural model for end-to-end frame semantic parsing,” in *Proc. 2021 Conf. Empirical Methods Natural Language Processing*, 2021, pp. 3864–3874.
- [21] Z. Zhong and D. Chen, “A frustratingly easy approach for entity and relation extraction,” in *Proc. 2021 Conf. North American Chapter Assoc. Computational Linguistics: Human Language Technologies*, 2021, pp. 50–61.
- [22] W. Zhou and M. Chen, “An improved baseline for sentence-level relation extraction,” in *Proc. 2nd Conf. Asia-Pacific Chapter Assoc. Computational Linguistics and 12th Int. Joint Conf. Natural Language Processing*, vol. 2, 2022, pp. 161–168.
- [23] D. Ye et al., “Packed levitated marker for entity and relation extraction,” in *Proc. 60th Annu. Meeting Assoc. Computational Linguistics*, vol. 1, 2022, pp. 4904–4917.
- [24] C. Fillmore, “Frames and the semantics of understanding,” in *Proc. Annu. Meeting Berkeley Linguistics Society*, 1985, pp. 1–10.
- [25] C. Baker, M. Ellsworth, and K. Erk, “SemEval-2007 task 19: Frame semantic structure extraction,” in *Proc. 4th Int. Workshop Semantic Evaluations (SemEval-2007)*, 2007, pp. 99–104.
- [26] J. Devlin et al., “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proc. 2019 Conf. North American Chapter Assoc. Computational Linguistics: Human Language Technologies*, vol. 1, 2019, pp. 4171–4186.
- [27] L. Zhuang et al., “A robustly optimized BERT pre-training approach with post-training,” in *Proc. 20th Chinese National Conf. Computational Linguistics*, 2021, pp. 1218–1227.
- [28] S. Tan and J. Na, “Positional attention-based frame identification with BERT: A deep learning approach to target disambiguation and semantic frame selection,” 2019. [Online]. Available: [arXiv:1910.14549](https://arxiv.org/abs/1910.14549).
- [29] T. Jiang and E. Riloff, “Exploiting definitions for frame identification,” in *Proc. 16th Conf. European Chapter Assoc. Computational Linguistics*, vol. 1, 2021, pp. 2429–2434.
- [30] X. Su et al., “A knowledge-guided framework for frame identification,” in *Proc. 59th Annu. Meeting Assoc. Computational Linguistics and 11th Int. Joint Conf. Natural Language Processing*, vol. 1, 2021, pp. 5230–5240.
- [31] K. An et al., “Coarse-to-fine dual encoders are better frame identification learners,” in *Findings Assoc. Computational Linguistics: EMNLP 2023*, 2023, pp. 13455–13466.
- [32] C. Zheng, Y. Wang, and B. Chang, “Query your model with definitions in FrameNet: An effective method for frame semantic role labeling,” in *Proc. 37th AAAI Conf. Artificial Intelligence*, 2023, pp. 1–10.
- [33] A. Vaswani et al., “Attention is all you need,” in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 5998–6008.
- [34] J. Liu, Y. Chen, and J. Xu, “Saliency as evidence: Event detection with trigger saliency attribution,” in *Proc. 60th Annu. Meeting Assoc. Computational Linguistics*, vol. 1, 2022, pp. 4573–4585.
- [35] J. Ruppenhofer et al., “FrameNet II: Extended theory and practice,” 2016. [Online]. Available: <https://framenet2.icsi.berkeley.edu/docs/r1.7/book.pdf>.
- [36] P. Qi et al., “Stanza: A Python natural language processing toolkit for many human languages,” in *Proc. 58th Annu. Meeting Assoc. Computational Linguistics: System Demonstrations*, 2020, pp. 101–108.
- [37] A. Kalyanpur et al., “Open-domain frame semantic parsing using transformers,” 2020. [Online]. Available: [arXiv:2010.10998](https://arxiv.org/abs/2010.10998).
- [38] Z. Dai et al., “Funnel-transformer: Filtering out sequential redundancy for efficient language processing,” in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020, pp. 4271–4282.

- [39] Z. Huang, X. Jia, T. Chen, and Z. Zhang, “A general scalable approach for knowledge selection based on iterative hybrid encoding and re-ranking,” *Data Intelligence*, vol. 7, no. 1, pp. 124–142, 2025.
- [40] S. Hartmann *et al.*, “Out-of-domain FrameNet Semantic Role Labeling,” in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2017, pp. 471–482.
- [41] R. Passonneau *et al.*, “The MASC Word Sense Corpus,” in *Proc. 8th Int. Conf. Lang. Resour. Eval. (LREC)*, 2012, pp. 3025–3030.
- [42] W. Zhao *et al.*, “A Survey of Large Language Models,” 2023. [Online]. Available: [arXiv:2303.18223](https://arxiv.org/abs/2303.18223).
- [43] H. Touvron *et al.*, “Llama 2: Open Foundation and Fine-Tuned Chat Models,” 2023. [Online]. Available: [arXiv:2307.09288](https://arxiv.org/abs/2307.09288).
- [44] S. Longpre *et al.*, “The Flan Collection: Designing Data and Methods for Effective Instruction Tuning,” 2023. [Online]. Available: [arXiv:2301.13688](https://arxiv.org/abs/2301.13688).
- [45] Z. Yan *et al.*, “Atomic Fact Decomposition Helps Attributed Question Answering,” 2024. [Online]. Available: [arXiv:2410.16708](https://arxiv.org/abs/2410.16708).

AUTHOR BIOGRAPHY



Xuefeng Su is currently a Ph.D. student at Shanxi University, China, and he is also an associate professor at Shanxi Vocational University of Engineering Science and Technology, Shanxi Province, China. He obtained his M.S. degree from Taiyuan University of Technology, China in 2007. His research interests include natural language processing and frame semantic parsing.



Ru Li is currently a professor at Shanxi University, China, and she received the Ph.D. degree from Shanxi University, China in 2012. She has published more than 100 papers in top international academic conferences and journals and received two prizes for scientific and technological progress in Shanxi. Her research interests include natural language processing, knowledge graph and information extraction.



Xiaoli Li is currently an IEEE fellow, and holds the positions of Director and Chief Scientist at the Institute of Machine Intelligence in the Institute for Info-Comm Research at A*STAR, Singapore. He received the Ph.D. degree from Institute of Computing Technology, Chinese Academy of Sciences, China in 2001. His research interests include AI, data mining, machine learning, and bioinformatics.



Zhichao Yan is currently a Ph.D. student at Shanxi University, China, and he received his M.S. degree from Shanxi University, China in 2023. He has participated in the project of research on Chinese frame semantic computing based on cognitive (key research project supported by the National Natural Science Foundation) mechanism behind language facts. His research interests include natural language processing and deep learning.