A Knowledge-Guided Framework for Frame Identification

Xuefeng Su[1,2], Ru Li[1,3,*], Xiaoli Li[4], Jeff Z.Pan[5], Hu Zhang[1], Qinghua Chai[1], Xiaoqi Han[1]
1. School of Computer and Information Technology, Shanxi University, Taiyuan, China
2. School of E-commerce and Logistics, Shanxi Vocational University of Engineering Technology, Taiyuan, China
3. Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, Shanxi University, Taiyuan, China
4. Institute for Infocomm Research, A*Star, Singapore
5. ILCC, School of Informatics, University of Edinburgh, UK
{suexf,xiaoqisev}@163.com, {liru,zhanghu,charles}@sxu.edu.cn
xlli@ntu.edu.sg, j.z.pan@ed.ac.uk

Abstract

Frame Identification (FI) is a fundamental and challenging task in frame semantic parsing. The task aims to find the exact frame evoked by a target word in a given sentence. It is generally regarded as a classification task in existing work, where frames are treated as discrete labels or represented using one-hot embeddings. However, the valuable knowledge about frames is neglected. In this paper, we propose a **K**nowledge-**G**uided **F**rame **I**dentification framework (**KGFI**) that integrates three types frame knowledge, including frame definitions, frame elements and frame-to frame relations, to learn better frame representation, which guides the KGFI to jointly map target words and frames into the same embedding space and subsequently identify the best frame by calculating the dot-product similarity scores between the target word embedding and all of the frame embeddings. The extensive experimental results demonstrate KGFI significantly outperforms the state-of-the-art methods on two benchmark datasets.

1 Introduction

Frame **I**dentification (FI) aims to find the exact frame evoked by a *target word* in a given sentence. A frame represents an event scenario, and possesses frame elements (or semantic roles) that participate in the event (Hermann et al., 2014), which is described in the FrameNet knowledge base (Baker et al., 1998; Ruppenhofer et al., 2016) grounded on the theory of Frame Semantics (Fillmore et al., 2002). The theory asserts that people understand the meaning of words largely by virtue of the frames which they evoke. In general, many words are polysemous and can evoke different frames in different contexts.

As shown in Figure 1, the word *stopped* evokes the frame **Activity_stop** and the frame **Process_stop** respectively in two sentences. It is a challenging task to distinguish the frames evoked by target words in sentences. Furthermore, FI is a key step before Frame Semantic Role Labeling (**FSRL**) (Das et al., 2010, 2014; Swayamdipta et al., 2017; Kalyanpur et al., 2020) which is widely used in event recognition (Liu et al., 2016), machine reading comprehension (Guo et al., 2020b,a), relation extraction (Zhao et al., 2020), etc. Through FI process, hundreds of role labels in FrameNet are reduced to a manageable small set (Hartmann et al., 2017), which can significantly improve the performance of FSRL models. Thus, FI is a fundamental and critical task in NLP.

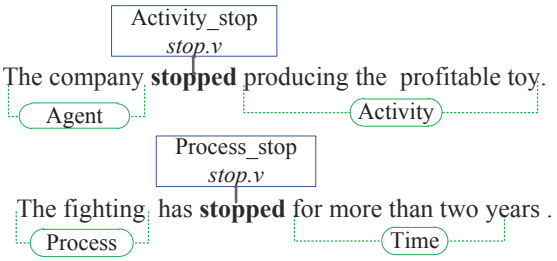

Figure 1: Two annotated examples with the target word marked in bold and frame elements (semantic roles) in rounded rectangles. The target word *stopped* (*stop.v* denotes its form of lexical unit) evokes the frame **Activity_stop** and the frame **Process_stop** respectively in different contexts. Here, the key to distinguish these two frames is identifying whether the subject (*The compony* or *The fighting*) of stopped is an Agent or a Process (see the frame definitions in Table 1).

FI is typically regarded as a classification task, in which class labels are frame names. In earlier studies, researchers manually construct features and then use supervised learning methods to learn classification models (Bejan and Hathaway, 2007; Johansson and Nugues, 2007; Das et al., 2010, 2014). These methods, however, do not take the valuable semantic information about frames into considera-

*Corresponding author.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 5230–5240
August 1–6, 2021. ©2021 Association for Computational Linguistics

	Frame: **Activity_stop**	Frame: **Process_stop**
Def	An *Agent* ceases an *Activity* without completing it	A *Process* stops at a certain *Time* and *Place*
FEs	core: *Agent, Activity* peripheral: *Degree, Duration, Manner, Time* extra-thematic: *Depictive,Purpose, Result,...*	core: *Process* peripheral: *Manner, Place, Time* extra-thematic: *Depictive, Duration, ...*
LUs	abandon.v, cease.v, halt.v, quit.v, **stop.v**, ...	cease.v, halt.n, shutdown.n, **stop.v**,...
FRs	Inherits from: Process_stop Subframe of: Activity Is Inherited by: Halt Uses: Eventive_affecting	Inherits from: Event Subframe of: Process Is Inherited by: Activity_stop

Table 1: The structured descriptions for frame **Activity_stop** versus frame **Process_stop** in FrameNet1.7. The description of a frame is mainly composed of frame definition (**Def**), frame elements (**FEs**), lexical units (**LUs**) and frame-to-frame relations (**FRs**). Note that the elements of FEs, LUs and FRs are partially listed due to the limited space[†]. Lexical unit is expressed in the form of *lemma.POS* (e.g. *stop.v*).

tion, and merely treat them as discrete labels.

The recent studies of FI use distributed representations of target words and their syntactic context to construct features, and construct classification models with deep neural network (Hartmann et al., 2017; Kabbach et al., 2018). These methods usually transform frame labels into one-hot representations (Hermann et al., 2014; Täckström et al., 2015), and then learn the embeddings of target words and frames simultaneously. However, the abundant semantic information and structure knowledge of frames contained in FrameNet are still neglected.

Knowledge of frames defined by linguists, such as frame definition, frame elements and frame-to-frame relations, can enrich frame labels with rich semantic information that can potentially guide FI models to learn more unique and distinguishable representations. Thus, in this paper, we propose a **K**nowledge **G**uided **F**rame **I**dentification framework (**KGFI**) which consists of a Bert-based context encoder and a frame encoder based on a specialized graph convolutional network (FrameGCN). In particular, the frame encoder incorporates multiple types of frame knowledge into frame representation which guides the KGFI to jointly map target words and frames into the same embedding space. Instead of predicting the frame label directly, KGFI chooses the best suitable frame evoked by the target word in a given sentence by calculating the dot-product similarity scores between the target word embedding and all of the frame embeddings. In summary, our contribution is threefold:

- To the best of our knowledge, we are the first to propose a unified FI method which leverages heterogeneous frame knowledge for building rich frame representations.

- We design a novel Framework KGFI, consisting of a Bert-based context encoder and a GCN-based frame encoder, which learns the model from a combination of annotated data and FrameNet knowledge base, and maps target words and frames into the same embedding space.

- Extensive experimental results demonstrate our proposed KGFI framework outperforms the state-of-the-art models across two benchmark datasets.

2 FrameNet and FI Task Definition

2.1 FrameNet

FrameNet is built on the hypothesis that people understand things by performing mental operations on what they already know (Baker et al., 1998). Such knowledge reflecting people's cognitive experience is described as structured information packets called **frames**. A frame represents an event scenario, associated with a set of semantic roles (**frame elements (FEs)**). **Lexical units (LUs)** are capable of evoking the scenario (Kshirsagar et al., 2015). Frame elements in terms of how central they are to a particular frame can be divided into three distinguishing levels: core, peripheral and extra-thematic. Each frame has a textual **definition (Def)**, depicting the scenario and how the roles interact in the scenario. Frames are organized as a network with several kinds of **frame-to-frame relations (FRs)**.

[†]See the details in https://FN.icsi.berkeley.edu/fndrupal/

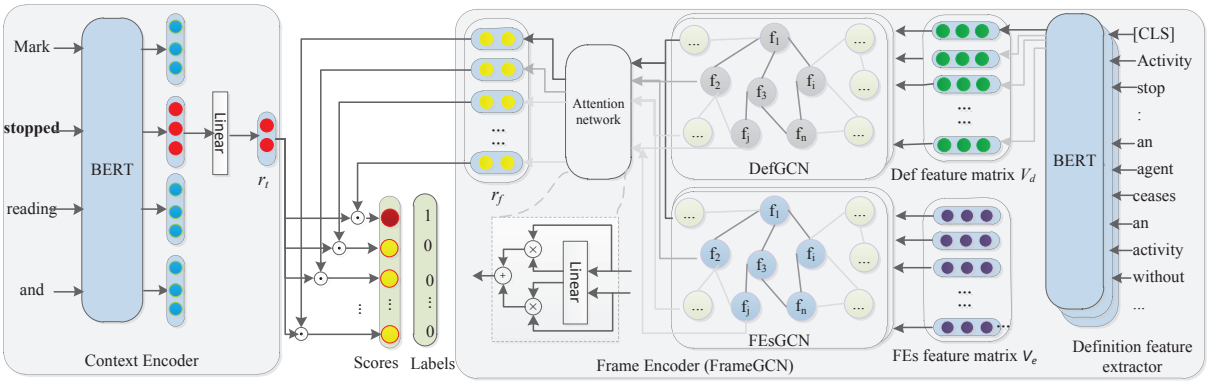

Figure 2: The overall architecture of KGFI.

Table 1 shows the structure of frame **Activity_stop** and frame **Process_stop** in FrameNet.

2.2 FI Task Definition

Frame **I**dentification (**FI**) is the task of predicting a frame evoked by the target word in a sentence. Let **c**=w_0,w_1,...,w_{st},...,w_{en},...,w_n denote a given sentence, and **t**=w_{st},...,w_{en} ($t \subset c$) represent the target word, where *st* and *en* are the *start* and *end* index respectively for the target word **t** in the sentence. Let $F = (f_1, f_2, ..., f_{|F|})$ denote the set of all frames in FrameNet. The FI model is defined as a mapping function $G : (\mathbf{c}, \mathbf{t}, st, en) \rightarrow f_j$, subject to $f_j \in F$.

3 Methodology

Table 1 illustrates the structured knowledge (**Def**, **FEs**, **LUs**) of two different frames and their frame-to-frame relations (**FRs**). We explicitly leverage them to enrich the frame embeddings with semantic information. The resulted informative frame representations serve two purposes: 1) guide our model to learn more distinguishable embeddings of target words, and 2) improve FI model's generalization performance in the prediction phase.

The proposed **KGFI** framework consists of three components: **context encoder**, **frame encoder** and **scoring module**, as shown in Figure 2. Specifically, context encoder is used to represent the context-aware target word into an embedding with a Bert-based module, and frame encoder is used to incorporate three types of knowledge about a frame into frame embeddings. With the guidance of the knowledge about frames, two encoders jointly learn the embeddings of target words and frames. Finally, a scoring module is used to calculate the similarity scores between the given target word embedding and all frames' embeddings, to identify the best frame with the highest score.

3.1 Context Encoder

To get the context-aware embeddings of target words, we employ Bert (Devlin et al., 2019) for our context encoder, since its architecture is a multi-layer bidirectional Transformer which can aggregate information from context into the target word through the self-attention mechanism. As we know, Bert model is pre-trained on a large corpus and can transfer language knowledge into the context encoder, which is very helpful for the target word representation as the manually labeled training data of FI is very small.

The context encoder, which we define as E_c, takes given sentence **c** containing a target word **t** as input. We denote the last layer of Bert's output as H_t. The context encoder can be expressed as :

$$r_t = E_c(\mathbf{c}, \mathbf{t}, st, en) = W_c^T h_t + b_c \quad (1)$$

where

$$h_t = \frac{1}{en + 1 - st} \sum_{i=st}^{en} (H_t[i]), \quad (2)$$

$W_c \in \mathbb{R}^{n \times m}$and $b_c \in \mathbb{R}^m$ are learned parameters.

3.2 Frame Encoder

In FrameNet, all the frames are connected into a directed graph through the frame-to-frame relations, as shown in Figure 3. Moreover, the graph convolutional network(GCN) (Kipf and Welling, 2017) has been proved to be effective to model the relationship between labels (Yan et al., 2019; Chen et al., 2019; Cheng et al., 2020; Linmei et al., 2019), and it can enrich the representation of the node through aggregating information from its neighbors. In order to make better use of frame knowledge and the advantage of GCN, we propose a *specialized GCN*, called FrameGCN, to incorporate multiple frame knowledge into frame representations.

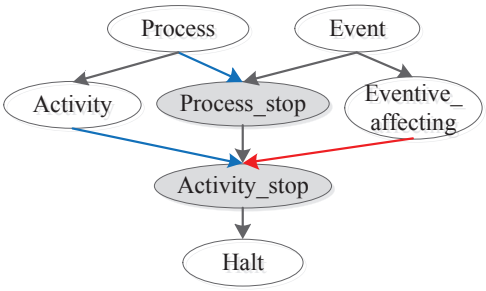

Figure 3: The sub-graph of overall graph of FrameNet1.7 corresponding to frame **Activity_stop** and **Process_stop**. The nodes denote frames and the directed edges denote frame-to-frame relations. The black "→", red "→" and blue "→" denote *Inheritance*, *Using* and *Subframe* relations respectively, and the direction of an arrow is from **super-frame** to **sub-frame**.

3.2.1 Structure of FrameGCN

FrameGCN is a combination of two dedicated GCNs (FEsGCN and DefGCN) and an attention network, as shown in Figure 2. FEsGCN is used to represent frame by aggregating the FEs features of its neighbors, while DefGCN is used to represent frame by aggregating the Def features of its neighbors. The attention network is responsible for incorporating the outputs of two GCNs into one unified embedding where adjacent matrix A is shared by the two dedicated GCNs.

Frame-to-frame relation in FrameNet is a asymmetric relation between two frames, where one frame is called **super-frame** and the other is called **sub-frame**, as shown in Figure 3. A frame typically obtains/inherits more information from its super-frame than from its sub-frame. Therefore, we define the adjacent matrix of the graph as a weighted asymmetric matrix denoted as $A = (a_{ij})_{|F|\times|F|}$, where

$$a_{ij} = \begin{cases} 3, & f_j = f_i \\ 2, & f_j \; is \; a \; super-frame \; of \; f_i \\ 1, & f_j \; is \; a \; sub-frame \; of \; f_i \\ 0, & other \end{cases}. \quad (3)$$

Three types of frame-to-frames relations, including *Inherits*, *Using* and *Subframe*, are used in this study.

3.2.2 FEsGCN

The FEs of a frame express its semantic roles and structure. Frames which have similar structures imply that they have close semantic, so we regard FEs as features and use them to represent frames. Let $FE = (e_1, e_2, ..., e_{|FE|})$ denote the set of all frame elements in FrameNet, and $V_e \in \mathbb{R}^{|F|\times|FE|}$ denote the feature matrix of frames represented by FEs. The ith row of V_e is the feature vector of ith frame f_i, and can be expressed as $V_e[i,:] = (ve_1, ve_2, ..., ve_{|FE|})$, where

$$ve_j = \begin{cases} 1, & e_j \in FE_{f_i} \\ 0, & other \end{cases}, \quad (4)$$

$FE_{f_i} \subset FE$ is the FEs of frame f_i.

FEsGCN is used to learn a map function which maps the node (frame) vectors represented by FEs to a new representation via convolutional operation defined by A. We take a two-layer GCN to implement the map function, which can be expressed as:

$$\begin{aligned} g_e^{(0)}(A, V_e) &= ReLU(AV_eW_e^{(0)}), \\ g_e^{(1)}(A, V_e) &= Tanh(Ag_e^{(0)}(A, V_e)W_e^{(1)}). \end{aligned} \quad (5)$$

Here, $W_e^{(0)} \in \mathbb{R}^{|FE|\times h}$ is an input-to-hidden weight matrix for the hidden layer and $W_e^{(1)} \in \mathbb{R}^{h\times m}$ is a hidden-to-output weight matrix.

3.2.3 DefGCN

Since the frame definition is a short text that depicts an event scenario and frame elements that participate in the event, we employ Bert as a feature extractor to construct the feature matrix V_d of frames. Specifically, we first input a frame definition into Bert, and subsequently take the first token's representation (corresponding to the input [CLS] token) in Bert's last layer as the feature vector of the frame. Since the name of a frame is also meaningful, we concatenate the frame name and frame definition into one string, e.g. *Activity stop: an agent ceases an activity without completing it*.

DefGCN is used to learn a map function which maps the node (frame) vectors represented by definition to a new representation via convolutional operation defined by A. We use a network similar to FEsGCN, which can be expressed as:

$$\begin{aligned} g_d^{(0)}(A, V_d) &= ReLU(AV_dW_d^{(0)}), \\ g_d^{(1)}(A, V_d) &= Tanh(Ag_d^{(0)}(A, V_d)W_d^{(1)}). \end{aligned} \quad (6)$$

Here, $W_d^{(0)} \in \mathbb{R}^{n\times h}$ is an input-to-hidden weight matrix for a hidden layer with h feature maps, and $W_e^{(1)} \in \mathbb{R}^{h\times m}$ is a hidden-to-output weight matrix.

3.2.4 Attentive Graph Combination

We use an attention network to dynamic incorporate the outputs of FEsGCN and DefGCN into one frame embedding through the attention weighting

mechanism. The incorporation operation takes the following function:

$$r_{f_i} = \sum_{k \in \{e,d\}} a_{i,k} g_k^{(1)}(A, V_k)_i \quad (7)$$

where $r_{f_i} \in \mathbb{R}^m$ is the embedding of ith frame, $g_k^{(1)}(A,V_k)_i$ is the ith row of convolved representation of graph k, and $a_{i,k}$ is a weight of ith frame against the graph k, which is computed as:

$$a_{i,k} = \frac{exp(w_a g_k^{(1)}(A,V_k)_i)}{\sum_{k' \in \{e,d\}} exp(w_a g_{k'}^{(1)}(A,V_{k'})_i)} \quad (8)$$

where $w_a \in \mathbb{R}^m$is a learnable vector.

3.3 Scoring and Prediction

After obtaining the embeddings of target words and frames through context encoder and frame encoder respectively, we score a target word t with each frame $f_j \in F$ by computing the dot product similarity between r_t and each r_f for $f_j \in F$:

$$S(r_t, r_{f_j}) = r_t . r_{f_j}, j = 1, 2, ..., |F| \quad (9)$$

During training, all model parameters are jointly learned by minimizing a cross-entropy loss:

$$L(\theta) = -\frac{1}{|D|} \sum_{i}^{|D|} \sum_{j}^{|F|} y_{ij} log(\hat{y}_{ij}) \quad (10)$$

where D is the number of the training data, $|F|$ is the total number of frames in FrameNet, y_{ij} (one-hot representation of frame labels) and $\hat{y}_{ij}$ are true labels. The predicted probability over frames is calculated by the *softmax* function over the scores.

During prediction, we predict the frame evoked by the target word t to be $f_j \in F$, whose representation r_{f_j} has the highest score with r_t. The prediction function is defined as:

$$\hat{f} = argmax_{f_j \in F} S(r_t, r_{f_j}) \quad (11)$$

Note most of the frames contain a set of lexical units (**LUs**) in the form of *lemma.POS* (e.g. stop.v). As shown in Table 1, the LUs of the frame **Activity_stop** and the frame **Process_stop** are listed in the fourth row. Therefore, we adopt the *lexicon filtering* operation to reduce the possible candidate frame set. Firstly, we utilize lemmatization and POS tools to convert the target word t into the form of LU (e.g. stop.v). Secondly, we use this LU to match the frames whose LUs contains this LU, and then use the matched frames as the possible candidate frame set F_t for the target word t. At last, we predict the frame label by the following function:

Datasets	Train	Dev	Test	$\|F\|$	$\|FE\|$
FN1.7	19391	2272	6714	1221	1285
FN1.5	16621	2284	4428	1019	1170

Table 2: Statistics for FrameNet datasets.

$$\hat{f} = argmax_{f_j \in F_t} S(r_t, r_{f_j}) \quad (12)$$

In the light of the coverage issue of FrameNet (see Section 4.4), these two prediction functions (11 and 12) can be used in different circumstances. In general, we can first use LU to obtain candidate frame set F_t by performing *lexicon filtering* and then use function 12 to identify best frame from F_t. However, if we can not find any candidate frame using LUs, i.e. $F_t = \emptyset$, then we have to identify best frame from F using function 11. Note that F_t only contains a couple of candidate frames, while F contains more than one thousand of frames. This requires FI models have very good generalization performance to handle a big F set.

4 Experimental Settings

4.1 Datasets

We have employed two knowledge bases, i.e. FrameNet1.5 and FrameNet1.7. Both of them contain various documents which have been annotated manually, including target words and corresponding evoked frames. Documents and corresponding annotations in FrameNet1.7 are extended from FrameNet 1.5 and thus are more complete. Note train, dev and test documents in both data have been partitioned following (Swayamdipta et al., 2017). Given a sentence in documents may contain multiple target words, we regard it as multiple pairs of target word and sentence in train, dev and test sets. The statistics of two datasets are illustrated in Table 2.

To test the model's performance on the more challenging **ambiguous data**, following the previous studies, we constructed a specialized dataset by extracting pairs of target word and annotated sentence from test data, in which the target words are *polysemous* or can evoke multiple frames.

4.2 Baselines

We first compare the KGFI against five existing models. **Semafor** (Das et al., 2014) is a conditional log-linear model which uses statistical features about target word to predict the frame label. **Hermann-14** (Hermann et al., 2014) is a

joint learning model which maps frame labels and the dependency path of target word into a common embedding space. **SimpleFrameId** (Hartmann et al., 2017) models a classifier based on the embeddings of entire words in the sentence. **Open-Sesame** (Swayamdipta et al., 2017) models a classifier based on bi-directional LSTM. Hermann-14 converts frame labels into onehot embeddings, while other models treat frame labels as discrete supervision signals. **Peng's model** (Peng et al., 2018) is a joint learning model for FI and FSRL, which both uses exemplars in FrameNet knowledge base and the full-text annotation training data to train the model.

In addition, we also implemented two additional Bert-based baselines for fair comparison. One is called **Bert-cls** that uses Bert to represent the target word in a sentence and treats discrete frame labels as supervision signals. The other is called **Bert-onehot**, which also uses the dual-encoder architecture (Context encoder and frame encoder) and maps target words and frames into a common embedding space. The difference between KGFI and Bert-onehot is that KGFI uses GCN-based modules to incorporate frame knowledge into frame embeddings, while Bert-onehot uses a linear network to map onehot vector of frame labels into frame embeddings *without incorporating knowledge*. Clearly, we will test if the knowledge plays a significant role for better frame embeddings and subsequent FI task.

4.3 Parameter Settings

All Bert modules in KGFI were initialized with Bert-base. We set both the dimensions of target word embedding r_t and frame embedding r_f to 128 (m=128), the hidden layer size of FEsGCN and DefGCN to 256 (h=256). The size of Bert embedding is n=768. The dimensions of FEs and FRs feature vectors are related to FrameNet version (see Table 2). For optimization, we use BertAdam optimizer and set learning rate to $5e-5$. As for parameter tuning, our parameters are tuned using the development set with the early stop strategy.

4.4 Test Settings

FrameNet has a few coverage issues in that: (1) the LUs set is incomplete for some frames; (2) many words that should evoke frames are not included in LUs set of frames. Thus, we design two types of test settings: *test without lexicon filtering, or test that does not use LUs* (use Fun 11) and *test with lexicon filtering, or test that uses LUs* (use Fun 12).

	FN 1.7		FN 1.5	
Models	**All**	**Amb**	**All**	**Amb**
Semafor	-	-	83.60	69.19
Hermann-14	-	-	88.41	73.10
SimpleFrameId	83.00	71.70	87.63	73.80
Open-Sesame	86.55	72.40	86.40	72.80
Peng's model*	89.10	77.50	90.00	78.00
Bert-cls	90.17	79.87	90.13	78.32
Bert-onehot	90.57	80.66	91.46	80.78
KGFI(1-layer)	91.71	82.98	**92.13**	**82.34**
KGFI(2-layers)	**92.40**	**84.41**	91.91	81.84
Max △	1.83	3.75	0.67	1.56

Table 3: Frame identification accuracy *with lexicon filtering* setting on FrameNet test dataset. '**ALL**' and '**Amb**' denote testing on test data and on ambiguous data respectively. 'Max △' denotes the accuracy difference between our best KGFI model and the strongest baseline Bert-onehot. '*' denotes the training data and exemplars in FrameNet are both used in training phase.

5 Evaluation

5.1 Overall Results

The overall testing results, as shown in Table 3, demonstrate that Bert-cls and Bert-onehot are two strong baselines, outperforming all of the prior work that does not incorporate pre-training modules into their systems. Bert-onehot slightly outperforms Bert-cls in all of the testing settings, indicating joint learning target word embedding and frame embedding is helpful for FI task.

Our best KGFI models, including KGFI (2-layers) for FrameNet1.7 and KEFI (1-layer) for FrameNet1.5, outperform all the baseline models of FI in terms of accuracy. Compared with the stronger Bert-onehot model, our model achieves absolute 1.83% and 0.67% improvements on two datasets respectively in **All** test setting. With the help of lexicon filtering with LUs in FrameNet, the model predicts the exact frame evoked by the target word among a small set of candidate frames. Clearly, the improvements are credited to the model's performance improvement in predicting frames for ambiguous target words, since the model achieves absolute 3.75% and 1.56% improvements in **Amb** test setting on two datasets respectively.

To the best of our knowledge, few previous work focus on frame prediction without lexicon filtering

	FN 1.7		FN 1.5	
Models	**All**	**Amb**	**All**	**Amb**
SimpleFrameId	76.10	-	77.49	-
Bert-onehot	80.09	75.29	82.00	76.11
KGFI(1-layer)	84.95	79.78	**85.63**	**80.07**
KGFI(2-layers)	**85.81**	**80.66**	85.00	79.66
Max △	5.72	5.37	3.63	3.96

Table 4: Frame identification accuracy *without lexicon filtering* on FrameNet test dataset. '**ALL**' and '**Amb**' denote testing on test data and on ambiguous data respectively. 'Max △' denotes the accuracy difference between our best KGFI model and the strongest baseline Bert-onehot.

Models	**top-1**	**top-2**	**top-3**	**top-5**
Bert-onehot	80.09	87.17	88.96	90.12
KGFI(2-layer)	85.81	90.22	91.59	92.88

Table 5: Top-K accuracy of frame identification without lexicon filtering on FrameNet1.7.

except for SimpleFrameId model, so we choose SimpleFrameId and the stronger Bert-onehot model as our baseline to compare our best model's performance under no-lexicon filter setting. As shown in Table 4, in comparison with the stronger Bert-onehot model, our model achieves absolute 5.72% and 3.63% improvements on two datasets respectively in **all** setting (without using LUs and compared with more than 1000 frames), signifying the generalization performance of our model achieves significant improvement, considering that the model predicts the exact frame evoked by the target word among all the frames without knowing the possible candidate frames of the target word in no-lexicon filtering setting.

To further test the performance of our best KGFI model, we use the top-K accuracy to measure the model performance without lexicon filtering. The higher top-K accuracy indicates that the model has learned better frame representations. Furthermore, the model can reduce the candidate frame set into a small frame subset (containing K most possible frames), which is useful for the downstream tasks, such as LUs induction for FrameNet, FSRL, etc. As shown in Table 5, compared with Bert-onehot baseline, our best KGFI model achieves higher top-K (K=1,2,3,5) accuracy, which further demonstrates the model has learned the better frame representation through incorporating the frame knowledge.

Models	**All-L**	**All-nL**
Bert-onehot	90.57	80.09
KGFI(*w/* FrameGCN)	92.40	85.81
w/ DefGCN	91.49	82.10
w/ FEsGCN	92.01	85.00
w/o attention	92.31	85.19

Table 6: Ablation analysis on FrameNet1.7 dataset in All-L and All-nL setting. The sign 'w/' and 'w/o' denote that the KGFI is constructed with and without the corresponding module respectively. '-L' and '-nL' denote testing with and without lexicon filtering respectively.

Considering FrameNet1.5 dataset is relatively small, the performance of simple structure model (using 1-layer GCN) achieves the best performance, while the performance of the model using 2-layers GCN drops slightly. In general, no matter how many layers are adopted, our models outperform all the baselines and achieve the best performance on two datasets in all settings consistently.

5.2 Ablation Studies

To test the function of each component in KGFI, we conduct the ablation study. As shown in Table 6, the results demonstrate that all of the three components, i.e. DefGCN, FEsGCN and attention network, are helpful for enhancing the model's performance. Even with DefGCN or FEsGCN individually, the performance of our model is still better than the stronger baseline Bert-onehot, which indicates the frame definition, FEs and FRs are all useful knowledge for frame representation, and our proposed GCN-based model architecture is effective to incorporate them into the informative embeddings. Compared with frame definition, FEs are more useful for frame representation, since the performance of GKFI (with FEsGCN) outperforms KGFI (with DefGCN), although it slightly lags behind KGFI full model (with FrameGCN). Note that the attention module is removed when DefGCN or FEsGCN is used as the frame encoder.

As for the attention module, the performance of KGFI (with FrameGCN) drops when we replace it with a simple addition operation, suggesting it is necessary to use attention mechanism to integrate the outputs of DefGCN and FEsGCN.

5.3 Weighting Method for Adjacent Matrix

To test the rationality of our proposed weighting method for adjacent matrix A, we conduct a set of

Models	*A*	All-L	All-nL
KGFI(*w/* FrameGCN)	W	**92.4**	**85.81**
KGFI(*w/* FrameGCN)	B	91.80	83.10
KGFI(*w/* DefGCN)	W	**91.49**	**82.10**
KGFI(*w/* DefGCN)	B	91.26	81.11
KGFI(*w/* FEsGCN)	W	**92.01**	**85.00**
KGFI(*w/* FEsGCN)	B	91.78	82.10

Table 7: The results of KGFI models on FrameNet1.7 dataset under different value settings of adjacent matrix *A*. 'W' and 'B' denote the matrix *A* is weighted and binary respectively. '-L' and '-nL' denote testing with and without lexicon filtering respectively.

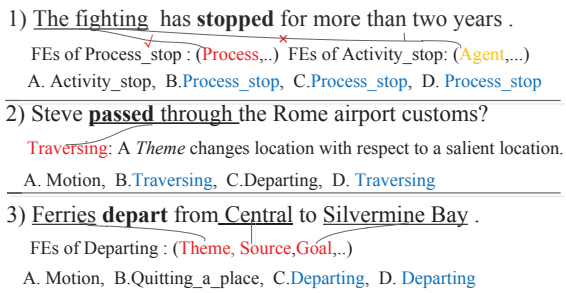

Figure 4: The case studies of our proposed models and Bert-one baseline. A, B, C and D denote the predicted frames of the following FI models: Bert-onehot, KGFI(w/ DefGCN), KGFI(w/ FEsGCN) and KGFI(w/ FrameGCN). The correct frames are marked in blue. The target words are in bold in each sentence.

comparison experiments, in which the weighted matrix is replaced with a binary matrix. Binary matrix is widely used approach to express the relations between nodes in graph modeling. Our weighting method expresses the hierarchy relationships between frames straightforwardly. The results demonstrate that the weighted method has significant impact on the model's performance, and our proposed weighting method for adjacent matrix is quite reasonable, since the performance of all the models using weighted matrix outperforms their counterparts using binary matrix, shown in Table 7.

5.4 Case Studies

Figure 4 shows that KGFI (w/FEsGCN) model tends to predict correct frame by finding the semantic relatedness between FEs and the context of target word. For instance, in sentence 1), the target word *stopped* may evoke **Activity_stop** or **Process_stop**, and the phrase *the fighting* is the key to distinguish two frames evoked by the word *stopped*, since these two frames differ in that the subject of *stopped* is an *Agent* or a *Process*. Our KGFI(w/FEsGCN) model has learned the semantic relation between *the fighting* phrase and FE *Process*, and outputs the correct frame, since FE *Agent* is related to an entity in general. The Bert-onehot model can't grasp this relation, so it outputs a wrong prediction **Activity_stop**. On the other hand, the KGFI(w/ DefGCN) model tends to predict the frame with the semantic similarities between frame definition and the sentence. For instance, in sentence 2), the word *Traversing* in definition is similar to phrase *passed through*, so the model outputs the correct frame **Traversing**.

In sentence 3), the KGFI(w/ DefGCN) model outputs a wrong prediction **Quitting_a_place** due to the similar meaning of the word *depart* in the sentence and the word *leaves* in the frame definition (*Quitting_a_place: a Self_mover leaves an initial Source location.*). The KGFI(w/ FEsGCN) model, on the other hand, has learned that the word *Ferries* in the sentence is more closely related to FE *Theme* of frame **Departing** (*Departing: a Theme moves away from a Source.*) rather than FE *self_mover* of frame **Quitting_a_place**, and outputs the correct frame **Departing**, since the *self_mover* generally refers to a living object (e.g. a person, an animal). Note that the frame **Departing** is inherited by the frame **Quitting_a_place**, so they have nearly the same FEs set except for FE *Theme* and FE *self_mover*. In other words, our KGFI(w/ DefGCN) and KGFI(w/ FEsGCN) are complementary to each other to some extent. KGFI(w/ FEsGCN) can capture the subtle differences between different frames, even if the frames have close frame-to-frame or semantic relations.

The case studies show that KGFI models can incorporate frame knowledge into its representations and guide the context encoder to learn the semantic relations between frames and the context-aware representations of target words and frames through joint learning.

6 Related work

Researchers have made great effort to tackle the FI problem since it has been proposed in the Semeval-2007 (Baker et al., 2007). It is generally regarded as a classification task. The best system (Johansson and Nugues, 2007) in the SemEval-2007 adopted SVM to learn the classifier to identify frames with a set of features, such as target lemma, target word, and so on. SEMAFOR (Das et al., 2014) uti-

lized a conditional model that shares features and weights across all targets, frames, and prototypes. These approaches use manually designed features and traditional machine learning methods to learn the classification models, while the class labels as supervision signals are discrete frame names.

Recently, distributed feature representation and models based on neural network are used to tackle FI. According to the model architecture, there are two trends of work. One is joint learning approach that converts the discrete frame labels into continuous embedding by learning the embeddings of target words and frames simultaneously. For instance, Hermann-14 (Hermann et al., 2014) implemented a model that jointly maps possible frame labels and the syntax context of target words into the same latent space using the WSABIE algorithm, and the syntax context was initialized with concatenating their word embeddings. SimpleFrameId (Hartmann et al., 2017) useed the concatenation of SentBOW (the average of embeddings of all the words in the sentence) to represent the context and then learns the common embedding space of context and frame labels following the line of (Hermann et al., 2014). The other trend is to construct the classifier model using deep neural network and regard discrete frame labels as supervision signals, which is similar to those earlier work. Open-Sesame (Swayamdipta et al., 2017) used a bidirectional LSTM to construct the FI classifier. Peng (Peng et al., 2018) proposed a joint learning model for FI and FSRL, which adopted a multitask model structure.

Different from previous studies, this paper focuses on how to represent frames by incorporating frame knowledge into frame representations and enriching frame labels with semantic information.

7 Conclusion

In this work, we propose a novel idea that leverages frame knowledge, including frame definition, frame elements and frame-to-frame relations, to improve the model performance of FI task. Our proposed KGFI framework mainly consists of a Bert-based context encoder and a GCN-based frame encoder which can effectively incorporate multiple types of frame knowledge in a unified framework and jointly map frames and target words into the same semantic space. Extensive experimental results demonstrate that all kinds of knowledge about frames are useful for enriching the representation of frames, and the better frame representation is helpful for FI task. The experimental results also show that the proposed model achieves significantly better performance than seven state-of-the-art models across two benchmark datasets.

Acknowledgments

We thank the anonymous reviewers for their helpful comments and suggestions. This work was supported by the National Natural Science Foundation of China (No.61936012, No.61772324) and the Open Project Foundation of Intelligent Information Processing Key Laboratory of Shanxi Province (No. CICIP2018007).

References

Collin F. Baker, Michael Ellsworth, and Katrin Erk. 2007. Semeval-2007 task 19: Frame semantic structure extraction. In *Proceedings of the 4th International Workshop on Semantic Evaluations, SemEval@ACL 2007, Prague, Czech Republic, June 23-24, 2007*, pages 99–104. The Association for Computer Linguistics.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, COLING-ACL '98, August 10-14, 1998, Université de Montréal, Montréal, Quebec, Canada. Proceedings of the Conference*, pages 86–90. Morgan Kaufmann Publishers / ACL.

Cosmin Adrian Bejan and Chris Hathaway. 2007. UTD-SRL: A pipeline architecture for extracting frame semantic structures. In *Proceedings of the 4th International Workshop on Semantic Evaluations, SemEval@ACL 2007, Prague, Czech Republic, June 23-24, 2007*, pages 460–463. The Association for Computer Linguistics.

Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. 2019. Multi-label image recognition with graph convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xingyi Cheng, Weidi Xu, Kunlong Chen, Shaohua Jiang, Feng Wang, Taifeng Wang, Wei Chu, and Yuan Qi. 2020. SpellGCN: Incorporating phonological and visual similarities into language models for Chinese spelling check. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 871–881, Online. Association for Computational Linguistics.

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

Dipanjan Das, Nathan Schneider, Desai Chen, and Noah A. Smith. 2010. Probabilistic frame-semantic parsing. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 2-4, 2010, Los Angeles, California, USA*, pages 948–956. The Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Charles J. Fillmore, Collin F. Baker, and Hiroaki Sato. 2002. The framenet database and software tools. In *Proceedings of the Third International Conference on Language Resources and Evaluation, LREC 2002, May 29-31, 2002, Las Palmas, Canary Islands, Spain*. European Language Resources Association.

Shaoru Guo, Yong Guan, Ru Li, Xiaoli Li, and Hongye Tan. 2020a. Incorporating syntax and frame semantics in neural network for machine reading comprehension. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2635–2641, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Shaoru Guo, Ru Li, Hongye Tan, Xiaoli Li, Yong Guan, Hongyan Zhao, and Yueping Zhang. 2020b. A frame-based sentence representation for machine reading comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 891–896. Association for Computational Linguistics.

Silvana Hartmann, Ilia Kuznetsov, Teresa Martin, and Iryna Gurevych. 2017. Out-of-domain framenet semantic role labeling. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, pages 471–482. Association for Computational Linguistics.

Karl Moritz Hermann, Dipanjan Das, Jason Weston, and Kuzman Ganchev. 2014. Semantic frame identification with distributed word representations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 1448–1458. The Association for Computer Linguistics.

Richard Johansson and Pierre Nugues. 2007. LTH: semantic structure extraction using nonprojective dependency trees. In *Proceedings of the 4th International Workshop on Semantic Evaluations, SemEval@ACL 2007, Prague, Czech Republic, June 23-24, 2007*, pages 227–230. The Association for Computer Linguistics.

Alexandre Kabbach, Corentin Ribeyre, and Aurélie Herbelot. 2018. Butterfly effects in frame semantic parsing: impact of data processing on model ranking. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 3158–3169. Association for Computational Linguistics.

Aditya Kalyanpur, Or Biran, Tom Breloff, Jennifer Chu-Carroll, Ariel Diertani, Owen Rambow, and Mark Sammons. 2020. Open-domain frame semantic parsing using transformers. *CoRR*, abs/2010.10998.

Thomas Kipf and M. Welling. 2017. Semi-supervised classification with graph convolutional networks. *ArXiv*, abs/1609.02907.

Meghana Kshirsagar, Sam Thomson, Nathan Schneider, Jaime G. Carbonell, Noah A. Smith, and Chris Dyer. 2015. Frame-semantic role labeling with heterogeneous annotations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*, pages 218–224. The Association for Computer Linguistics.

Hu Linmei, Tianchi Yang, Chuan Shi, Houye Ji, and Xiaoli Li. 2019. Heterogeneous graph attention networks for semi-supervised short text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4821–4830, Hong Kong, China. Association for Computational Linguistics.

Shulin Liu, Yubo Chen, Shizhu He, Kang Liu, and Jun Zhao. 2016. Leveraging framenet to improve automatic event detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Hao Peng, Sam Thomson, Swabha Swayamdipta, and Noah A. Smith. 2018. Learning joint semantic parsers from disjoint data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1492–1502, New Orleans, Louisiana. Association for Computational Linguistics.

Josef Ruppenhofer, Michael Ellsworth, Miriam R. L.Petruck, Christopher R. Johnson, Collin F. Baker,

and Jan Scheffczyk. 2016. Framenet ii: extended theory and practice.

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

Oscar Täckström, Kuzman Ganchev, and Dipanjan Das. 2015. Efficient inference and structured learning for semantic role labeling. *Trans. Assoc. Comput. Linguistics*, 3:29–41.

Haoran Yan, Xiaolong Jin, Xiangbin Meng, Jiafeng Guo, and Xueqi Cheng. 2019. Event detection with multi-order graph convolution and aggregated attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5766–5770, Hong Kong, China. Association for Computational Linguistics.

Hongyan Zhao, Ru Li, Xiaoli Li, and Hongye Tan. 2020. CFSRE: context-aware based on frame-semantics for distantly supervised relation extraction. *Knowl. Based Syst.*, 210:106480.