Computers and Electrical Engineering 104 (2022) 108445

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

SwitchNet: A modular neural network for adaptive relation extraction☆

Hongyin Zhu [a,b,*], Prayag Tiwari [c,*], Yazhou Zhang [d], Deepak Gupta [e], Meshal Alharbi [f], Tri Gia Nguyen [g], Shahram Dehdashti [h,i]

[a] Inspur Electronic Information Industry Co., Ltd., Jinan 250101, China
[b] State Key Laboratory of High-end Server & Storage Technology, Beijing 100085, China
[c] School of Information Technology, Halmstad University, Sweden
[d] Software Engineering College, Zhengzhou University of Light Industry, Zhengzhou 450002, China
[e] Maharaja Agrasen Institute of Technology, Delhi, India
[f] Department of Computer Science, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, P.O. Box 151, Alkharj 11942, Saudi Arabia
[g] FPT University, Danang 50509, Viet Nam
[h] School of Information Systems, Queensland University of Technology, Brisbane 4000, Australia
[i] Department of Electrical and Computer Engineering, Technical University of Munich, 80333 Munich, Germany

ARTICLE INFO

Keywords:
Relation extraction
Modular neural network
Information flow
Joint optimization
Entity pair

ABSTRACT

This paper presents a portable toolkit, SwitchNet, for extracting relations from textual input. We summarize four data protocols for relation extraction tasks, including relation classification, relation extraction, triple extraction, and distant supervision relation extraction. This neural architecture is modular, so it can take as input data at different stages of the information extraction process (simple text, text and entities or entity pairs as relation candidates) and compute the rest of the process (named entity recognition and relation classification). We systematically design four information flows to integrate the above protocols by sharing network building blocks and switching different information flows. This framework can extract multiple triples (subject, predicate, object) in one pass. This framework enhances the use of relation classification models in end-to-end triple extraction by inferring pairs of entities of interest and using the shared representation mechanism.

1. Introduction

Knowledge Graph (KG) is a technology that can store relational facts in the form of triples. For example, "The Eiffel Tower is located in Paris" can be represented as a machine-readable triple (The Eiffel Tower, locatedIn, Paris) in a knowledge graph. Knowledge graphs are designed to describe various entities or concepts and their relations that exist in the real world, and ultimately constitute a huge semantic network, in which vertices represent entities or concepts, and edges are composed of attributes or relations, thus determining the semantic relations between entities is a critical task. Relation extraction aims to find the semantic relation between entities from the textual input. This task can be used to complement missing triples in the knowledge graph.

☆ This paper is for CAEE special section VSI-hci2. Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Dr. Carlos Enrique Montenegro Marin.

* Corresponding author.
E-mail addresses: zhuhongyin@inspur.com (H. Zhu), prayag.tiwari@ieee.org (P. Tiwari), yzzhang@zzuli.edu.cn (Y. Zhang), deepakgupta@mait.ac.in (D. Gupta), Mg.alharbi@psau.edu.sa (M. Alharbi), tri@ieee.org (T.G. Nguyen), shahram.dehdashti@gmail.com (S. Dehdashti).

https://doi.org/10.1016/j.compeleceng.2022.108445
Received 2 October 2021; Received in revised form 11 October 2022; Accepted 20 October 2022
Available online 5 November 2022

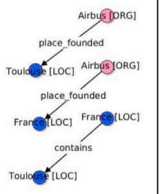
Tasks	Input	Output
Relation classification	[Airbus]ORG is based in [Toulouse]LOC, [France]LOC. (Airbus, relation, Toulouse) (Toulouse, relation, France) (Airbus, relation, France)	
Relation extraction	[Airbus]ORG is based in [Toulouse]LOC, [France]LOC. Relation PAD	
Triple extraction	Airbus is based in Toulouse, France. Entity PAD Relation PAD	

Fig. 1. Examples for text-bound RC, RE, and TE tasks.

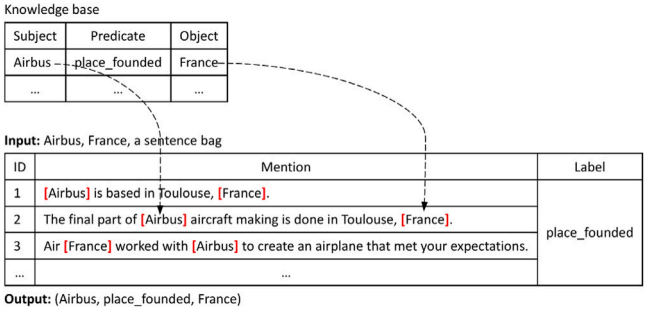


Fig. 2. Example for DS-RE.

Given the input data $\mathcal{X}$ (e.g., simple text and entity mentions), the relation extraction function $\mathcal{F}(\cdot)$ can automatically generate D (a set of triples). In practice, when the data protocol of $\mathcal{X}$ changes to $\hat{\mathcal{X}}$ (e.g., only simple text), existing model architectures cannot be easily transformed to $\hat{\mathcal{F}}(\cdot)$. Minor changes to the data protocol may lead to poor model performance and even make the model unusable. Designing different $\hat{\mathcal{F}}(\cdot)$ independently to adapt to changing data protocols is costly and inefficient. We introduce SwitchNet,[1] a portable toolkit, to resolve the problem of "reinventing the wheel". The main reason for the "wheel" problem lies in the independence and uninheritance of many studies. They implement the same relation extraction setup in different programming ways. Although the basic ideas of different works overlap, there are significant differences in programming implementation that are incompatible with each other. SwitchNet summarizes different protocols for relation extraction, divides the relation extraction process into multiple pluggable modules, and integrates different modules through neural network information flow. By choosing different neural network information flows, the model is adapted to changing data protocols.

This paper investigates the overlap and differences among four relation extraction task forms, including three text-bound tasks, i.e., relation classification (RC), relation extraction (RE) and triple extraction (TE), and the distant supervision relation extraction (DS-RE). Then, we propose a modular neural network that works towards real end-to-end relation extraction. Fig. 1 shows the input–output scheme, where the RC, RE, and TE forms provide the pairs of entity of interest (POEOI), the entity of interest (EOI), and no label respectively. EOI in the text is equivalent to the region of interest (ROI) [1] in a picture. ROI represents any particular portion of the image that seems important for the task. EOI means the important entity spans in the representation matrix of text. POEOI is composed of EOIs, and the holding relation can be inferred from this composition. POEOIs are usually two discontinuous spans, and it is uncertain whether and what kind of relation exists between the two spans.

"PAD" is a special symbol used to fill missing lengths, thus keeping the tensor dimensionality consistent, and allowing program parallelization. In addition to the benefits of PAD for parallelizing batch operations, it guarantees normal processing even if there are no entities or relations in the sample. Fig. 2 demonstrates the DS-RE task, where the input is a bag of sentences that contains the same entity pair. However, there is a noise problem in the data because not all sentences that mention the two entities express the target relation, e.g., the third sentence does not express "*place founded*" relation, so it is important to reduce the impact of wrong labeling. Traditional approaches for relation extraction rely heavily on engineered natural language processing (NLP) pipelines and external resources. Recent approaches use deep learning models to jointly extract entities and relations. These methods might need to design new task schemes or rely on syntactic parsing and separate model training. The new task schemes potentially limit the performance and versatility of their approaches. Ideally, we would like to overcome these two problems, eliminating the reliance on a new task scheme and conducting joint extraction in a modular way.

Existing joint extraction models mainly combine named entity recognition (NER) and RC from the subtask level, rarely considering the overlap and differences between relation extraction tasks. When the data protocol changes, most joint extraction

[1] https://nnntt.github.io/SwitchNet/

models cannot be easily applied to individual subtasks because innovative task schemes may not have an architecture to utilize all the input information or they may choose a simplified relation classification component. Different from them, we use the standard and effective neural components. Although task-specific neural networks have achieved good performance, it is unclear what the relationship between different RE tasks is and how to integrate them adaptively. For humans, different brain building blocks and neural pathways [2] form various cognitive functions. We propose a modular neural network that can share network building blocks by switching different information flows. We map subtasks/subfunctions to modules and use different strategies to selectively activate modules for processing information at different stages. Our main effort lies in systematically designing four information flows that are essential for model integration. In addition, not all triples in the knowledge base are static, e.g., "*place_lived*" and "*company*" relations, and our approach significantly reduces the problem complexity when dealing with multiple data sources for real-time KB updates.

The secondary problem of this framework is how to extract multiple triples in one pass in the TE protocol. This task consists of two subtasks, NER and RE. Prior works might train two subtasks separately, which leads to the drawback that the information between two subtasks cannot be fully exploited. This is because the NER and RE systems are independent of each other during model training, so they cannot be optimized together in the deep learning framework. In contrast, the differentiability of our multi-stage model is achieved through the POEOI inference and the shared representation mechanisms.

Not all context words contribute equally to the relation type, and some less informative words may produce noise. To overcome this issue, we use the attention mechanism [3] to assign different weights to different tokens for relation classification. The RC task can be further divided into single-relation extraction (SRE) and multiple-relations extraction (MRE). Our model can deal with MRE in one pass which resolves the inefficient multi-pass issue. Our model supports 1-of-n and multi-label classification forms corresponding to extracting one or different relations for two entities. Multi-label classification is a generalization of multi-class classification with no limit to the number of classes an instance can be assigned, it maps the input to a binary vector that assigns each element a label of 0 or 1, which naturally resolves the overlapping relation issue [4].

In practice, the advantage of using the proposed framework lies in its modular encapsulation of different functionalities. Task adaptation can be flexibly implemented through the combination of multiple modules and can be applied to a variety of different data protocols to meet various information extraction settings. From the perspective of theoretical research, it supports improving the model architecture in modules to achieve a partial or overall performance improvement, and can also further optimize the collaboration mechanism between different modules. We describe the usage of this toolkit in Section 5. We conduct experiments of TE, RE, RC, and DS-RE tasks on a large but noisy Riedel New York Times (NYT) dataset, a manually annotated SemEval-2018 dataset, a large TACRED dataset, and the NYT Large dataset respectively, as these datasets are well-known and represent different forms of relation extraction. Experimental results show the effectiveness of our approach in different data protocols. This model exhibits a complementary idea for the current model architecture design. In summary, the distinctive properties of this paper can be summarized below:

(i) We systematically design a modular neural network and four information flows to integrate four relation extraction tasks.

(ii) Our model integrates NER and RE subtasks through the POEOI inference and the shared representation mechanisms. We integrate DS-RE with the text-bound relation extraction protocols.

(iii) We conduct experiments on 4 relation extraction data protocols, and the experimental results show that our method can adapt to different data protocols and achieve performance improvements.

The paper is organized as follows: Section 2 presents a literature review on relation classification, joint entity relation extraction, remotely supervised entity extraction, and open information extraction. Section 3 summarizes the task definitions for the 4 relation extraction settings. Section 4 describes the proposed SwitchNet framework with novel neural network information flow and POEOI inference methods in detail. Section 5 describes the toolkit specification and design philosophy. Section 6 presents experimental results and analysis of different relation extraction task settings. Section 7 analyzes the factors that affect model performance. Finally, Section 8 presents the conclusions of this study.

2. Related work

2.1. Relation classification

For the fully-supervised methods, Zeng et al. [5] leverage convolutional deep neural networks to extract lexical and sentence-level features, and to specify entity pairs that should be assigned relation labels, they propose position features to encode the relative distances of words to target entity pairs in the convolutional neural network. Experimental results show that position features are crucial for relation classification. Xu et al. [6] propose a new neural network model, called long short term memory networks along shortest dependency paths (SDP-LSTM), for relation classification. Neural models are necessary to process information in a direction-sensitive manner, and they divide the SDP into two sub-paths, each from an entity to a common ancestor node. The model concatenates the features extracted along the two sub-paths for final classification. A highlight feature of this model is that the shortest dependency path preserves the most relevant information while eliminating irrelevant words in sentences; the multi-channel long short term memory networks (LSTM) network allows for efficient integration of information from heterogeneous sources through dependency paths. They utilize LSTM cells for information dissemination and integration.

2.2. Joint entity-relation extraction

Miwa and Bansal [7] propose an end-to-end model to extract relations between word sequences and entities on dependency structures. Models based on recurrent neural networks capture word sequences and relying on tree substructure information, entities and relations can be represented simultaneously in a single model. The model first detects entities, then extracts relations between detected entities using a single incrementally decoded neural network structure, and jointly updates model parameters with entity and relation labels. This allows models to collectively represent entities and relations within a single model using shared parameters. This model detects entities during training and uses entity information in relation extraction through entity pre-training and scheduled sampling which replaces the predicted labels with gold labels with a certain probability. Zheng et al. [4] propose a new tag scheme that can transform the joint entity-relation extraction task into a sequence labeling problem. Based on this tagging scheme, they investigated different end-to-end models to directly extract entities and their relations without identifying entities and relations separately. They also developed an end-to-end model with a bias loss function to adapt to new labels. This method can enhance the associations between related entities but still falls short in identifying overlapping relations. Zhang et al. [8] propose a graph convolutional network-based neural model for relation extraction, an extension of graph convolutional networks tailored for relation extraction, which can efficiently aggregate information of arbitrary structures in parallel. They propose path-centric pruning to improve the robustness of dependency models by removing irrelevant content without ignoring key information. To incorporate relevant information while maximally removing irrelevant content, they further applied a new pruning strategy to the input tree, allowing words to immediately surround the shortest path between two entities that may be related. Sun et al. [9] propose a graph convolutional network (GCN) based joint model to perform joint type inference for entity-relation extraction tasks. A binary relation classification task is introduced to explore the structure of entity relation bipartite graphs in a more efficient and interpretable manner. To address the joint type inference task, a novel GCN operation on entity-relation bipartite graphs is proposed. Compared with existing joint extraction methods, it provides a new way to explicitly capture interactions of multiple entity types and relation types in a sentence. By introducing the task of binary relation classification, the structure of entity-relation bipartite graphs can be exploited in a more efficient and interpretable way. Wang et al. [10] propose a scheme to simultaneously extract multiple relations and encode the input passages of the MRE task once. They let the self-attention layer know the positions of all entities in the input paragraph, building on a pre-trained self-attention model (Transformer). A structured prediction and entity-aware self-attention layer are proposed on top of BERT. Since all relations are computed at once, it can easily scale to larger datasets.

Bekoulis et al. [11] propose a joint neural model to simultaneously extract entities and relations from textual data. They model the entity recognition task using a conditional random field (CRF) layer, and model the relation extraction task as a multi-head selection problem (i.e., potentially identifying multiple relations for each entity). State-of-the-art performance is achieved in different domains (i.e. news, biomedical, real estate) and languages (i.e. English, Dutch) without relying on any human-crafted features or additional NLP tools. The downside is that this method uses a token to represent an entity. Han et al. [12] propose OpenNRE, an open and extensible toolkit for relation extraction. OpenNRE strikes a balance between system encapsulation, operational efficiency, model scalability, and ease of use. The toolkit prioritizes operational efficiency based on TensorFlow and PyTorch which support fast model training and validation. OpenNRE provides various functional RE modules based on TensorFlow and PyTorch to maintain sufficient modularity and extensibility so that new models can be easily incorporated into the framework. Online systems can also be used to satisfy real-time extraction without training and deployment. For developers whose goal is to train a custom model, they can quickly spin up an OpenNRE-based RE system without knowing too much technical detail and writing tedious glue code. This toolkit did not integrate different components in one model. Our model integrates different modules of the information extraction (IE) process, can jointly optimize the parameters of subtasks, and can extract multiple triples in one pass.

2.3. Distant supervision relation extraction

For the distantly-supervised methods [13], a training set is generated by aligning the relations in an existing knowledge base onto the free text. They address the problem of data annotation. Mintz et al. [13] propose a distant supervision algorithm capable of extracting high-precision patterns for a large number of relations. The combination of grammatical and lexical features provides better performance than separate feature sets. Their algorithms can use large amounts of unlabeled data, and a pair of entities may appear multiple times in the test set. Their experiments use Freebase, a large semantic database of thousands of relations, to provide distant supervision. Their algorithm combines the advantages of supervised IE (combining 400,000 noise pattern features in a probabilistic classifier) and unsupervised IE (extracting large numbers of relations from large corpora in any domain). Existing methods [14] adopt multi-instance learning to alleviate the wrong labeling problem. However, the above methods rely heavily on the quality of featuring engineering. Zeng et al. propose the piece-wise max-pooling strategy over multi-instance learning. This approach can automatically learn features through deep learning models. Lin et al. [15] propose sentence-level attention to alleviate the wrong labeling problem, but the sentence representation needs to be further improved. Reinforcement learning (RL) is also applied to enhance the instance selection process by proposing innovative schemes for the episode, state, action, reward, and optimization, but the stability and computational efficiency of the model are challenges. The DS-RE methods usually compare the precision and recall curves and often put precision before the recall, while the supervised methods obtain determined scores. Our research shows the relationship between text-bound RE and DS-RE.

2.4. Open information extraction

For the Open IE [16], it aims to generate triples from textual input without requiring a pre-specified vocabulary. The subjects and objects extracted by Open IE are fragments in the text, not necessarily entities, such as values, dates, prices, etc. The predicates are also freer than the predicates of our triple extraction task. Open relation extraction methods can be divided into two categories: (1) Relation extraction using explicit rules. Relation extraction is performed by building a combination of methods including dependency parsing, part-of-speech tagging, entity linking, and rule templates. (2) Relation extraction using implicit rules. Supervised training of neural network models on a large human-annotated corpus can learn implicit rules for general triple extraction. However, such methods may extract some meaningless triples and require further filtering and completion of the results.

3. Task definition

Let $\mathcal{X} = [w_1, \ldots, w_n]$ denote a sequence of text, where w_i is the ith token. A subject entity s and an object entity o denote two non-overlapping consecutive spans: $\mathcal{X}_s = [w_{s1}, w_{s1+1} ..., w_{s2}]$ and $\mathcal{X}_o = [w_{o1}, w_{o1+1} ..., w_{o2}]$. R is the set of predefined relations. We formalize the RC, RE, TE, and DS-RE tasks as follows.

- **RC** Given the textual input $\mathcal{X}$ and the positions of all (s, o) pairs, the goal is to predict the corresponding relation $r \in \mathcal{R}$ that holds between each (s, o) pair or no relation otherwise.
- **RE** Given the textual input $\mathcal{X}$ and the positions of all entity spans $\mathcal{E}_e = \{[w_{e11}, w_{e11+1}, \ldots, w_{e12}], \ldots, [w_{ek1}, w_{ek1+1}, \ldots, w_{ek2}]\}$, the goal is to predict the positions of all (s, o) pairs and the relation $r \in \mathcal{R}$ for each (s, o).
- **TE** It only takes as input text $\mathcal{X}$, and the goal is the same as the RE task. This task is equivalent to NER plus RE.
- **DS-RE** It takes as input two entities $(e1, e2)$ and a bag of sentences (C) that mention $e1$ and $e2$ and predict the relation $r \in \mathcal{R}$ (sometimes multiple-relations) that holds between $e1$ and $e2$, or no relation otherwise. This method assumes that any of the relations is supported by at least one sentence in C. Using distant supervision, relations in KBs can be aligned with plain text to produce bags of relation mentions for model training.

4. Approach

Fig. 3 shows the architecture of SwitchNet. This framework mainly contains five modules, i.e., the encoder, the NER module, the RC module, the POEOI inference module, and the DS-RE module. These modules are connected by the designed information flows. Different information flows correspond to different input data protocols. The input mainly consists of three elements, including simple text and two optional elements, the entity annotation (EOI) or the relation annotation (POEOI). In the following, we describe the model architecture, the information flows, and the training methods.

4.1. Model architecture

We divide the model into different modules and then integrate the modules together through neural network information flow to achieve specific functions. In this section, we first introduce the working mechanism of neural network information flow and then introduce the composition of each module.

4.1.1. Neural network information flows

Different tasks can be achieved by switching different information flows. We will describe the mechanism of these information flows to implement adaptive relation extraction, as shown in Fig. 3. We explain the functionality, the location, and the collaboration of the modules in various relation extraction pathways. By controlling the information flow, this model can selectively activate specific operations and features at different stages of information extraction. We use $\mathcal{P}ij$ to indicate the information flow, that is, the functional connection between modules in this neural network. Some information flows may contain specific data operations. We describe the information flows as follows.

(1) When we process the RC task, this model activates the pathway $[\mathcal{P}11, \mathcal{P}12] \rightarrow \mathcal{P}13$, as detailed in Section 4.1.4. The entity relation annotation directly indicates the POEOI through $\mathcal{P}11$. Then, the input to the RC layer comes from the path $\mathcal{P}12 \rightarrow \mathcal{P}13$ which connects the contextual representation and the POEOI modeling directly.

(2) When we process the RE task, this model activates the pathway $[\mathcal{P}21, \mathcal{P}22] \rightarrow \mathcal{P}23$, as detailed in Section 4.1.5. The input provides the pre-annotated EOIs through $\mathcal{P}21$, and then the function of $\Psi(\cdot)$ activates $\mathcal{P}23$ via the generated pair-wise combination of EOIs (POEOIs). Then, the path $\mathcal{P}22 \rightarrow \mathcal{P}23$ connects the contextual representation and the POEOI modeling.

(3) When we process the TE task, this model uses the pathway $\mathcal{P}31 \rightarrow [\mathcal{P}32, \mathcal{P}34] \rightarrow \mathcal{P}33$, as detailed in Section 4.1.5. This model first predicts/infers EOIs via $\mathcal{P}31$ which contains the function of $\Phi(\mathcal{F}_{ner}(\cdot))$. Then, the path $\mathcal{P}32 \rightarrow \mathcal{P}33$ indirectly generates POEOIs by the function of $\Psi(\cdot)$. Then, the path $\mathcal{P}34 \rightarrow \mathcal{P}33$ connects contextual representation and the POEOIs.

(4) When we process the DS-RE task, this model uses the pathway *Distant supervision*$\rightarrow \{\mathcal{P}40 \rightarrow [\mathcal{P}41, \mathcal{P}42] \rightarrow \mathcal{P}43\} \overset{\times k}{\rightarrow} \mathcal{P}44$, as detailed in Section 4.1.6. Suppose there is a knowledge base, the distant supervision technique is used to automatically align the relations in KBs to free text. Each relation will be mapped to a bag of sentences. Then the model takes as input each sentence

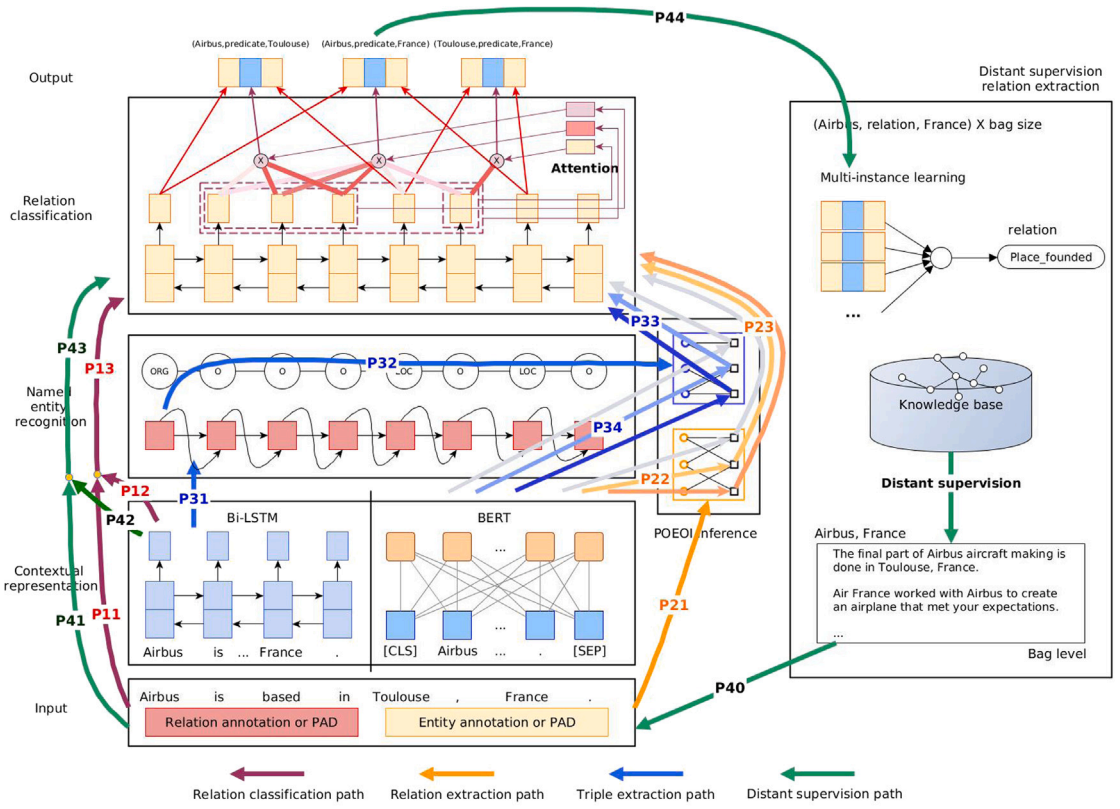


Fig. 3. Neural network architecture overview.

bag via path $\mathcal{P}40$. For each sentence, the information flow that generates the sentence representation is similar to the RC task, i.e., $[\mathcal{P}41, \mathcal{P}42] \rightarrow \mathcal{P}43$. Each sentence only generates one vector. $\overset{\times k}{\rightarrow}$ means this model parallel process k sentences in a bag. Then the model uses MIL to compose a vector representation of the bag.

In the above four pathways, our network adaptively models relation representation based on the POEOIs. The difference among the above RE pathways is whether there are EOI inference and POEOI inference to predict the positions of EOIs and POEOIs as relation candidates during model training.

4.1.2. Encoder

This module encodes the textual input into contextual representation. The neural network can be bidirectional LSTM (Bi-LSTM) or pre-trained language models (e.g., BERT) [17]. Let $[w_1, w_2, \ldots, w_m]$ be the sequence of text with m tokens. The encoder generates the hidden states $[h_1, h_2, \ldots, h_m]$ which can be used as the shared representation for different modules, where $h_i = Encoder(w_i)$. When we use the Bi-LSTM encoder, the $h_i = \overrightarrow{h}_i \oplus \overleftarrow{h}_i$ is generated by the following equation.

$$\overrightarrow{h}_i = \text{LSTM}(\overrightarrow{h}_{i-1}, x_i) \tag{1}$$

$$\overleftarrow{h}_i = \text{LSTM}(\overleftarrow{h}_{i-1}, x_i) \tag{2}$$

where $\overrightarrow{h}_0$ and $\overleftarrow{h}_0$ are initialized to zero vectors. $x_i = v_i \oplus t_i$ is the word representation where $\oplus$ is the concatenation operation. The word representation is composed of the word embedding and character-level representation t_i encoded by a convolutional neural network (CNN). Word embeddings [18] refer to the technique of encoding the meaning of words in the form of low-dimensional dense vectors, so that words that are closer in the vector space are also semantically similar. Character-level embeddings [19] learn vector representations of characters, which are then aggregated through a 1D convolutional neural network or LSTM neural network to generate an overall representation of the word.

Pre-trained language models are first pre-trained on a large-scale corpus based on self-supervised tasks and then fine-tuned for downstream tasks. BERT is pre-trained with the masked language model task and the next sentence prediction task. When we use the BERT as an encoder, [CLS] is a special symbol added in front of each input sample, the hidden state corresponding to this token is used as the aggregated sequence representation for the classification task, [SEP] is a special separator token used to separate discontinuous token sequences (e.g., [CLS] question [SEP] answer [SEP]). An input sentence is denoted as

$[[\text{CLS}], w_1, w_2, \ldots, w_m, [\text{SEP}]]$. BERT consists of multiple layers of transformer encoders that generate contextual representations through the multi-head self-attention mechanism.

$$[h_{cls}, h_1, h_2, \ldots, h_m, h_{sep}] = \mathcal{F}_{bert}([[\text{CLS}], w_1, w_2, \ldots, w_m, [\text{SEP}]]) \tag{3}$$

where $\mathcal{F}_{bert}(\cdot)$ denotes the architecture defined in BERT [17]. We only use the hidden representation of each token $h_i (i = 1, 2, \ldots, m)$. The core idea of the transformer block is a self-attention mechanism through which contextual representations can be generated, as shown in Eq. (4).

$$h_{att} = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{4}$$

The above equation shows the scaled dot-product attention where Q, K, and V are the query, key, and value matrices. The input consists of queries and keys of dimension d_k, and values of dimension d_v.

4.1.3. Named entity recognition

This module aims to discover the EOIs from simple text. BiLSTM-CRF [20] model achieves good performance, and this task can be modeled as finding an optimal label sequence $\mathbf{y}^*$ that maximizes the conditional probability $p(\mathbf{y}|\mathbf{h}; \theta)$ as shown below.

$$\mathbf{y}^* = \arg \max_{\mathbf{y} \in \mathcal{Y}(\mathbf{h})} p(\mathbf{y}|\mathbf{h}; \theta) \tag{5}$$

where θ is the model parameter. $\mathbf{h} = [h_1, \ldots, h_m]$ is the output sequence of the encoder and $\mathbf{y} = [y_1, \ldots, y_m]$ is the label sequence. $\mathcal{Y}(\mathbf{h})$ denotes the set of possible label sequences for $\mathbf{h}$.

$$p(y|\mathbf{h}; W, b) = \frac{\prod_{i=1}^{n} \exp(W^T_{y_{i-1} y_i} h_i + b_{y_{i-1} y_i})}{\sum_{y' \in \mathcal{Y}(h)} \prod_{i=1}^{n} \exp(W^T_{y'_{i-1} y'_i} h_i + b_{y'_{i-1} y'_i})} \tag{6}$$

where $\{[h_i, y_i]\}, i = 1, 2...n$ represents the vector of the ith word h_i and the ith label y_i in the input sequence respectively. $\mathcal{Y}(\mathbf{h})$ denotes all the possible label sequences for the input sequence $\mathbf{h}$. W and b are weight matrix and bias vector, in which W_{y_{i-1}, y_i} and b_{y_{i-1}, y_i} are the weight vector and bias corresponding to the successive labels (y_{i-1}, y_i). $p(y|\mathbf{h}; W, b)$ is the probability of generating this tag sequence over all possible tag sequences.

LSTM-based representations and linear layers can be directly used for class labeling of words, and we introduce the following improvements to this method. In addition to the current input, the LSTM-RNN takes the hidden state of the previous step as input. $\overrightarrow{h}_i^{(ner)} = \text{LSTM}(\overrightarrow{h}_{i-1}^{(ner)}, h_i \oplus \overrightarrow{h}_{i-1}^{(ner)})$ where $\overrightarrow{h}_i^{(ner)}$ is the representation of the ith token in the NER module. The superscript (ner) is used for the notation of the NER module. $\overrightarrow{h}_0^{(ner)}$ is initialized to a zero vector. This representation is used for classification.

$$p(y_i|y_{1:i-1}) = softmax(W \overrightarrow{h}_i^{(ner)} + b) \tag{7}$$

where W and b are weight and bias parameters.

4.1.4. Relation classification

Single-relation extraction For SRE, suppose we have the position of a POEOI (s and o), this module generates the triple vector $v = s^{(rc)} \oplus h_r \oplus o^{(rc)}$, where the superscript (rc) denotes the RC module. The subject $s^{(rc)} \in \mathbb{R}^\mu$ or object $o^{(rc)} \in \mathbb{R}^\mu$ is represented by the average/max pooling of the corresponding hidden states $\mathbf{h}^{(rc)}$. The hidden states are encoded by a Bi-LSTM.

$$\overrightarrow{h}_i^{(rc)} = \text{LSTM}(\overrightarrow{h}_{i-1}^{(rc)}, h_i) \tag{8}$$

$$\overleftarrow{h}_i^{(rc)} = \text{LSTM}(\overleftarrow{h}_{i-1}^{(rc)}, h_i) \tag{9}$$

where $h_i^{(rc)} = \overrightarrow{h}_i^{(rc)} \oplus \overleftarrow{h}_i^{(rc)}$. The predicate h_r is calculated by the attention mechanism. Let $\mathcal{F}_{att} : \mathbb{R}^{n \times \mu} \rightarrow \mathbb{R}^\mu$ denote an attention function that maps n input vectors to a relation vector h_r. μ is the vector dimension. $h_0^{(rc)}$ is initialized to a zero vector.

We introduce the POEOI modeling and the position-aware POEOI modeling. For the POEOI modeling, the most relevant portion of text to determine the relation type is usually the one contained between and including the entities [21]. Not all words contribute equally to the relation type, so we use the weighted aggregation of informative hidden states between two entities to represent the relation. We use a self-attention method [22] to extract informative tokens. This method first captures the main context information of relation by the average-pooling of hidden states as shown in Eq. (10).

$$q = \frac{1}{c_2 - c_1} \sum_{i=c_1}^{c_2} W_Q h_i^{(rc)} \tag{10}$$

where $W_Q \in \mathbb{R}^{\mu \times \mu}$ is model parameter. $h_i^{(rc)} \in \mathbb{R}^\mu$ is the ith hidden state of the relation context. c_1 and $c_2 \in [1, m]$ are the start and end indices of relation context respectively. Then, this method measures the token weights by considering their importance in the context and obtains the normalized importance weights through a softmax function.

$$\alpha_i = \frac{\exp(q^T W_K h_i^{(rc)})}{\sum_{j=c_1}^{c_2} \exp(q^T W_K h_j^{(rc)})} \tag{11}$$

where $h_r \in \mathbb{R}^{\mu}$ is the relation representation. $W_K \in \mathbb{R}^{\mu \times \mu}$ are model parameters.

The relation representation is obtained by calculating the weighted sum of the hidden states. The self-attention mechanism outputs as many vectors as it inputs as Eq. (4). To obtain the representation of text span, inspired by Hierarchical attention networks (HAN) [23], we use a variant of the self-attention mechanism, where the output vector is a weighted aggregation of contextual representations.

$$h_r = \sum_{i=c_1}^{c_2} \alpha_i \cdot W_V h_i^{(rc)} \tag{12}$$

where $W_V \in \mathbb{R}^{\mu \times \mu}$ are model parameters. This is also known as the single-head attention which is highly extensible, because adding more heads will generate multi-head attention.[2]

$$h_r = h_{r,1} \oplus h_{r,2} \ldots \oplus h_{r,n} \tag{13}$$

where n is the number of heads. $h_{r,i}$ represents the ith head.

For the position-aware POEOI modeling, the input to RC module becomes $h_i^{(s,o)} = h_i \oplus p_i^{(s)} \oplus p_i^{(o)}$ where (s, o) is a pair of entities. p_i is the position embedding. Position embedding [5] is a way to encode relative position from current word to s and o. Position embedding is a technique for representing the spatial positional relationship of words in a sentence. Relation classification is a complex task, and structural features in sentences are important, such as the spatial location relationship between context words and entities. Such structural information cannot be captured by word embeddings alone, so position embeddings are needed to obtain spatial positions. For example, as shown in Fig. 1, the relative position of "based" to "Airbus" (s) and "France" (o) are -2 and 4 respectively. Then, the relative position is encoded using a fixed-length vector that is randomly initialized and jointly trained. This feature explicitly encodes the position of POEOI, so it allows the model to consider the entire token sequence. The single-head position-aware POEOI modeling is similar to the position-aware attention LSTM (PA-LSTM) model to some extent. The main difference is that our model can perform MRE in one pass.

This module supports the above two methods. The former is more computationally efficient but may lose some information. The latter can consider more contexts, but add more computation. The two methods achieve comparable results in the TE and RE tasks because the predicted POEOIs is not always correct. Using position embedding in these tasks may increase noise. When the input indicates the POEOI, we prefer to use the latter method because this allows the model to consider more context information. Finally, the triple vector is input into a single layer neural network for classification.

$$p(y^{(rc)}) = softmax(W^{(rc)} v + b^{(rc)}) \tag{14}$$

where $W^{(rc)} \in \mathbb{R}^{\tau \times \rho}$ and $b^{(rc)} \in \mathbb{R}^{\tau}$ are weight and bias parameters. ρ and τ are the input dimension and class number respectively.

Multiple-relations extraction For MRE, suppose we have multiple POEOIs. We use the above process to model each triple and stack them to generate the triple matrix A, i.e., $[v_1, \ldots, v_u]$ where u is the number of triple candidates. Then the model classifies all relation candidates through a softmax function on columns.

$$p(Y^{(rc)}) = softmax(W^{(rc)} A + b^{(rc)}) \tag{15}$$

where $W^{(rc)} \in \mathbb{R}^{\tau \times \rho}$ and $b^{(rc)} \in \mathbb{R}^{\tau}$ are weight and bias parameters. This setting allows the model to update parameters from more perspectives, and each sample is encoded only once.

4.1.5. Inference for pairs of entities of interest

When the POEOI is uncertain, each entity phrase can be a subject or an object, so we consider the relation type and direction or no relation. The output of NER subtask is completely different from the input of RE subtask. We propose the POEOI inference scheme to integrate NER and RE subtasks in the joint optimization process by predicting the POEOIs as below.

$$A = \mathcal{F}_{rc}(\mathbf{h}, \Psi(\Phi(\mathcal{F}_{ner}(\mathbf{h})))) \tag{16}$$

where A is the triple matrix. The inference of this framework means "prediction" instead of traditional pipeline reasoning. Each triple can be predicted as a specific relation type (including relation direction) or NA. $\mathcal{F}_{ner}(\cdot)$ is the NER function that predicts the label sequence. $\Phi(\cdot)$ denotes the function that identifies the EOIs. This can be seen as an action selection process for discovering EOIs. $\Psi(\cdot)$ is the function that infers POEOIs as relation candidates. The POEOIs are generated by using all possible pair-wise combinations of EOIs. POEOI inference is an operation that generates the pair-wise entity combination. For example, for the three entities A, B, and C, after this operation, three combinations of AB, AC, and BC will be formed. For n entities, C_n^2 combinations will be generated.

$$\Psi(\{A, B, C\}) = \{AB, AC, BC\} \tag{17}$$

The different number of relation candidates leads to different learning speed and performance. Extensive experiments will be discussed in Section 7.2. $\mathcal{F}_{rc}(\cdot)$ denotes the function of the relation classification module that encodes triple representation. Eq. (15) can be seen as an action selection process for predicting the relation type. $\mathcal{F}_{ner}(\cdot)$ and $\mathcal{F}_{rc}(\cdot)$ are cascaded subtasks, and we connect

[2] Multi-head attention performs nearly the same as the single-head attention in text-bound RE. For DS-RE, it achieves significant improvement.

them using the POEOI inference and the shared representation mechanisms. The inference function $\Psi(\Phi(\mathcal{F}_{ner}(\cdot)))$ generates a heuristic to add regularization to the network, similar to the effect of dropout operations. This heuristic selects and emphasizes the locally optimal features [24]. However, the function $\Psi(\cdot)$ is a discrete action selection process, so there is a sort of information loss under the gradient-based optimization framework. It is difficult to solve the discrete optimization problems [25] using Stochastic gradient descent (SGD) like optimization algorithms. Other non-SGD optimization mechanisms (e.g., genetic algorithms, ranking mechanisms, etc.) can be explored to augment the model optimization.

We integrate the process of discovering EOIs and modeling POEOIs. We optimize the model at the end of the forward pass. The triple matrix contains the memory of all possible relation candidates. Finally, we predict the relations through Eq. (15). This method works towards minimizing the loss function based on the ground truth, so it is supervised learning with inference. The POEOI inference scheme makes the RC module widely compatible with other SRE models, e.g., PCNN and the PA-LSTM. This expansion enhances the use of existing SRE/MRE methods to make them conduct joint entity and relation extraction.

We introduce the techniques for controlling the ratio of candidate relations in detail in Section 7.2, and this paper does not control the ratio of entities, and related work can be further studied in the future. There are two ways to implement batch processing. (1) Favorable way for program parallelization. It is to first determine the coordinates of each entity, then generate a combination of coordinates, and obtain slices of tensors according to the coordinates (for example, using functions such as torch.index_select() or tensorflow.gather()), which is realized by carefully designed tensor dimension expansion and reshape. (2) Unfavorable way of program parallelization. This method is relatively simple and is implemented through a "for loop". The entity vector set of each sample is processed separately, combined according to the entity pair, and finally all the relation candidate vectors are concatenated, and then the gradient of the batch samples is accumulated by gradient accumulation. To reduce the computational complexity, we abandon the POEOIs that have a distance longer than 50 words.

4.1.6. Distant supervision relation extraction

The distant supervision assumption is that if two entities preserve a relation in a KB, then all sentences that mention the two entities express this relation. This assumption improves the efficiency of automatic data annotation but introduces noise into the training data. In order to alleviate the wrong labeling problem, multi-instance learning (MIL) is a reasonable choice for DS-RE. We further describe how to expand the RC information flow to the DS-RE task.

Suppose there is a bag C containing n sentences, each of which mentions $e1$ and $e2$, i.e. $C = \{s_1, s_2, \ldots, s_n\}$. This model takes as input C and generates the representation $V = \{A_1, A_2, \ldots, A_n\}$. Note that each A_i contains only one POEOI. This task can be seen as n parallel RC processes. Sentence-level attention approach [15] is reasonable and achieves high performance to de-emphasize the noisy sentences, so we use this method as the base. This method aims to calculate the bag representation by weighted aggregation of sentences as Eq. (18). The bag representation is derived from all sentences that mention the two entities.

$$c = \sum_i \beta_i A_i \tag{18}$$

where β_i is the weight of each sentence vector A_i as shown in Eq. (19). This weight is calculated by the softmax function. A_i is calculated by the word-level attention mechanism in the RC module.

$$\beta_i = \frac{\exp(e_i)}{\sum_k \exp(e_k)} \tag{19}$$

where e_i is calculated by Eq. (20) which reflects how well the input sentence s_i matches the relation r. We use the bilinear form as follows.

$$e_i = A_i^T M \mathbf{r} \tag{20}$$

where M is a weighted diagonal matrix, and $\mathbf{r}$ is the query vector associated with a relation r. $\mathbf{o}$ is the output of this module and it represents the scores for all relation types, as shown below.

$$\mathbf{o} = Oc + b^{(ds)} \tag{21}$$

where O is the representation matrix of relations. $b^{(ds)}$ is a bias vector. Finally, we compute the conditional probability of $p(y_i|C;\theta)$ through a softmax function.

$$p(y_i|C;\theta) = \frac{\exp(o_i)}{\sum_{j=1}^{k} \exp(o_j)} \tag{22}$$

where k is the total number of all relations. In the RC module, the single/multi-head attention first extracts informative tokens for instance representation. Then, the MIL selects high-quality instances for bag representation. The entire process constructs hierarchical attention.

4.2. Training methods

Different tasks require different training objectives because the modules have different collaborative methods in various relation extraction pathways. This framework needs to optimize the model to adapt to the corresponding tasks. This framework supports all

the training objectives for different tasks. For the TE and RE tasks, we can optimize the combined objective function at the end of the forward pass.

$$\min_{\Theta} \mathcal{L} = \sum_{i=1}^{|\mathbb{D}|} (\mathcal{L}_i^{(ner)} + \lambda \mathcal{L}_i^{(re)}) \tag{23}$$

where $\mathcal{L}_i^{(ner)}$ and $\mathcal{L}_i^{(re)}$ are the objectives of two subtasks respectively. $|\mathbb{D}|$ is the dataset size. Θ is the model parameter. λ is a hyper-parameter to weight the influence of RE subtask. The training loss is to sum the deviation of the relation prediction and the deviation of entity prediction. We adopt the negative log-likelihood as the objective, as shown below.

$$\mathcal{L}^{(ner)} = -\sum_{j} \log p^{(ner)}(y_j^{(ner)}|\mathbf{x};\Theta) \tag{24}$$

$$\mathcal{L}^{(re)} = -\sum_{k} \log p^{(re)}(y_j^{(re)}|\mathbf{x};\Theta) \tag{25}$$

where $p^{(ner)}(\cdot)$ and $p^{(re)}(\cdot)$ represent the probabilities of true class of NER and RE subtasks respectively. $\mathbf{x}$ denotes the input sequence. j and k denote the indices of entity and relation respectively. We can train the two tasks jointly, and they interact with each other through the encoder. The POEOI inference and the shared representation mechanisms enable the cascaded subtasks to learn together.

By learning the multi-label classification model, we can also predict different relations between two entities. This model will perform binary classification for each relation type separately. The objective of $\mathcal{L}^{(re)}$ becomes the log loss in Eq. (26).

$$\mathcal{L}^{(re)} = -\sum_{k}\sum_{c} y_{k,c}^{(re)} \log p^{(re)}(y_{k,c}^{(re)}|\mathbf{x};\Theta) + (1 - y_{k,c}^{(re)}) \log(1 - p^{(re)}(y_{k,c}^{(re)}|\mathbf{x};\Theta)) \tag{26}$$

where k and c are the triple index and class index respectively. The output gives a probability distribution over all labels, and a threshold is used to determine the final subset of labels.

For RC task, we already know the POEOIs in the sentence, so we can model POEOIs directly. The training objective is shown blew.

$$\min_{\Theta} \mathcal{L} = \sum_{i}^{|\mathbb{D}|} \mathcal{L}_i^{(re)} \tag{27}$$

For DS-RE task, each sentence bag is used to model one POEOI. The training objective is shown blew.

$$\min_{\Theta} \mathcal{L} = -\sum_{i}^{|\mathbb{K}|} \log p(y_i^{(re)}|C_i;\Theta) \tag{28}$$

where $|\mathbb{K}|$ is the number of sentence bags. C_i is a bag of sentences. By selecting different information flows and corresponding training objectives, we can train different functional neural networks for relation extraction tasks.

The relation extraction task can be viewed as solving a mathematical problem. Indicating all entities in advance can reduce the number of independent variables, making the task more deterministic, which is also in line with the laws of solving mathematical problems. The final result is obtained by continuously solving the unknowns. In cases where entities are not indicated, intermediate results need to be inferred from existing information during model training, which can lead to noise. The limitation of this framework is that first of all, the programming of the whole framework is very difficult. Although this version implements basic functions, there is still a large room for optimization in performance. It can be achieved by using better design patterns and programming tricks. When the framework cannot extract entities and contains nested entities, the relation extraction module will be deactivated. A low-quality dataset makes the model learn the wrong rules. Noise in the training set interferes with the model getting a high-quality representation. These noises are harmful, but the model itself is unaware of these mistakes. For relation extraction tasks, it relies on hidden state representations of deep learning models, so high-quality representations are critical. Training a model on low-quality data can cause the model to fail to track the cause of the error, thereby exacerbating the black-box nature of the model.

Integrating existing models is an important function, and in fact, we provide a new perspective, rethinking the contributions of previous research. By reorganizing and improving previous methods, making them suitable for new relation extraction tasks. The data format is a very important issue, especially when agents want to achieve big data intelligence, they need to face various changes and uncertainties in the network environment. When faced with special data that needs to be processed, a single approach will fail, allowing the agent to adapt to a variety of complex data. The work of this paper is an effort to solve this important problem. Joint extraction methods often change the form of the task, which leads developers to lose sight of a standardized entity recognition and relation extraction pipeline. For the two-stage relation extraction pipeline, although the standardized entity recognition and relation extraction pipelines are retained, it may be difficult to jointly optimize the subtasks. Few methods can clearly conform to standardized named entity recognition and relation extraction pipelines while enabling joint optimization of subtasks, and our method addresses this problem. This model is potential to be expanded to consider the longer context in other tasks, e.g., coreference resolution and document-level relation extraction.

5. Toolkit instructions

The working mode of this toolkit is determined by the data protocol. This toolkit can automatically switch to the corresponding task state according to the format of the input data without additional configuration. The input data is stored in JSON format, and

Table 1
Description of input data fields.

Fields	Description
token	Token sequence of text.
pos	Part-of-speech tagging sequence of text.
dep	Dependency parsing tree of text, including the edge types.
relation	Ground truth triple labels, expressed in the form of $[(relation_k, (e_i^{start}, e_i^{end}), (e_j^{start}, e_j^{end})), \dots]$
ner	Ground truth NER label sequences of text.
ner-feature	NER tags extracted by using other toolkits such as Stanford CoreNLP, Spacy, etc.
text	The original input text.
relation-set	List of all predefined relation types.
entity-set	List of all predefined entity types.

the main data fields include "`token, pos, dep, relation, ner, ner-feature, text, relation-set, entity-set`". The description of these fields is shown in Table 1. We chose the JSON format because it is a common data transfer format for Internet data, including the response data returned after an agent requests a service.

The working mode is mainly determined by `ner` and `relation` fields. For model prediction, when the `ner` field is `None`, it means that the data protocol does not provide entity tags, so it is a TE task. When the `ner` field contains entity labels, the RE or RC tasks will be executed. When the `ner` labels contain only two entities, the model works in the RC mode. When the `ner` labels contain three or more entities, the model works in the RE mode. When the input data fields contain `head` and `tail`, the model works in the DS-RE mode. For model training, when the `ner` labels contain three or more entities, the model is trained in the TE mode.

In the model configuration, the main parameters related to word vectors are `word_representation` and `fine_tune`. The `word_representation` parameter can be set to the path of the context-independent word vectors, e.g. GloVe, FastText, etc. or the path of the BERT model. Setting the `fine_tune` parameter to True means that the word vectors or the BERT will be fine-tuned during the training process, and vice versa. In addition, we can choose whether to use the CRF model to decode NER results. The `head_number` parameter can be used to set the number of heads used for relation representation. For the DS-RE model, we mainly use the OpenNRE [12] framework as the baseline and integrate our sentence-level multi-head relation representation.

6. Experiments

6.1. Experimental settings

6.1.1. Dataset

NYT dataset[3] is developed by [14] by aligning Freebase relations with New York Times news articles. Provided for public access from March 2007, Freebase[4] is an open, shared world knowledge sharing database, which consists of a large number of collaboratively edited cross-linked data (currently 1.9 billion triples). Freebase is constructed by extracting entities and relations from knowledge bases such as Wikipedia, WordNet, etc. to form a structured Wikipedia. The training data[5] contains 1.18M sentences with 47 entity types, e.g., *person, location, organization, etc.* and 24 relation types, e.g., *nationality*, *place_founded*, *employee_of*. For more detailed dataset structures, please refer to Appendix A and Figure A.1. We exclude the *None* label (*NA*) relation, as [4], since the relation positions are uncertain. The test set contains 395 samples manually annotated by Hoffmann et al. This dataset is created for DS-RE, but the data format satisfies the settings of the text-bound TE. We use this dataset to evaluate the performance of sentence-level TE. This enhances the use of automatically annotated data in supervised relation extraction. **NYT Large** dataset is further developed by Zeng et al. and Lin et al.[6] based on Riedel NYT. This dataset has 53 relation labels including the *NA* labels. The training set contains 522,611 sentences, 279,786 pairs of entity, and 18,252 facts which cover all sentences in Riedel NYT. We use this dataset to evaluate the DS-RE task.

SemEval-2018 Task 7 dataset is provided for the task of semantic relation extraction and classification in scientific papers. This dataset defines 6 relation types, e.g., *PART_WHOLE*, *USAGE*. We adopt the dataset for subtask 2. The training/test data is composed of 350/150 abstracts of scientific publications from the Association for Computational Linguistics (ACL) Anthology with manually annotated entities and relations. For more detailed dataset structures, please refer to Appendix A and Figure A.2. One of the main challenges is the limited size of the training data. To overcome it, we also select a part of the data from the noisy data of subtask 1.2 to extend the training set which finally contains 3419 sentences.

TACRED dataset is introduced in Zhang et al.. It contains 106k sentences with entity mention pairs drawn from the yearly Text Analysis Conference (TAC) Knowledge Base Population (KBP)[7] challenge. They organized crowdsourced annotations on the Mechanical Turk platform, enabling the annotation of subject and object entity spans and relation types. Sentences are annotated

[3] http://iesl.cs.umass.edu/riedel/ecml/
[4] https://en.wikipedia.org/wiki/Freebase_(database)
[5] https://github.com/shanzhenren/CoType
[6] https://github.com/thunlp/NRE
[7] https://tac.nist.gov/2017/KBP/index.html

with 41 person- and organization-oriented relation types, e.g., *per:employee_of*, *org:founded*, and no relation for negative examples. The advantage of TACRED is that it contains a large number of high-quality relation instances, which enables adequate training of model parameters, and that these entity and relation types are more generic to downstream applications. They annotated all negative instances that emerged during data collection, making the data fit the contextual complexity of relational expressions in the real-world text. For more detailed dataset structures, please refer to Appendix A and Figure A.3. Entity mentions are typed, with subjects classified into person and organization, and objects classified into 16 fine-grained types (e.g., date and location).

6.1.2. Hyper-parameters

We conduct experiments based on the 200-D pre-trained GloVe, the 300-D pre-trained FastText, the 300-D randomly initialized word vectors, and the bert-base-uncased[8] representation respectively. For the SemEval-2018 dataset, we train the domain-specific word embeddings, like (Rotsztejn et al. 2018), using GloVe and Fasttext respectively. Our corpus is composed of all the abstracts since 2001 (5.4 million tokens) collected using the application programming interface (API)[9] on arXiv.org and the ACL Anthology Reference Corpus (ARC)[10] (90 million tokens). We trained word embeddings for 500 epochs with 60 threads. We use 50-D randomly initialized character embeddings. For the TACRED and NYT Large datasets, we use the 300-D GloVe embeddings, like Zhang et al..

For regularization we apply dropout with $p = 0.5$. The output dimension of character CNN is 100-D. We set the hidden size of the LSTM unit to 300D. We set $\lambda = 1.0$. To select better models, we divide each training epoch of NYT into 100 subdivisions and the training epoch of SemEval-2018 into 4 subdivisions randomly and evenly, and each subdivision retains a model for model selection, as this method can generate better models within the epoch. For the NYT and SemEval-2018 dataset, we use the Adam optimization algorithm to update the model parameters with an initial learning rate = 0.001, decay rate = 0.9, batch size = 32 and epoch = 30. For the TACRED dataset, we use Stochastic Gradient Descent (SGD) with the initial learning rate = 0.3, decay rate = 0.9, batch size = 64, cutoff = 5.0 for gradient clipping and epoch = 50. For NYT Large, we use SGD with the learning rate = 0.5, batch size = 64 and epoch = 30. When we fine-tune the model using BERT representation, we set learning rate = 3e−5, batch size = 8 and epoch = 5. We conduct experiments on an Intel(R) Xeon(R) CPU E7-4830 v3 @ 2.10 GHz (Mem: 976G) and the GPU Tesla K40c and TITAN RTX.

6.1.3. Evaluation

For the NYT dataset, we adopt the standard micro F1 score, recall (R), and precision (P) as metrics for NER and RC subtasks. A correct prediction is that the extracted triple matches the ground truth including two entities, relation direction, and type. For the SemEval-2018 dataset, we use the official script. The evaluation is carried out in two steps. First, the relation label and directionality are ignored, so it only evaluates the quality of entity pairs by the F1 score. Second, the evaluation of relation classification is the macro-average F1 score. For the TACRED dataset, we report the micro F1 score as Zhang et al.. For the DS-RE task, we adopt the held-out evaluation as [15], which is an effective evaluation method for a large dataset without costly human intervention. We compare the precision and recall curve. The curve is drawn by ranking all predicted instances according to their confidence scores and traversing the ranking list from the high score to the low score to measure the precision and recall at each position.

6.2. Results of triple extraction

The first part of Table 2 lists some baseline systems for joint entity and relation extraction. The other part is our methods based on context-independent and context-dependent representation. To eliminate the influence of random factors, we did 3 separate runs and took the average. When there is massive training data, our model can achieve good results even without using pre-trained word embeddings. This is because our model enhances the relation classification module. The joint training scheme enables two subtasks to interact with each other through shared representation. Pre-trained word embeddings generally improve model performance. Using GloVe embedding in this model can increase the F1 by 3.42%. Feature-based (BERT-FB) and fine-tuning (BERT-FT) based representation further improve results. BERT representation helps the model consider context information of different granularities. Table 3 lists the results of two subtasks. Pre-trained word embeddings also help to improve the NER subtask. BERT-based representation brings more improvements to the performance of the RE subtask. This indicates that the RE subtask needs to consider more context information.

To observe the training process, we divide the first training epoch into 25 intervals and evaluate the model on the test set. Fig. 4 shows the model performance. We observe that using pre-trained word embeddings achieves a faster learning speed than using the randomly initialized word vectors. The result of the RE subtask changes more significantly because the context of RE is more diverse. Our model learns the two subtasks jointly, making the learning process effective. The training set of the NYT dataset contains a lot of noise introduced by automatic data annotation, and denoising methods can be further studied in the future. Our method could be a reasonable choice when the training data is noisy. The benefits of deep learning lie in its powerful representation capabilities and end-to-end learning schemes. Deep learning can combine pre-trained word embeddings with the capacity to model contextual representations. When using word embeddings, it can achieve significant improvement. Even with randomly initialized word vectors, the deep learning model can update the parameters of the model through downstream tasks to obtain acceptable results. Without pre-annotated entities, the relation extraction results may be unstable, and the problem of enhancing model stability will be further studied in the future.

[8] https://huggingface.co/bert-base-uncased
[9] https://arxiv.org/help/api/index
[10] http://acl-arc.comp.nus.edu.sg/

Table 2
Results on the NYT dataset.

Methods	P	R	F1
MultiR	33.8	32.7	33.3
DS-Joint	57.4	25.6	35.4
Line	33.5	32.9	33.2
FCM	55.3	15.4	24.0
SPTree[a]	37.3	15.4	23.4
CoType	42.3	51.1	46.3
LSTM-LSTM-Bias	61.5	41.4	49.5
Transition	64.3	42.1	50.9
MRT	**67.4**	42.0	51.7
SwitchNet+Random	55.09	49.66	**52.23**
SwitchNet+GloVe	60.28	**52.27**	**55.95**
SwitchNet+FastText	58.61	**53.11**	**55.66**
BERT+TE	50.68	**56.20**	**53.30**
SwitchNet+BERT-FB	64.27	**52.35**	**57.69**
SwitchNet+BERT-FT	62.82	**57.49**	**60.02**

[a]This experiment is conducted by Wu et al. in the ReQuest System.

Table 3
Results of subtasks on the NYT dataset.

Tasks	NER			RE		
Metrics	P	R	F1	P	R	F1
SwitchNet+Random	91.00	87.22	89.06	55.09	49.66	52.23
SwitchNet+GloVe	92.44	91.77	92.10	60.28	52.27	55.95
SwitchNet+FastText	92.01	91.55	91.76	58.61	53.11	55.66
SwitchNet+BERT-FB	89.73	91.03	90.37	64.27	52.35	57.69
SwitchNet+BERT-FT	91.13	93.43	92.26	62.82	57.49	60.02

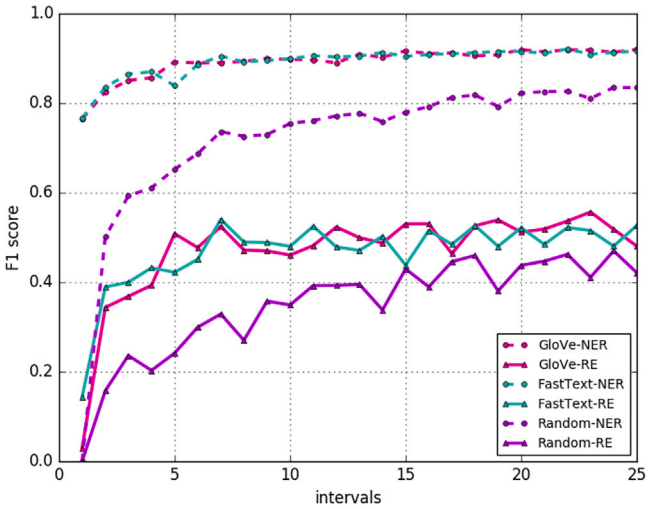


Fig. 4. Learning process in the first epoch (average sampling of 25 intervals).

6.3. Results of relation extraction

To evaluate the performance of the RE task, we conduct experiments on the SemEval-2018 dataset. Entity mentions in each sample are available, so our model does not need to predict the EOIs actively. Formulation (16) becomes $A = \mathcal{F}_{rc}(\mathbf{h}, \Psi(\mathbf{y}))$ where $\mathbf{y}$ is the label sequence. Our model achieves good performance as shown in Table 4. The first part of Table 4 lists baseline models and the second part is our approach. For the RC subtask, our model score is higher than the second (UWNLP) on the official leaderboard. For the RE subtask, our model score is higher than the third (SIRIUS-LTG-UiO). However, the official results were obtained within a limited time and the number of submissions. Their systems used NLP pipelines to extract some features, while the CNN-based system that did not use much feature engineering achieved 18.46% Macro-F1 on the RC subtask. Feature engineering can provide much prior knowledge directly when the model did not fully learn the features. ETH-DS3Lab system achieves the highest score, but this system relies heavily on feature engineering and data augmentation, while our model is jointly trained.

Table 4
Results on the SemEval-2018 dataset.

Systems	Relation instance			Relation classification		
Metrics	P	R	F1	P	R	Marco-F1
UC3M-NII (5th)	–	–	35.4	–	–	18.5
Bf3R (4th)	–	–	33.4	–	–	20.3
SIRIUS-LTG-UiO (3rd)	–	–	37.4	–	–	33.6
UWNLP (2nd)	–	–	**50.0**	–	–	39.1
ETH-DS3Lab (1st)	–	–	48.8	–	–	**49.3**
BERT+RE	–	–	40.86	–	–	42.10
SwitchNet+Random	47.19 ± 2.46	19.52 ± 3.22	27.39 ± 2.80	30.45 ± 1.09	23.57 ± 0.97	26.57 ± 0.95
SwitchNet+GloVe	35.25 ± 0.88	49.77 ± 3.49	41.18 ± 0.66	43.50 ± 2.12	41.05 ± 3.86	42.04 ± 1.30
SwitchNet+FastText	34.63 ± 0.70	50.59 ± 6.81	40.87 ± 1.94	44.99 ± 8.63	36.39 ± 4.13	39.34 ± 0.94
SwitchNet+BERT-FB	34.62 ± 2.60	59.76 ± 2.18	43.84 ± 2.61	52.93 ± 3.19	49.83 ± 5.81	45.58 ± 0.71

Table 5
Results of relation classification.

Settings	P	R	Marco-F1
SwitchNet+Random	66.44 ± 1.40	72.85 ± 1.81	69.46 ± 0.41
SwitchNet+GloVe	74.53 ± 1.89	79.16 ± 2.82	77.35 ± 1.80
SwitchNet+FastText	75.96 ± 0.67	79.98 ± 1.04	77.91 ± 0.62
SwitchNet+BERT-FB	71.20 ± 0.34	73.83 ± 2.37	72.01 ± 0.94

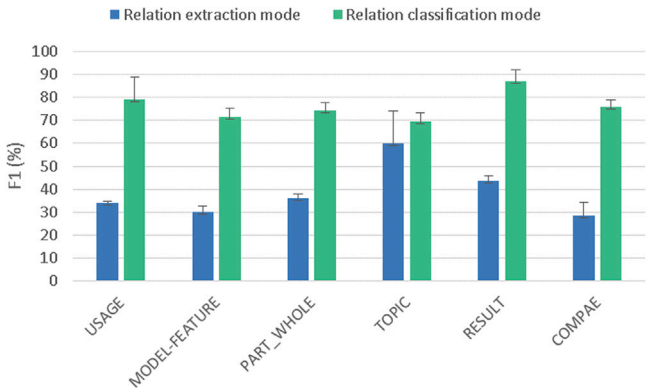


Fig. 5. Class-level comparison of RC and RE.

We only input tokens of simple text and entities. Compared with the models not using much feature engineering, our model achieves significant improvements (8.44% F1 and 27.12% Macro-F1) on RE and RC results respectively. This is because our model generates more relation candidates by using POEOI inference and joint training. These candidates provide rich supervision signals that can update model parameters from more perspectives. The SemEval-2018 dataset is manually annotated but still noisy, so mechanisms to improve data annotation can be investigated in the future. The experimental results are affected by hyper-parameter tuning. In addition, programming skills also directly affects the process of data processing and model parameter learning. In the future, we will further optimize the design patterns and programming tricks. Other systems do a lot of optimization in hyper-parameter tuning, feature engineering, model selection, model integration, etc. To win the competition, these systems require a lot of data processing and model optimization work. Unlike their motivations, this paper mainly focuses on evaluating the adaptability of the proposed framework to different data protocols. We simplify many processes, and the model sometimes achieves relatively good but not optimal results. We do not delve into data optimization strategies and techniques beyond the model, and such work can be further investigated in the future.

6.4. Results of multiple relation classification

To evaluate the influence of pre-annotated relation instances for the RC task, we conduct experiments based on Section 6.3. Note that this is a MRE task because each sentence may preserve multiple relations. Formulation (16) becomes $A = \mathcal{F}_{rc}(\mathbf{h}, \text{POEOIs})$ because POEOIs are provided in the input sample. Compared with Table 4, the pre-annotated POEOIs significantly improve the results of relation extraction.

For the BERT-FB, domain adaption of context-dependent representation requires more supervision signals than other pre-trained word embeddings, while the supervision signals are more sparse in this task. We visualize the prediction results of each class in RE

Table 6
Results on the TACRED.

Systems	P	R	F1
LR	**73.5**	49.4	59.4
SDP-LSTM	66.3	52.7	58.7
Tree-LSTM	66.0	59.2	62.4
PA-LSTM	65.7	64.5	65.1
C-GCN	69.9	63.3	66.4
C-GCN+PA-LSTM	71.3	65.4	**68.2**
TRE	70.1	65.0	67.4
BERT	67.23	64.81	66.00
ERNIE	69.97	**66.08**	67.97
SwitchNet	71.6	61.1	65.9
SwitchNet+C-GCN	73.3	62.9	67.6

and RC tasks in Fig. 5 where the x- and y- axes are the relation type and the F1 score respectively. The RE and RC results are from Table 4 (blue bars) and Table 5 (green bars) respectively. This figure shows that the RC task achieves a more balanced result than the RE task. This is because, during model training, using the pre-annotated POEOIs enhances the correlation of the entity pairs and improves the quality of supervision signals. During testing, the input explicitly indicates the relation positions so the prediction uncertainty is reduced and the false positive prediction also decreases.

6.5. Results of single relation classification

To analyze the generality of the RC component, we also conduct experiments on the TACRED. Note that this is the SRE task because each sentence only focuses on one POEOI. Following Zhang et al. we concatenate word embedding, position embedding, POS tag, and entity label embedding and consider the entire sentence. Our model achieves a higher result than the PA-LSTM as shown in Table 6. This is because the query vector q of our attention captures the global context of a sentence and the hidden representation contains position information.

The RC result is slightly lower than GCN, and this indicates that the POEOI inference scheme has more contribution. GCN has proven very effective in relation classification because its representation captures long-range syntactic relations by encoding the dependency structure. We added GCN to encode syntactic features, that is, $y = \mathcal{F}_{rc}(\mathbf{h} \oplus \mathcal{F}_{gcn}(\mathbf{x}), \text{POEOIs})$ where $\mathbf{h}$ and $\mathbf{x}$ are the hidden states and the dependency tree respectively. $\mathcal{F}_{gcn}(\cdot)$ denotes the GCN model. In this setting, the F1 score is increased by 1.7%, because syntactic information can provide complementary benefits when we extract informative tokens by a sequence model. ERNIE achieves a higher score because it incorporates entities in KGs to enhance the pre-trained language model, but the downside is the computation costs.

We observe that "*per:age*", "*per:title*"and "*org:founded*" relations are easier to extract. "*org:member_of*" and "*per:country_of_death*" are more problematic because model performance is concerned with the class imbalance problem and the context length ($s - o$ distance) of the input text. There are few instances of these missing relations, e.g., "*per:country_of_death*" only has 63 samples, so these relations are not fully learned. The context length of each relation type also influences the result. We observe that relations typically expressed in a longer context tend to be more difficult to extract. We visualize this dataset in Appendix A in supplementary material.[11] It is promising to introduce entity embedding of knowledge graphs or explore the few-shot learning to alleviate this problem. Our proposed method does not achieve the best results because the prototype proposed in this paper may not be good enough in program implementation. C-GCN and PA-LSTM have many optimizations in data preprocessing, dropout and gradient clipping, and these projects are also better organized in terms of design patterns and programming techniques. In the future, we can further optimize the design patterns and programming implementation of the framework.

6.6. Results of distant supervision relation extraction

For DS-RE, prior works show that sentence-level attention effectively reduces the effects of noisy instances. Following [15], we concatenate the position embeddings. For all the methods, we use the common GloVe word embeddings. Table 7 lists the precision of the top-ranked relations and the area under the curve (AUC) of the model. AUC can measure model performance, as shown in Fig. 6. The higher the AUC, the better the model. We observe that the single-head SwitchNet (S-SwitchNet) and multi(12)-head SwitchNet (M-SwitchNet) both achieve significant improvements in terms of P@N and AUC. As the recall increases, our methods have higher precision than other settings. This means that our methods can extract more high-quality relations.

We observe that SwitchNet+ONE achieves a higher AUC than CNN/PCNN+ONE because the word-level attention enhances the sentence representation. Sentence-level attention (SwitchNet+ATT) further improves model performance because sentence-level attention can alleviate the effects of noisy instances, making the model more robust. After we add multi-head attention, this model improves 1.8% AUC value. This is because the multi-head attention can generate relation representation by attending to different tokens from multiple perspectives. Ablation studies can be found in Section 7.1.2. In summary, our model uses hierarchical attention to better encode bag representation.

[11] https://github.com/nnntt/SwitchNet/blob/master/Appendix/appendix.pdf

Table 7
P@N for relation extraction in the entity pairs with different number of top-ranked predictions.

P@N(%)	100	200	300	Mean	AUC(%)
CNN+ONE	77.2	72.1	69.1	72.8	33.6
CNN+ATT	74.3	72.1	66.1	70.8	33.5
PCNN+ONE	79.2	79.1	71.1	76.5	34.0
PCNN+ATT	73.3	71.1	66.8	70.4	34.1
SwitchNet+ONE	81.1	75.1	68.4	74.9	35.0
SwitchNet+ATT	**82.2**	**77.1**	71.4	**76.9**	36.1
+Multi-head	81.2	74.1	**72.1**	75.8	**37.9**

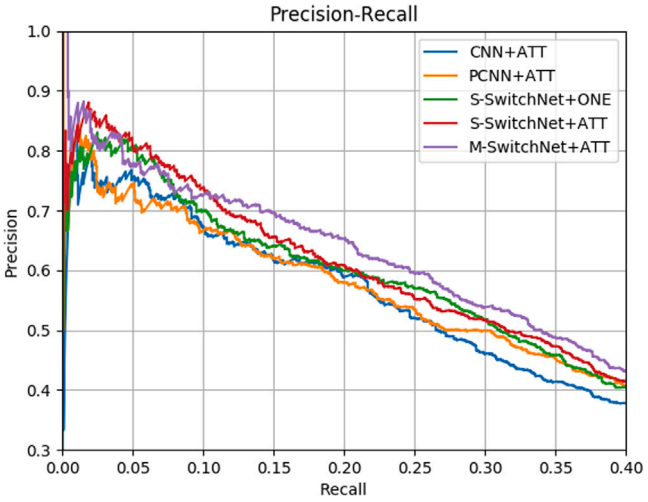


Fig. 6. Precision and recall curves of different settings.

7. Analysis

In this section, we performed ablation studies on the text-bound and distant supervision relation extraction respectively. Then, we analyze the influence of relation candidate selection on model training. Finally, we use a case study to visualize the model's attention distribution.

7.1. Ablation study

7.1.1. Text-bound relation extraction

Table 8 shows an ablation study of multi-task training and pipeline training on the NYT dataset. Two systems denote a fine-tuned NER system and a SwitchNet system. We use a BERT model as the NER system. We first compare the multi-task training in different settings. A first observation is that relying on the fine-tuned NER system slightly reduces the final results by 1.72% F1. Although the NER system is enhanced, some entities still hurt the final precision. Because extracting more entities does not always help the relation extraction subtask and some entities may increase the risk of predicting false positives. The above training process is a multi-stage pipeline and relation candidates are determined by the NER system. We first fine-tune the NER system and then predict the label sequence $\hat{Y}_{ner}$ which are also written to disk. Then we train the joint entity and relation extraction model that can use the $\hat{Y}_{ner}$ information. When we replace $\hat{Y}_{ner}$ with the ground truth NER labels, the result is significantly improved (5.03% F1). This means that a high-quality label sequence can help improve the final results.

Removing the multi-task training degrades the performance by 0.94% F1. This means that multi-task training benefits the RE subtask from the NER subtask. After removing the NER system, the joint optimization process does not rely on the NER system, and the performance improves by 1.72% F1. Then, we remove the NER system and multi-task training. We first train the NER subtask, then freeze the parameters of the shared layer, and then train the RE subtask. The F1 score drops by 2.51% because the RE subtask cannot be encoded at the lower layer to interact with the NER subtask. This means that multi-task joint training is critical for subtask interaction. When we do not freeze the lower layer, the result drops significantly. Because when we train the RE subtask, the memory of the NER subtask is weakening. This phenomenon is also known as catastrophic forgetting.

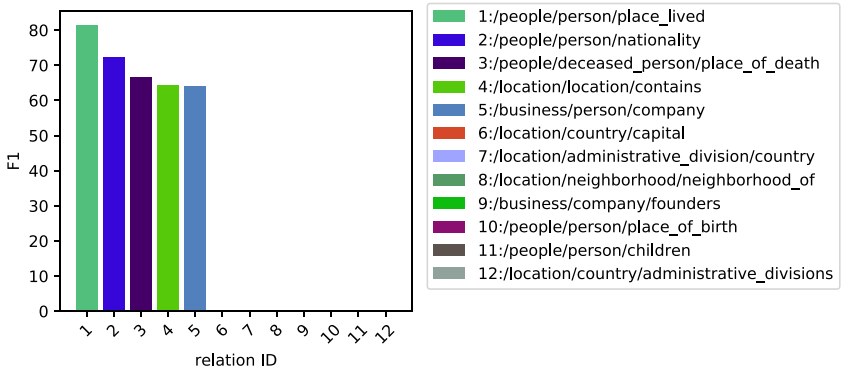
We also apply our training pipeline to BERT. We fine-tune a simple BERT model for TE. Table 9 shows the results. When only using multi-task training, BERT achieves slightly lower results than our SwitchNet. When we remove the multi-task, NER system, and Frozen NER, this model almost forgets all the NER memory, so the RE subtask fails. Fine-tuning models can achieve different functions, which also means that BERT is sensitive to parameter changes.

Table 8
SwitchNet setting ablation.

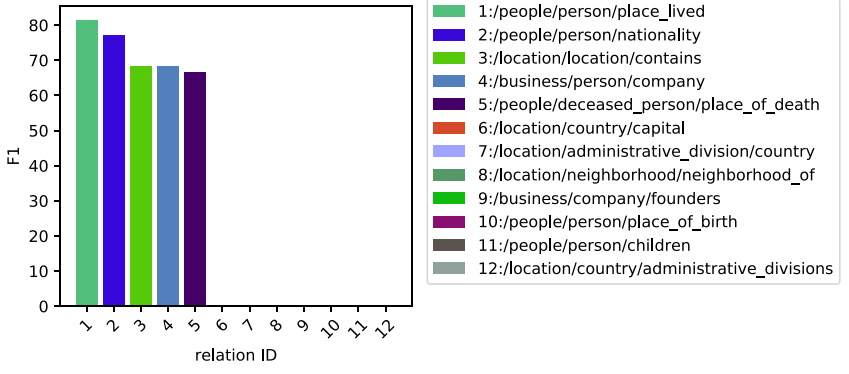
Model	P	R	F1
Two systems + multi-task	56.78	51.89	54.23
+ NER label	61.18	57.46	59.26
– multi-task	55.64	51.13	53.29
– NER system	60.28	52.27	55.95
– multi-task, NER system	50.33	56.96	53.44
– multi-task, NER system, Frozen NER	55.88	43.29	48.78

Table 9
BERT model setting ablation.

Model	P	R	F1
Two systems + multi-task	51.30	54.93	53.05
+ NER label	54.89	61.01	57.79
– multi-task	50.46	54.43	52.37
– NER system	50.68	56.20	53.30
– multi-task, NER system	46.46	54.93	50.34
– multi-task, NER system, Frozen NER	–	–	–



(a) Multi-task training



(b) Multi-task training + NER label

Fig. 7. Class-aware result.

Fig. 7 shows the class-aware prediction result. Fig. 7(a) and 7 (b) denote the model with or without the ground truth NER labels when extracting triples. We observe "*place_lived*", "*nationality*", "*place_of_death*", "*contains*" and "*company*" relations are easier to extract. Other relations in the test set sometimes are not extracted. This is because some sparse relations are not fully learned and there is a class imbalance problem in the training data. We visualize and analyze this dataset in Appendix A. We observe that providing NER labels enhances the extraction of well-trained relations, while other relations are not obviously improved.

Table 10
SwitchNet setting ablation.

Model	100	200	300	Mean	AUC(%)
M-SwitchNet+ATT	81.2	74.1	72.1	75.8	37.9
+ subject, object	79.2	72.1	69.8	73.7	35.0
– Multi-head	82.2	77.1	71.4	76.9	36.1
– Multi-head, ATT	81.1	75.1	68.4	74.9	35.0
– Multi-head, Single-head	73.3	73.6	69.8	72.2	35.1
– Multi-head, Single-head, ATT	75.2	71.6	70.8	72.6	34.7

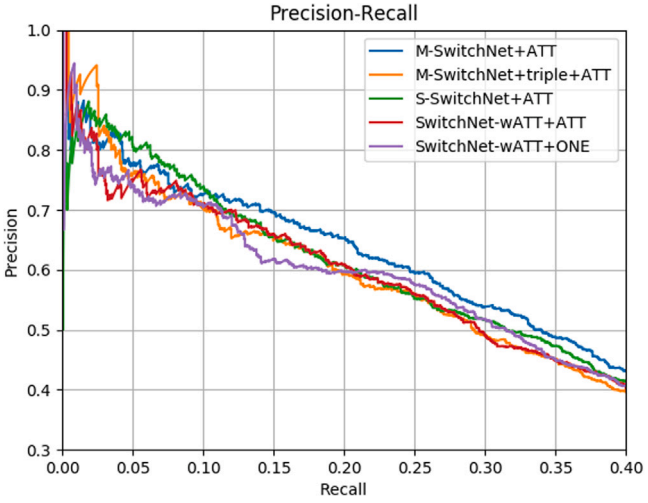


Fig. 8. Precision and recall curves of different settings.

7.1.2. Distant supervision relation extraction

We performed ablation studies on DS-RE. Table 10 shows the results of different settings. A first observation is that adding representations of subject and object to the multi-head attention reduces the results. This is because all positive and noisy samples in the bag contain the same subject and object, so using this information may hurt model training. When we remove the multi-head attention and keep only one head, the AUC decreases, but the P@N of the top-ranked relations improves. When we also remove the sentence-level attention, the result drops. This means that sentence-level attention is important in this framework. When we remove only the word-level attention, the result also drops. This means the word-level and sentence-level attention mechanisms complement each other to form the hierarchical attention mechanism. When we remove the two-level attention, the results will drop further.

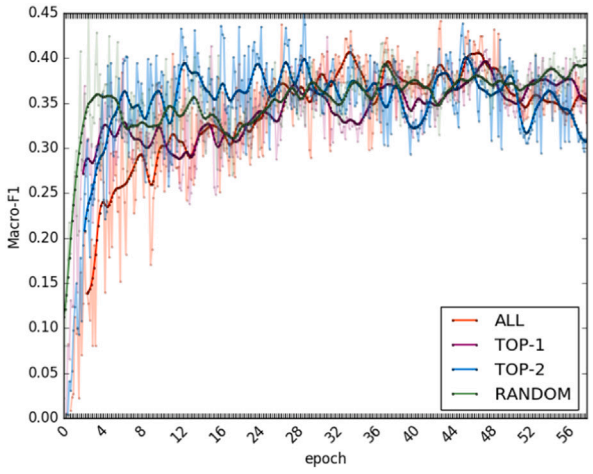
Fig. 8 shows the precision and recall curves of different settings. We observe that when we do not use the **w**ord-level **att**ention (wATT), the curves generally move down. Precision drops faster as recall increases. This means that word-level attention can help achieve higher performance. When we remove the sentence-level attention (ATT), the curve fluctuations increase. This means that the sentence-level selection enhances the stability of the model.

7.2. Analysis of relation candidates selection

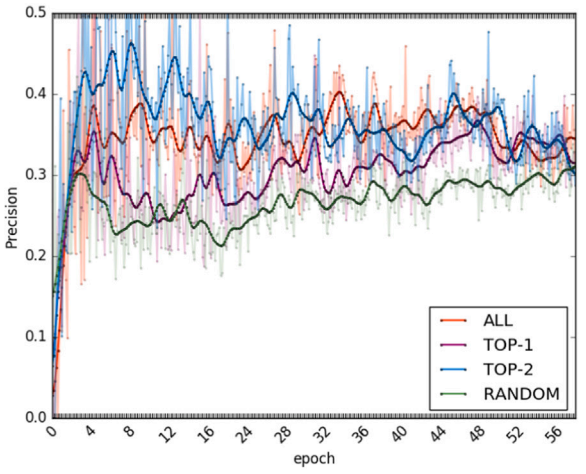
Candidates' feedback can be positive and negative and both types of feedback have great potential to boost recall and precision. However, the number of possible relations is $\mathcal{O}(n^2)$, where n is the number of entities, which potentially increases computational complexity and may lead to the class imbalance problem. Existing approaches for this question typically perform random sampling, which might include some inefficient relation candidates.

The relation extraction task is also a mathematical model. When there are too many candidate relationships in a sample, the generalization of the model will be affected during the learning process, and it will face the problems mentioned in the support vector machine (SVM). SVM is based on the VC dimension theory of statistical learning and the principle of minimum structural risk to obtain the best generalization ability. Inspired by the idea of the optimal hyperplane, we can think of these indistinguishable relation candidates as an approximation of the support vectors. This setting not only reduces the computational complexity but also helps to enhance the generalization performance of the model, especially for noisy datasets.

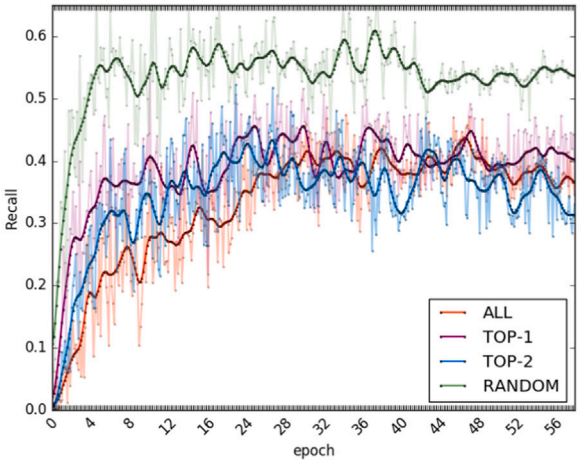
Inspired by the idea of SVM, a few support vectors are effective to decide the classification hyperplane. We assume that difficult candidates that are closer to the classification hyperplane can improve the classifier more effectively. This can reduce computational complexity. During model training, we rank the negative candidates according to their prediction probability of being a None relation and only keep top-k negative ones with the least probabilities. These relations are more likely not to be predicted as NA. k is a



(a) Macro-F1



(b) Precision



(c) Recall

Fig. 9. Prediction results on SemEval-2018 based on different selection strategies for relation candidates.

hyper-parameter, and we experimented with $k \in \{1, 2, \infty\}$. ∞ denotes all candidates. This ranking mechanism filters a vast number of negative candidates leaving the classifier with a small set.

We compared several selection strategies, i.e. random selection (RANDOM), top-k selection (TOP-K), and all candidates (ALL), as shown in Fig. 9. We find that using top-k selection can achieve comparable F1 scores to using all candidates, but the precision and recall might be not balanced enough. When more negative candidates are retained, the recall will decrease and the precision will increase. This method potentially enhances the use of the ranking mechanism to reduce the computational complexity of RE models.

7.3. Case study

It is instructive to analyze which words the model attends to when classifying relations. We hand-picked some examples in the SemEval-2018 dataset and visualize the attention patterns of these samples. Fig. 10 shows how our model extracts informative words for relation representation. The first column is the sample id. The second column contains the extracted triples, and the third column shows the textual input. The first sentence shows that this model is capable of focusing on informative words to identify the "*PART_WHOLE*" relation type for "*English-Chinese Bitexts*" and "*Web*". The second sentence shows this model resolves the comparative relation by attending to "*narrower than*". The third sentence shows that "*are described in*" means the "*TOPIC*" relation.

The fourth sentence shows that "*produced by means*" for "*Translations*" and "*beam search decoder*" denotes the "*MODEL-FEATURE*" relation, while the fifth sentence shows this model extracts the relation type "*RESULT*" by attending to "*produce best*". This suggests

Id	Triple	Sentences
1	(English-Chinese bitexts, PART_WHOLE, Web)	This piece of work has also laid a foundation for exploring and harvesting English-Chinese bitexts in a larger volume from the Web .
		This piece of work has also laid a foundation for exploring and harvesting English-Chinese bitexts in a larger volume from the Web .
2	(domains, COMPARE, MUC-4 terrorism domain)	These previous domains were much narrower than the MUC-4 terrorism domain .
		These previous domains were much narrower than the MUC-4 terrorism domain .
3	(paper, TOPIC, overview)	An overview of HowNet and information structure are described in this paper .
		An overview of HowNet and information structure are described in this paper .
4	(beam-search decoder, MODEL-FEATURE, Translations)	Translations are produced by means of a beam-search decoder .
		Translations are produced by means of a beam-search decoder .
5	(Bayesian classifiers, RESULT, recall performance)	In our evaluation , Bayesian classifiers produce the best recall performance of 80 % but the precision is low (60%) .
		In our evaluation , Bayesian classifiers produce the best recall performance of 80 % but the precision is low (60%) .
6	(WordNet, USAGE, Word Sense Disambiguation (WSD) task)	WordNet has been used extensively as a resource for the Word Sense Disambiguation (WSD) task , both as a sense inventory and a repository of semantic relationships .
		WordNet has been used extensively as a resource for the Word Sense Disambiguation (WSD) task , both as a sense inventory and a repository of semantic relationships .

Fig. 10. Visualization of some cases where the underlined phrase represents an entity in the extracted triple and the red degree denotes the word weight for the relation representation.

that this model considers entity semantics and sentence context. The sixth sentence shows that this model can extract informative tokens "*resource for*" instead of the important verb in a long context. These results indicate that the model considers the entity semantics and the relational context while attending to the informative words for relation representation.

This model also extracts some wrong results. For the sentence *"In essence this can start to rewrite the history of photography", said Grant Romer, director of the advanced residency program in photograph conservation at the George Eastman House in Rochester.*, *George Eastman House in Rochester* implies that (*Rochester, contains, George Eastman House*). However, the result is (*Rochester, place_lived, George Eastman House*). *"Grant Romer ... at George Eastman House in Rochester"* indicates a state but does not mean the relation type *"place_lived"*. The understanding of the temporal state of the relation needs to be further improved. There are still some errors when extracting the relation of the spatial position, such as `neighbor_of`. The understanding of the spatial relations needs to be further improved.

8. Conclusion

Our SwitchNet is implemented for different information extraction configurations, namely NER, RC, RE, TE, and DS-RE. This framework designs a modular neural architecture and uses different information flows to tackle different relation extraction tasks, which significantly reduces the problem complexity for relation extraction tasks. Our model predicts EOIs using a standard NER setup and uses an attention-based classifier to classify relations of POEOIs. Reducing the unknowns of model prediction by indicating the POEOIs can help improve the performance. We also address the secondary problem of jointly learning NER and RE through the POEOI inference and the shared representation mechanisms. However, the challenge is that there is a sort of information loss in the discrete entity detection process. Extensive experiments are conducted on multiple datasets, and our framework achieves good performance.

The relation classifier of this framework is highly extensible, and the joint optimization method is beneficial to upgrade the pipeline method to an end-to-end information extraction model. The systematic design of information flow and modular neural networks is a promising way to reduce the problem complexity when designing an artificial intelligence (AI) system with diverse functions. In the future, we will combine this framework with entity linking technology so that the extracted knowledge can be directly used in the knowledge base. Furthermore, we plan to extend this framework to open information extraction.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.compeleceng.2022.108445.

Data availability

Github link is provided: https://nnntt.github.io/SwitchNet/

References

[1] Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2014, p. 580–7.
[2] Poo M-m, Du J-l, Ip NY, Xiong Z-Q, Xu B, Tan T. China brain project: Basic neuroscience, brain diseases, and brain-inspired computing. Neuron 2016;92(3):591–6.
[3] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Proceedings of NIPS. 2017, p. 5998–6008.
[4] Zheng S, Wang F, Bao H, Hao Y, Zhou P, Xu B. Joint extraction of entities and relations based on a novel tagging scheme. In: Proceedings of ACL. Association for Computational Linguistics; 2017, p. 1227–36.
[5] Zeng D, Liu K, Lai S, Zhou G, Zhao J. Relation classification via convolutional deep neural network. In: Proceedings of COLING 2014. 2014, p. 2335–44.
[6] Xu Y, Mou L, Li G, Chen Y, Peng H, Jin Z. Classifying relations via long short term memory networks along shortest dependency paths. In: Proceedings of EMNLP. The Association for Computational Linguistics; 2015, p. 1785–94.
[7] Miwa M, Bansal M. End-to-end relation extraction using LSTMs on sequences and tree structures. In: Proceedings of ACL. The Association for Computer Linguistics; 2016, p. 1105–16.
[8] Zhang Y, Qi P, Manning CD. Graph convolution over pruned dependency trees improves relation extraction. In: Proceedings of EMNLP. Association for Computational Linguistics; 2018, p. 2205–15.
[9] Sun C, Gong Y, Wu Y, Gong M, Jiang D, Lan M, et al. Joint type inference on entities and relations via graph convolutional networks. In: Proceedings of ACL. Association for Computational Linguistics; 2019, p. 1361–70.
[10] Wang H, Tan M, Yu M, Chang S, Wang D, Xu K, et al. Extracting multiple-relations in one-pass with pre-trained transformers. In: Proceedings of ACL. Association for Computational Linguistics; 2019, p. 1371–7.
[11] Bekoulis G, Deleu J, Demeester T, Develder C. Joint entity recognition and relation extraction as a multi-head selection problem. Expert Syst Appl 2018;114:34–45.
[12] Han X, Gao T, Yao Y, Ye D, Liu Z, Sun M. OpenNRE: An open and extensible toolkit for neural relation extraction. In: Proceedings of EMNLP-IJCNLP - system demonstrations. Association for Computational Linguistics; 2019, p. 169–74.
[13] Mintz M, Bills S, Snow R, Jurafsky D. Distant supervision for relation extraction without labeled data. In: Proceedings of ACL. The Association for Computer Linguistics; 2009, p. 1003–11.
[14] Riedel S, Yao L, McCallum A. Modeling relations and their mentions without labeled text. In: Joint European conference on machine learning and knowledge discovery in databases, vol. 6323. Springer; 2010, p. 148–63.
[15] Lin Y, Shen S, Liu Z, Luan H, Sun M. Neural relation extraction with selective attention over instances. In: Proceedings of ACL. 2016, p. 2124–33.
[16] Banko M, Cafarella MJ, Soderland S, Broadhead M, Etzioni O. Open information extraction from the web. In: Proceedings of IJCAI 2007. p. 2670–6.
[17] Devlin J, Chang M, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT. Association for Computational Linguistics; 2019, p. 4171–86.
[18] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. In: Proceedings of ICLR workshop track. 2013.
[19] Zhang X, Zhao J, LeCun Y. Character-level convolutional networks for text classification. In: Proceedings of NIPS, vol. 28. 2015, p. 649–57.
[20] Ma X, Hovy EH. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In: Proceedings of ACL. The Association for Computer Linguistics; 2016, p. 1064–74.
[21] Rotsztejn J, Hollenstein N, Zhang C. ETH-DS3lab at SemEval-2018 task 7: Effectively combining recurrent and convolutional neural networks for relation classification and extraction. In: Proceedings of the international workshop on semantic evaluation. 2018.
[22] He R, Lee WS, Ng HT, Dahlmeier D. An unsupervised neural attention model for aspect extraction. In: Proceedings of the ACL. 2017, p. 388–97.
[23] Yang Z, Yang D, Dyer C, He X, Smola A, Hovy E. Hierarchical attention networks for document classification. In: Proceedings of NAACL. 2016, p. 1480–9.
[24] Vafaie H, Imam IF. Feature selection methods: Genetic algorithms vs. greedy-like search. In: Proceedings of the international conference on fuzzy and intelligent control systems, vol. 51. 1994, p. 28.
[25] Dorigo M, Caro GD, Gambardella LM. Ant algorithms for discrete optimization. Artif Life 1999;5(2):137–72.

Hongyin Zhu received the B.A. degree in Automation from Shandong University, Jinan, China, in 2014, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2020. He worked as a Postdoctoral Researcher with the Department of Computer Science and Technology, Tsinghua University, Beijing. He is currently working at Inspur Electronic Information Industry Co., Ltd., Beijing.

Prayag Tiwari received his Ph.D. degree from the University of Padova, Italy. He is currently working as an Assistant Professor at Halmstad University, Sweden. Previously, he worked as a Postdoctoral Researcher at the Aalto University, Finland, and Marie Curie Researcher at the University of Padova, Italy. His research interests include Machine Learning, Deep Learning, Quantum Machine Learning, NLP, Healthcare, and IoT.

Yazhou Zhang received his Ph.D. degree in the College of Intelligence and Computing from Tianjin University (Tianjin, China) in 2020. He is currently a Lecturer in Software Engineering College at Zhengzhou University of Light Industry (Zhengzhou, China). He is also a Postdoctoral Fellow in Tianjin University-China Mobile Communications (Tianjin) Joint Laboratory. He was a visiting researcher at Harbin Institute of Technology in 2013, Beijing Institute of Technology in 2019, and University of Macau in 2019.

Deepak Gupta received a B.Tech. Degree in 2006 from the Guru Gobind Singh Indraprastha University, Delhi, India. He received an M.E. degree in 2010 from Delhi Technological University, India, and Ph.D. degree in 2017 from Dr. APJ Abdul Kalam Technical University (AKTU), Lucknow, India. He completed his Post-Doc from the National Institute of Telecommunications (Inatel), Brazil, in 2018. He has co-authored more than 207 journal articles, including 168 SCI papers and 45 conference articles.

Meshal Alharbi is an Assistant Professor of Artificial Intelligence in the Department of Computer Science at Prince Sattam Bin Abdulaziz University in the Kingdom of Saudi Arabia. He received Ph.D. degree in Computer Science from Durham University, UK, in 2020, and M.Sc. degree in Computer Science from Wayne State University, USA, in 2014. He has 10 years of Experience in Teaching/Research,/Industry. His research interests lie in the Artificial Intelligence

Applications and Algorithms, Agent-Based Modelling and Simulation Applications, Disaster/Emergency Management and Resilience, Optimization Applications, and Machine Learning.

Tri Gia Nguyen received the B.Ed. degree in computer science from the Hue University of Education, Vietnam, in 2011, the M.Sc. degree in computer science from Duy Tan University, Vietnam, in 2013, and the Ph.D. degree in computer science from Khon Kaen University, Thailand, in 2017. He is currently the Head of Department of Information Security, FPT University, Danang, Vietnam. His research interests include the Internet of Things, sensor networks, wireless communications, wireless energy harvesting networks, mobile computing, edge computing, software-defined networking, network functions virtualization, and network security.

Shahram Dehdashti received his Ph.D. degree from the University of Isfahan, Iran. He worked as an assistant researcher at Zhejiang university, China, from 2014 to 2017. He then moved to Madison, USA, to work at the University of Wisconsin 2017–2018. He has spent time at Queensland University of Technology, employed as an assistant researcher from 2018 to 2021. Currently he is employed as senior researcher at Technical University of Munich, Germany. His research interests include Quantum Machine Learning, Quantum Networks, Quantum Cognition, etc.