Semi-Open Information Extraction

Bowen Yu
Institute of Information Engineering,
Chinese Academy of Sciences
yubowen@iie.ac.cn

Zhenyu Zhang
Institute of Information Engineering,
Chinese Academy of Sciences
zhangzhenyu1996@iie.ac.cn

Tingwen Liu
Institute of Information Engineering,
Chinese Academy of Sciences
liutingwen@iie.ac.cn

ABSTRACT

Open Information Extraction (OIE), the task aimed at discovering all textual facts organized in the form of (*subject, predicate, object*) found within a sentence, has gained much attention recently. However, in some knowledge-driven applications such as question answering, we often have a target entity and hope to obtain its structured factual knowledge for better understanding, instead of extracting all possible facts aimlessly from the corpus. In this paper, we define a new task, namely Semi-Open Information Extraction (SOIE), to address this need. The goal of SOIE is to discover domain-independent facts towards a particular entity from general and diverse web text. To facilitate research on this new task, we propose a large-scale human-annotated benchmark called SOIED, consisting of 61,984 facts for 8,013 subject entities annotated on 24,000 Chinese sentences collected from the web search engine.

In addition, we propose a novel unified model called USE for this task. First, we introduce subject-guided sequence as input to a pre-trained language model and normalize the hidden representations conditioned on the subject embedding to encode the sentence in a subject-aware manner. Second, we decompose SOIE into three uncoupled subtasks: predicate extraction, object extraction, and boundary alignment. They can all be formulated as the problem of table filling by forming a two-dimensional tag table based on a task-specific tagging scheme. Third, we introduce a collaborative learning strategy that enables the interactive relations among subtasks to be better exploited by explicitly exchanging informative clues. Finally, we evaluate USE and several strong baselines on our new dataset. Experimental results demonstrate the advantages of the proposed method and reveal insight for future improvement.

CCS CONCEPTS

• **Natural language processing → Information extraction**;

ACM Reference Format:
Bowen Yu, Zhenyu Zhang, and Tingwen Liu. 2021. Semi-Open Information Extraction. In *WWW '21: 30th The Web Conference, April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 12 pages. https://doi.org/xx.xx

WWW '21, April 19–23, 2021, Ljubljana, Slovenia
© 2021 Association for Computing Machinery.
ACM ISBN XX…$XX.XX
https://doi.org/xx.xx

1 INTRODUCTION

With the explosive growth of the web corpora, extracting structured knowledge from unstructured, open-domain and diverse web text has become increasingly important to the web content mining [11, 18, 22]. This task is known as Open Information Extraction (OIE). An OIE system usually converts natural text to semi-structured representations, by extracting a set of relational facts organized as a triple in the form of (*subject, predicate, object*) from plain text itself [5, 15], which does not rely on pre-defined ontology schema. The extracted facts can be applied as the source data for many downstream tasks, including knowledge base population [25], word analogy [16], text comprehension [21], etc.

While OIE aiming to extract all possible facts in the text, we observe that in lots of knowledge-driven applications, we are not interested in all facts, but those associated with a specific entity. For example, in the task of question answering [28] and entity typing [33], we often have a target entity, and hope to enrich it with related informative facts for better language understanding [19, 40]. Retrieving such facts from existing knowledge bases (KBs) serves as a possible solution. However, while current KBs are quite large, they are acknowledged as incomplete due to the dynamics of this ever-changing world, i.e., some target entities may lack facts in KBs rather than in the real world [2]. Therefore, it is necessary to explore how to extract open-domain facts from the web corpora towards a target entity. We name this new paradigm as Semi-Open Information Extraction (SOIE) because it inherits the domain-independent property of OIE while restricting one involved entity.

In this work, we study the problem of SOIE, and present SOIED, a large-scale human-annotated dataset for it. SOIED is constructed with the following three features: (1) SOIED annotates knowledge towards particular subject entities[1], requiring model to make extraction in the subject-aware manner. (2) Besides labeling relational facts, SOIED also focuses on three common types of lexical facts, including description, synonym and hyponymy of the given subject in each sentence. (3) The predicate and object in the relational facts are not limited to contain a contiguous sequence of words. In contrast, both of them can be discontiguous, and contain a list of spans. The resulting dataset contains 8,013 subjects and 61,984 facts annotated on 24,000 Chinese sentences retrieved from the web corpus, making it large, general and diverse enough to train an accurate extractor.

Intuitively, SOIE can be regarded as a special form of OIE. So, we can adapt OIE methods for SOIE by injecting the target subject information into extraction process. Existing OIE approaches usually follow the generation-based or labeling-based framework.

[1] We take the subject entity as an example to explore semi-open information extraction in this paper, while extracting facts towards specific object entity is also feasible.

Generation-based models [5, 17, 18, 27] cast OIE into a text generation problem and leverage the sequence-to-sequence architecture, which allows for selecting source tokens to generate arbitrary output facts [20]. Nevertheless, this kind of models inevitably suffer from the well known problem: exposure bias. At training time, the ground truth tokens are used as context while at inference the sequence is generated by the resulting model on its own, leading to a gap between training and inference [38]. Different from generation-based models, labeling-based models convert fact extraction to a sequence labeling problem [24, 26, 37]. By designing ingenious tagging schemes, they can make full use of word order information and achieve consistency between training and inference. However, for convenience of labeling, their underlying assumption is that the fact elements are contiguous spans in the sentence, which does not always hold in practice. Statistics on SOIED shows that 13.78% relational facts involve discontiguous predicates, and 6.42% contain discontiguous objects, indicating that identifying these discontiguous structures is necessary for information extraction.

In order to overcome the limitations of such prior works, we propose USE, a **U**nified model for **S**emi-open information **E**xtraction. It firstly generates a new input sequence by appending the given subject to the beginning of a sentence, and feed it into the pre-trained LM to encode the sentence in a subject-aware manner. A novel conditional layer normalization mechanism is then introduced to enhance the semantic dependency between the subject and the contextual representations by normalizing the activities of the neurons based on the subject embedding. On top of the subject-specific representation sequence, USE regards lexical facts as typed objects and decomposes the overall SOIE task into three subtasks: predicate extraction (PE), object extraction (OE) and boundary alignment (BA) from a table filling perspective. Given an n-word sentence, three $n \times n$ tag tables are formed for PE, OE and BA respectively by a table filling network, where each entry at row i and column j is assigned a unique tag according to the interaction between the i-th and j-th word of the input sentence. Specifically, for PE and OE, we devise a novel multi-hop tagging scheme to solve the discontinuous structure recognition problem by annotating all the spans subsequent to the one starting with the position i in the i-th row of the corresponding tag table, and merging these spans according to the global subsequent relation. BA is to distinguish and align the boundary tokens of (*predicate, object*) pairs from scratch. This is achieved by detecting the entry that the two corresponding positions are respectively the beginning tokens of a valid (*predicate, object*) pair or the ending tokens. Overall, the tag tables of PE, OE and BA are generated independently, and will be consumed together by a special decoding algorithm to recover desired facts from them, thus immune from the exposure bias problem. These subtasks are trained jointly based on a collaborative learning strategy, which fully exploits the interactive relations among them to mutually enhance their performance.

To evaluate our USE model, we adapt recent state-of-the-art information extraction methods to semi-open information extraction, and conduct thorough experiments on SOIED. Experimental results demonstrate that the proposed USE consistently outperforms baseline methods. Furthermore, detailed analysis shows that USE significantly improves the performance on multiple fact extraction, discontinuous fact extraction and unseen fact extraction, and reveals multiple promising directions worth pursuing. We make SOIED and the code for USE and baselines publicly available at https: //anonymous.com/for/review.

2 RELATED WORK

Although semi-open information extraction has been largely neglected in the literature, several related tasks have been well studied such as relation extraction, joint extraction, open relation extraction and open information extraction.

Relation Extraction (RE) aims to classify the semantic relation between given entities from plain text into pre-defined relations [29]. Zeng et al.[36] showed that CNN with position embeddings is effective for RE. Zhang et al.[39] proposed graph convolution over dependency trees and achieved promising results on public benchmarks. These RE methods seek to mine structured facts from text. But they suffer from two main limitations: (1) requiring recognized entities as input may be affected by error propagation [35]; (2) using a pre-defined set to cover those relations with open-ended growth is difficult [12].

Joint Extraction (JE) aims to detect entity pairs along with their relations using a single model. PATag [6] transforms joint extraction to several sequence labeling problem by tagging entity and relation labels simultaneously according to each query word position. ETL [35] performs subject recognition as the first step, and extracts the corresponding object and relations for each subject. Compared with RE, this extraction paradigm reduces the error propagation owing to the joint modeling. But it still focuses on answering narrow, well-defined requests over a predefined set of target relations [23].

Open Relation Extraction (ORE) aims to discover new relation types that hold between two entities mentioned in the text from unsupervised open-domain corpora. Elsahar et al.[10] extracted rich linguistic features for relation instances, and clustered semantic patterns into several relation types. Wu et al.[32] proposed to learn similarity metrics of relations from labeled data of pre-defined relations, and transfer the relational knowledge to identify novel relations in unlabeled data. Compared with conventional RE, ORE does not rely on specific relation types and extracts relational facts with minimized or even no human annotation.

Open Information Extraction (OIE) extracts textual triples comprising relation phrases and argument phrases from within a text, without requiring pre-specified relations or pre-identified entity pairs. Stanovsky et al.[26] proposed a novel formulation of OIE as a sequence tagging problem. However, this model lacks the elegance to identify discontinuous structures, which may lead to poor recall. Cui et al.[5], Sun et al.[27] and Liu et al.[18] built generation-based OIE extractor by directly decoding a prediction sequence containing a list of facts from the input source sentence, thus addressing the discontinuous structure recognition problem. Nevertheless, these methods have difficulty in capturing the token order within pieces of arguments and predicates, as they usually perform free token-level decoding operation. Besides, generation-based methods actually decompose OIE into several inner-dependency steps, since the decoder needs a recursive decoding process, inevitably causing the exposure bias problem [13].

Table 1: An example from SOIED, with target subject entity underlined for clarity.

	Chinese	English Translation
Sentence	全国社会保障基金（社保基金）是政府用以提供社会保障的基金，主要由用人单位和个人缴费构成，包括养老保险基金，医疗保险基金等，用于各项社会保险待遇的当期发放。	National Social Security Fund (Social Security Fund) is the fund used by the government to provide social security, mainly composed of employer and individual payment, including endowment insurance fund and medical insurance fund, used for the current payment of social insurance benefits.
Relational Facts	(由|构成, 用人单位缴费) (由|构成, 个人缴费) (包括, 养老保险基金) (包括, 医疗保险基金) (用于, 各项社会保险待遇的当期发放)	(composed of, employer payment)(composed of, individual payment) (including, endowment insurance fund) (including, medical insurance fund) (used for, the current payment of various social insurance benefits)
Description	政府用以提供社会保障的基金	the fund used by the government to provide social security
Synonym	社保基金	Social Security Fund
Hyponymys	养老保险基金, 医疗保险基金	endowment insurance fund, medical insurance fund

In the recent years, there has been a growing interest in developing open-domain information extraction (ORE and OIE) methods because they forgoes per-relation training data and is not bound by a fixed relation vocabulary in contrast to traditional closed-domain extraction (RE and JE). In this paper, we design a novel semi-open information extraction task which aims at discovering domain-independent facts towards a given entity. In essence, it fills the gap between ORE and OIE. On the one hand, when compared with ORE, our SOIE paradigm no longer requires all the entities are identified in advance, thus expanding the downstream application scenarios. On the other hand, SOIE sets up a more focused extraction direction than OIE, making it more suitable in obtaining structured knowledge for specific entities that we care about.

3 SOIE DATASET

In this section, we first introduce our knowledge expression form, based on which the SOIE facts are accurately expressed. Then we detail the collection process of the human-annotated data. Finally, we analyze various aspects of SOIED to provide a deeper understanding of the dataset and the task of SOIE.

3.1 Format

On the basis of previous fact annotation schemes [24, 27], we have analyzed a great many sentences. We conclude that most of the factual knowledge related to the target subject entity can be classified into the following classes: (1) Relational fact involved the given subject, denoted as (*predicate, object*) pair; (2) Lexical fact of the given subject, including description, synonym, and hyponymy, is represented in the form of a continuous phase. Our annotation format is designed to record all these types of facts.

Specifically, we adopt the ideology of "literally honest" following the philosophy of open-domain extraction. That is, as much as possible, we use the words in the original sentence to express knowledge, allowing SOIE systems to extract facts without relying on any pre-defined ontology schema. Table 1 shows an example sentence and the annotated facts for the target subject 全国社会保障基金 (National Social Security Fund). In some cases, predicates or objects in relational facts may be divided into several parts residing in discontinuous locations of the sentence; we categorize these cases into two classes. In the first class, we merge separated spans in order, which can form a continuous and complete expression after such processing, e.g., 用人单位缴费 (employer payment) in the relational fact of Table 1. In the second class, we introduce an extra symbol "|", which works like a placeholder in the predicate, denoting that the corresponding object should be inserted in this place to express the exact relation between the target subject and the extracted object, e.g., 由|构成 (composed of).

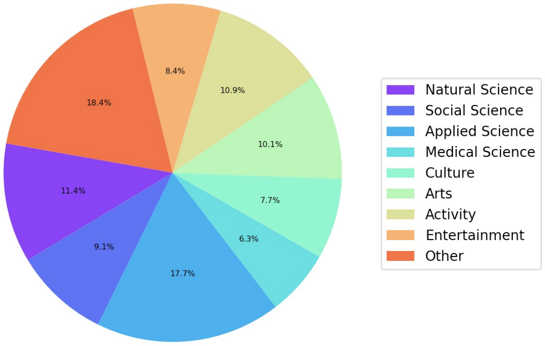


Figure 1: Subject entity domain distribution of SOIED.

Table 2: Sentence-level fact number distribution of SOIED.

	0	1	2	3	4	≥5
Number	5,431	6,786	3,303	2,295	1,777	5,606
Proportion	22.63%	28.28%	13.76%	9.56%	7.40%	23.36%

Table 3: Fact type distribution of SOIED.

	Relational Fact	Description	Synonym	Hyponymy
Number	34,882	7,491	6,443	13,168
Proportion	56.28%	12.09%	10.39%	21.24%

3.2 Human-Annotated Data Collection

Our human-annotated data is collected in three stages: (1) **Creating target subject entity set.** As SOIE is designed to scale to massive open-domain corpora such as the Web, and we hope our constructed dataset is completely domain-independent. Towards this goal, we propose to generate a diverse and general-purpose target entity set by sampling from the crowdsource entity dictionary used by the Chinese IME Sougou[2], which provides a large number of entities from many fields, including science, culture, art, entertainment, etc (2) **Collecting candidate sentences.** Each entity in the target set is used as a query term to retrieve the relevant web pages by the Baidu search engine[3]. Then we use goose3[4] to extract the text of web pages and retain the sentences containing the retrieval entities as candidate (*subject, sentence*) instances. (3) **Annotating the target-related facts.** Next, we invite some well-educated college students in computer science to annotate facts for the collected instances according to the defined annotation format. Two annotators label each instance, and if they have disagreements on one instance, one or more annotators are asked to judge it.

3.3 Data Analysis

The final SOIED dataset consists 61,984 relational facts annotated on 24000 sentences for 8,013 subjects. On average, each instance in SOIED contains 67.38 tokens and 2.58 facts. See Table 2 for fact number distribution. Table 3 shows the numbers and proportions of four types of facts contained in the data set. A notable property of our dataset is that both predicates and objects in relational facts can be discontiguous, and contain a list of spans. Detailed analysis reveals that 13.78% relational facts involve discontiguous predicates, and 6.42% contain discontiguous objects, indicating that identifying these discontiguous structures is necessary and important for information extraction. To verify the domain-independent property of our dataset, we randomly sampled 1000 target subject entities, and manually analyze their domains. As shown in Figure 1, subject entities straddling a number of areas, which means the model trained on SOIED is not constrained in any specific domain.

4 METHODOLOGY

This section provides our Unified model for Semi-open information Extraction (USE) in details. We will present the task decomposition strategy, the overall architecture, the design of each component, and the overall workflow.

4.1 Task Decomposition

The goal of SOIE is to extract relational facts and lexical facts (description/hyponymy/synonym) of the given subject entity. One intuitive solution is pipeline: starting with a sentence and a target subject entity, it first extracts all the candidate predicates, objects and lexical facts that may be related to the given subject entity, then enumerates all possible (*predicate, object*) pairs and classify whether each of these pair is valid or not concerning the subject entity. While being easy to implement, this process is vulnerable to errors cascading down the pipeline. The earlier extraction stage error is inherited and magnified in the later classification stage, causing the poor overall performance. Thus, our core observation is that if we can decouple the dependency between these two stages, such a framework would be unified and end-to-end.

In this work, in addition to encoding subject information, we propose to fold the semi-open information extraction process into three uncoupled subtasks: 1) predicate extraction: learning to detect all the (potentially discontinuous) corresponding predicates for the given subject; 2) object extraction: similar to predicate extraction but focusing on objects and lexical facts; 3) boundary alignment: learning to distinguish and align the boundary tokens of (*predicate, object*) pairs from scratch. Their prediction results can be generated independently and do not contain any inter-dependency extraction steps.

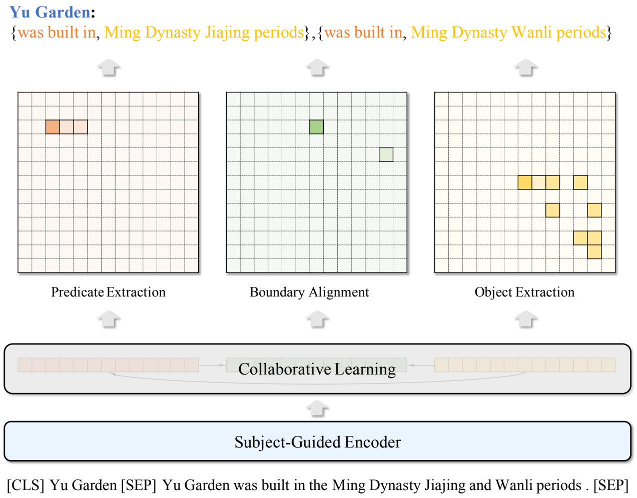

Yu Garden:
{was built in, Ming Dynasty Jiajing periods},{was built in, Ming Dynasty Wanli periods}
Predicate Extraction
Boundary Alignment
Object Extraction
Collaborative Learning
Subject-Guided Encoder
[CLS] Yu Garden [SEP] Yu Garden was built in the Ming Dynasty Jiajing and Wanli periods . [SEP]

Figure 2: The overall structure of the USE model. The subject is combined with the context as the subject-guided encoder's input, followed by collaborative learning. Three tag tables are then formed for predicate extraction, object extraction, and boundary alignment, respectively. Different colors in the tables represent different tags. Finally, the related facts are decoded from the tag tables as the output.

This task formulation comes with several key advantages: firstly, efficient one-stage modeling to extract paired predicates and objects at the same time is supposed to avoid the cascading errors in the pipeline model; secondly, treating predicate extraction and object extraction as individual subtasks are beneficial to capture their task-specific information in the learning process; thirdly, by regarding lexical fact as a particular object with fine-grained type and hidden predicate, this formulation provides a natural way to handle the relational fact and lexical fact in a unified framework; Fourthly, it allows us to exploit the well-developed multi-task learning techniques to comprehensively model interactive relations between different subtasks.

4.2 Overview

Figure 2 shows the overall architecture of USE. Formally, given a subject entity and a sentence containing it, the subject-guided encoder first builds the subject-aware contextual representations as the shared features. After that, the collaborative learning module learns the private representation for each subtask defined in the task formulation and enables their interactions to be better exploited. Upon the task-oriented private features, three table filling networks are deployed to synchronously deliver predicates, objects,

[2] https://pinyin.sogou.com/dict/
[3] https://baidu.com
[4] https://github.com/goose3/goose3

and aligned boundaries, which are finally consumed together to output factual knowledge of interest elegantly.

4.3 Subject-Guided Encoder

For each instance composing of a *subject, sentence* pair, the subject-guided encoder's goal is to integrate the subject information into the word representations, which is in favor of the following process of extracting subject-related facts. Specifically, we use Bidirectional Encoder Representations from Transformers (BERT) [9], a pre-trained bidirectional Transformer encoder that achieves state-of-the-art performances across a variety of NLP tasks, as our backbone network. To adapt BERT to consider target information straightforwardly, we design the target-guided input. Each training input is organized by concatenating the tokens of the target subject entity, denoted by T, together with the sentence tokens S, to form the packed sequence X : "[CLS]"+ T + "[SEP]" + S + "[SEP]". Then for each token x_i in X, we convert it into vector space by summing the token, segment, and position embeddings, thus yielding the input embeddings$\mathbf{X}^0 = [\mathbf{x}_1^0, \cdots, \mathbf{x}_n^0]$. Next, we use a series of L stacked Transformer blocks to project the input embeddings into a sequence of contextual vectors $\mathbf{X}^L = [\mathbf{x}_1^L, \cdots, \mathbf{x}_n^L]$ as:

$$\mathbf{X}^l = \text{TransformerBlock}(\mathbf{X}^{l-1}), \forall l \in [1, L]. \quad (1)$$

Furthermore, to give contextual vectors more guidance towards the target, inspired by [8], we introduce a novel conditional layer normalization (CLN) mechanism based on the well-known layer normalization (LN) [1]. LN was proposed to normalize neurons' activities to reduce the covariate shift problem in deep neural networks. It can be defined as a linear mapping function as:

$$\text{LN}(\mathbf{x}, \alpha, \beta) = \alpha \odot (\frac{\mathbf{x} - \mu}{\sigma}) + \beta, \quad (2)$$

$$\mu = \frac{1}{d}\sum_{i=1}^{d} x_i, \quad \sigma = \sqrt{\frac{1}{d}\sum_{i=1}^{d}(x_i - \mu)^2}, \quad (3)$$

where x_i is the i-th element of the input vector $\mathbf{x} \in \mathbb{R}^d$, μ and σ are the mean and standard deviation taken across the elements of $\mathbf{x}$, respectively. $\mathbf{x}$ is first normalized by fixing the mean and variance and then scaled and shifted by α, and β, which are learnable vectors shared between instances. Different from LN, CLN dynamically generates α and β based on prior knowledge rather than learning them as other parameters in neural networks. In SOIE, the subject is the essential guide for extraction, so we propose to take its feature vector as a condition to generate α and β as follows:

$$\alpha_e = \mathbf{W}_\alpha \mathbf{e} + \mathbf{b}_\alpha, \quad \beta_e = \mathbf{W}_\beta \mathbf{e} + \mathbf{b}_\beta, \quad \mathbf{e} = \frac{1}{t-1}\sum_{j=2}^{t} \mathbf{x}_j^L, \quad (4)$$

where $\mathbf{e}$ denotes the representation of the target subject entity, computed by averaging the embeddings of BERT over the concatenated subject tokens T, t is the ending index of T in X. For different subjects, different LN parameters are generated, which results in effectively adapting input representations to be more subject-specific; this process can be defined as:

$$\mathbf{H} = [\text{CLN}(\mathbf{x}_1^L, \alpha_e, \beta_e), \cdots, \text{CLN}(\mathbf{x}_n^L, \alpha_e, \beta_e)], \quad (5)$$

where $\mathbf{H}$ refers to the conditional normalized vector sequence. Considering that LN has been employed as the output layer of each Transformer block, we directly replace the LN layer of the last (L-th) block in BERT with our CLN, and $\mathbf{x}_*^L$ in Equations 4-5 denotes the representation before LN operation in the last block.

4.4 Table Filling Network

We utilize a unified architecture for predicate extraction (PE), object extraction (OE), and boundary alignment (BA) according to our task formulation. This paper wraps such architecture into a general model named table filling network. Formally, given an n-word sentence, our network constructs a $n \times n$ tag table by enumerating all possible token pairs and giving each token pair a unique tag according to their relation. The key information for recognizing the relationship between the i-th token x_i and the j-th token x_j include: (1) the semantic of x_i; (2) the semantic of x_j; (3) the contextual information related to these two tokens. Under this consideration, we generate the representation $\mathbf{p}_{i,j}$ for (x_i, x_j) as follows:

$$\mathbf{s}_{i,j,z} = \mathbf{v}^\top \tanh(\mathbf{W}_a[\mathbf{h}_i \oplus \mathbf{h}_j \oplus \mathbf{h}_z]), \quad (6)$$

$$a_{i,j,z} = \frac{\exp(s_{i,j,z})}{\sum_{m=1}^{n}\exp(s_{i,j,m})}, \quad \mathbf{c}_{i,j} = \sum_{z=1}^{n} a_{i,j,z}\mathbf{h}_z, \quad (7)$$

$$\mathbf{p}_{i,j} = \tanh(\mathbf{W}_p[\mathbf{h}_i \oplus \mathbf{h}_j \oplus \mathbf{c}_{i,j}]), \quad (8)$$

where $\mathbf{W}_a$ and $\mathbf{W}_p$ are parameter matrices and $\mathbf{v}$ is a vector to be learned, $\mathbf{h}_j, \mathbf{h}_p, \mathbf{h}_z$ are the hidden states at position j, p and z respectively, $\oplus$ denotes the concatenation operator. Equation 7 means the states at the two focused positions are used for comparing with all the token representations to collect relevant information from the context. Finally, we feed $\mathbf{p}_{i,j}$ into a fully-connected layer, which is followed by a Softmax function to compute label distribution:

$$P(y_{i,j}) = \text{Softmax}(\mathbf{W}_b \mathbf{p}_{i,j} + \mathbf{b}_b). \quad (9)$$

Then, by learning different table filling parameters for PE, OE and BA, we can generate different $P(y_{i,j}^{\mathcal{I}})$, where $\mathcal{I} \in \{\text{PE}, \text{OE}, \text{BA}\}$ is the subtask indicator. The label of (x_i, x_j) is predicted as:

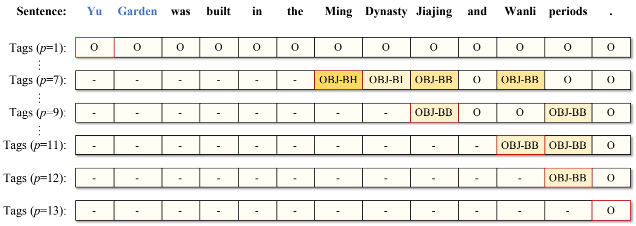
$$\text{tag}_{i,j}^{\mathcal{I}} = \arg\max_k P(y_{i,j}^{\mathcal{I}} = k), \quad (10)$$

where $P(y_{i,j}^{\mathcal{I}}) = k$ represents the probability of identifying the label of (x_i, x_j) as k in the subtask $\mathcal{I}$.

4.5 Predicate extraction and object extraction

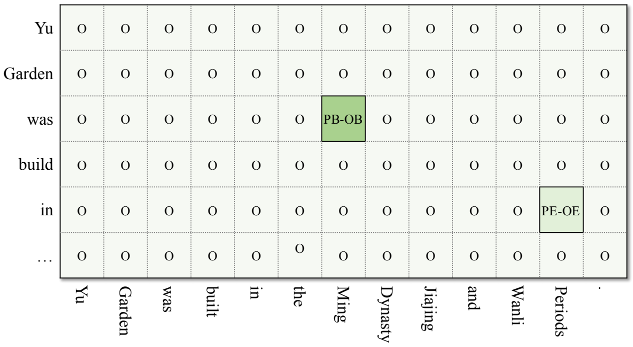
In this subsection, we introduce our multi-hop tagging scheme for extracting predicate and object. For the sake of generality, we do not distinguish them in some parts of the following paragraphs, and they are collectively referred to as an element.

First of all, we define a set of labels {B, I, |-B, BH, |-BH, IH, BD, |-BD, ID, O} for prediction extraction, and {OBJ-B, OBJ-I, OBJ-BH, OBJ-IH, OBJ-BD, OBJ-ID, DES-B, DES-I, SYN-B, SYN-I, HYP-B, HYP-I, O} for object extraction. Each label contains up to two parts: position and task-specific. In the position part, we extend the traditional BIO tag set with four new position indicators {BH, IH, BB, IB} to represent the discontinuous element: (1) BH indicates the word is the beginning of a head, where head stands for the first indivisible span of a discontinuous element. Here indivisible means that if different elements share a part of a span, this span needs to disassemble the shared part as a separate span; (2) IH indicates it locates inside of a head; (3) BB indicates it locates in the first place of a body,



Sentence:	Yu	Garden	was	built	in	the	Ming	Dynasty	Jiajing	and	Wanli	periods	.
Tags (p=1):	O	O	O	O	O	O	O	O	O	O	O	O	O
Tags (p=7):	-	-	-	-	-	-	OBJ-BH	OBJ-BI	OBJ-BB	O	OBJ-BB	O	O
Tags (p=9):	-	-	-	-	-	-	-	-	OBJ-BB	O	O	OBJ-BB	O
Tags (p=11):	-	-	-	-	-	-	-	-	-	-	OBJ-BB	OBJ-BB	O
Tags (p=12):	-	-	-	-	-	-	-	-	-	-	-	OBJ-BB	O
Tags (p=13):	-	-	-	-	-	-	-	-	-	-	-	-	O

Figure 3: An example of our multi-hop tagging scheme for object extraction, where p is the query word position. The words highlighted in blue denote the target subject entity.



	Yu	Garden	was	built	in	the	Ming	Dynasty	Jiajing	and	Wanli	Periods	.
Yu	O	O	O	O	O	O	O	O	O	O	O	O	O
Garden	O	O	O	O	O	O	O	O	O	O	O	O	O
was	O	O	O	O	O	O	PB-OB	O	O	O	O	O	O
build	O	O	O	O	O	O	O	O	O	O	O	O	O
in	O	O	O	O	O	O	O	O	O	O	O	PE-OE	O
…	O	O	O	O	O	O	O	O	O	O	O	O	O

Figure 4: An example of our boundary alignment tagging scheme for the instance presented in Figure 3.

where body defined as an indivisible span after the head; and (4) IB means inside of body. In the task-specific part, we associate tags with task-specific information if necessary. For example, in predicate tagging, we introduce a symbol "|" to instruct that "|" should be inserted before the corresponding token, adapting the tagging results consistent with our annotation format (Section 3.1). As to object tagging, we introduce four object type tags: OBJ, DES, SYN, and HYP to denote a general-typed object, description, synonym, and hyponymy, respectively. Note that lexical facts usually comprise continuous token sequences, so we only utilize B and I to represent their positions.

Nevertheless, this encoding may be lossy in some cases since the information on which parts constitute the same element is lost. For example, when extracting objects, even if the model can correctly predict the tag sequence O O O O O O OBJ-BH OBJ-IH OBJ-BB O OBJ-BB OBJ-BB for the instance presented in Figure 3, we cannot deduce that *Ming Dynasty periods* alone is not an object. To revolve this ambiguity issue, we present an effective multi-hop tagging scheme. For an n-word sentence, n different tag sequences are annotated according to different query position p. In each sequence, if the current query position p is the start of an indivisible span in an element, tokens in this span are labeled with special tags according to the span type, and other tokens locating in the spans next to this span in discontinuous elements are signed as non-O tags based on their corresponding roles. From another perspective, each tag sequence can be seen as a row of a tag table, so the tagging scheme can be well modeled by the table filling network. Considering that our tagging scheme only labels the current query position and the following words, we discard the lower triangle region of the tag table, so $\frac{n^2+n}{2}$ tags are actually generated for an n-word sentence.

Figure 3 provides an example of our tagging scheme for object extraction. When the query position p is 7, the token *Ming* at this position is labeled as OBJ-BH followed by the OBJ-BI tag of *Dynasty*, denoting that *Ming Dynasty* is the head of discontinuous element. Moreover, in the same sequence, *Jiajing* and *Wanli* are both labelled with OBJ-BB, showing that these two tokens are detected as the subsequent bodies of *Dynasty*. For *Jiajing*, we can obtain that its subsequent span is *periods* from the tag sequence of $p = 9$. Similarly, *periods* is also identified as the subsequent body of *Wanli* when $p = 11$. All of the tokens are labeled as O except *periods* when p is 12 because there is no a span beginning with *periods* has bodies. The tags in the lower triangular region of the table are marked with "-", denoting they are discarded in actuality. Taken together, these tagging results imply that *Ming Dynasty Jiajing periods* and *Ming Dynasty Wanli periods* should be extracted from the instance as the candidate objects, because of the recursive subsequent relation in {*Ming Dynasty*, *Jiajing*, *periods*} and {*Ming Dynasty*, *Wanli*, *periods*}. This operation is similar to the multi-hop reasoning in QA [34], therefore we call the tagging scheme as multi-hop tagging.

4.6 Boundary Alignment

Boundary alignment is to align the boundary tokens of valid (*predicate, object*) pair. Towards this goal, given an n-word sentence, we construct a $n \times n$ tag table and give each entry in the table a unique tag. Formally, two types of tags are defined as follows: (1) predicate beginning to object beginning (PB-OB) indicates it locates in a place that the two corresponding positions on the vertical and horizontal axis of the table are respectively the beginning tokens of a valid (predicate, object) pair; (2) predicate ending to object ending (PE-OE) is similar to PB-OB, but focusing on the ending token. Then this subtask can be handled by our table filling network. An example of boundary alignment tagging is provided in Figure 4, where the sentence contains two relational facts for the subject *Yu Garden*: (*was built in*, *Ming Dynasty Jiajing periods*), (*was built in*, *Ming Dynasty Wanli periods*). Thus, the tags of (*was*, *Ming*) is PB-OB. Similarly, PE-OE is labeled at the place of (*in*, *periods*).

4.7 Collaborative Learning

Overall, our proposed USE method contains three high-level modules: predicate extractor (PE), object extractor (OE), and boundary alignment (BA). These modules can be trained simultaneously by receiving a shared representation of the input sentence, known as the parameter-sharing learning, which is a practical approach to improve the performance of a single task with other related tasks. However, we argue that simply learning a common feature space is insufficient to yield optimal performance for the complete SOIE task because it fails to consider the interactive relations among different subtasks explicitly. These relations convey collaborative signals which can mutually enhance the subtasks.

Formally, after carefully analyzing the framework of USE, we summarize three kinds of relations, as shown in Figure 5, including: (1) R_1: the two-way relation between PE and OE; (2) R_2: the one-way relation between BA and PE; (3) R_3: the one-way relation

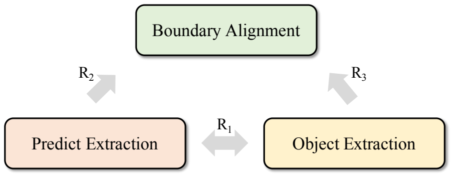


Figure 5: Interactive relations among subtasks.

between BA and OE. On the one hand, predicate and object are highly coupled together since the object is the predicate's target, and the predicate describes the relationship between the object and the given subject. Hence PE and OE might provide indicative clues to each other. On the other hand, BA aims to align the boundary between paired predicate and object, so the positions predicted as the beginning or ending tokens of predicate and object should be paid more attention when filling the boundary alignment table. Under these considerations, inspired by recent progress in sentiment analysis [4], we propose a collaborative learning strategy for USE. Firstly, multiple studies [3, 30] have shown that different subtasks in multi-task learning should select different task-specific features. So we encode the individual representation for each subtask by the task-private encoders:

$$\mathbf{H}^{\mathrm{pe}} = \mathrm{CNN}_{\mathrm{pe}}(\mathbf{H}), \quad \mathbf{H}^{\mathrm{oe}} = \mathrm{CNN}_{\mathrm{oe}}(\mathbf{H}), \quad \mathbf{H}^{\mathrm{ba}} = \mathrm{CNN}_{\mathrm{ba}}(\mathbf{H}). \tag{11}$$

Here builds encoders with the same convolutional neural networks (CNNs) structure but no shared parameters for each subtask.

After encoding task-private features, we introduce a message passing mechanism that propagates collaborative signals among subtasks to allow PE, OE, and BA modules to influence each other better. To learn the two-way relation R_1 between PE and OE, we want to build the connection between $\mathbf{H}^{\mathrm{pe}}$ and $\mathbf{H}^{\mathrm{oe}}$ to exchange informative clues based on their semantic relevance. Take the subtask PE for example, the semantic relevance between the i-th sequence item $\mathbf{h}_i^{\mathrm{pe}}$ in $\mathbf{H}^{\mathrm{pe}}$ and one another $\mathbf{h}_j^{\mathrm{oe}}$ in $\mathbf{H}^{\mathrm{pe}}$ is defined as follows:

$$s_{i,j}^{\mathrm{o2p}} = \mathbf{h}_i^{\mathrm{pe}\top}\mathbf{h}_j^{\mathrm{oe}}, \quad a_{i,j}^{\mathrm{o2p}} = \frac{\exp(\mathbf{s}_{i,j}^{\mathrm{o2p}})}{\sum_{m=1}^{n}\exp(\mathbf{s}_{\mathrm{i,m}}^{\mathrm{o2p}})}. \tag{12}$$

Here, pairwise relevance $s_{i,j}^{\mathrm{o2p}}$ are computed via the dot product, and then normalized as $a_{i,j}^{\mathrm{o2p}}$ for computation of a weighted sum of OE-private features as the collaborative signal to PE:

$$\mathbf{h}_i^{\mathrm{o2p}} = \sum_{j=1}^{n} a_{i,j}^{\mathrm{o2p}}\mathbf{h}_j^{\mathrm{oe}}. \tag{13}$$

We then fuse the original vector $\mathbf{h}_i^{\mathrm{pe}}$ and the received message $\mathbf{h}_i^{\mathrm{o2p}}$ from OE as the final representation of the i-th token for PE:

$$\tilde{\mathbf{h}}_i^{\mathrm{pe}} = \mathbf{W}_{\mathrm{pe}}[\mathbf{h}_i^{\mathrm{o2p}} \oplus \mathbf{h}_i^{\mathrm{pe}}] + \mathbf{b}_{\mathrm{pe}}. \tag{14}$$

Similarly, we can obtain the PE-enhanced OE feature vector $\tilde{\mathbf{h}}^{\mathrm{oe}}$.

In addition to inter-dependency, PE and OE can also work together to provide useful information for BA, as mentioned earlier. Clearly, only boundary tokens detected in PE and OE have the chance to be tagged in BA. However, if we make the BA module aware of PE and OE's results, the system would be vulnerable to errors cascading down the pipeline. To address this challenge, we propose incorporating the boundary information of predicate and object into BA-private representations with the help of auxiliary task and attention mechanism. Specifically, using PE as an example again, we introduce a new sequence labeling task named predicate boundary tagging (PB), which takes the same input features with PE but only annotating the beginning and ending of the predicate with tag set {B, E, O} as follows:

$$P(y_i^{\mathrm{pb}}) = \mathrm{Softmax}(\mathbf{W}_{\mathrm{pb}}\tilde{\mathbf{h}}_i^{\mathrm{pe}} + \mathbf{b}_{\mathrm{pb}}), \tag{15}$$

$$\mathrm{tag}_i^{\mathrm{pb}} = \arg\max_{k} P(y_i^{\mathrm{pb}} = k). \tag{16}$$

where $P(y_i^{\mathrm{pb}})$ is the predicate boundary tag probability distribution. In the similar way, we can obtain $P(y_i^{\mathrm{ob}})$ by applying another object boundary tagging (OB) layer on top of $\tilde{\mathbf{h}}_i^{\mathrm{oe}}$. Then following the intuition that the boundary tokens of predicate and object should also be the focal point of BA, we softly integrate $P(y_i^{\mathrm{pb}})$ and $P(y_i^{\mathrm{ob}})$ into BA features with the attention mechanism:

$$s_{i,j}^{\mathrm{p2b}} = \mathbf{h}_i^{\mathrm{ba}\top}\mathbf{h}_j^{\mathrm{pe}}, \quad s_{i,j}^{\mathrm{o2b}} = \mathbf{h}_i^{\mathrm{ba}\top}\mathbf{h}_j^{\mathrm{oe}}, \tag{17}$$

$$a_{i,j}^{\mathrm{p2b}} = \frac{\exp(s_{i,j}^{\mathrm{p2b}})}{\sum_{m=1}^{n}\exp(s_{i,m}^{\mathrm{p2b}})}, \quad a_{i,j}^{\mathrm{o2b}} = \frac{\exp(s_{i,j}^{\mathrm{o2b}})}{\sum_{m=1}^{n}\exp(s_{i,m}^{\mathrm{o2b}})}, \tag{18}$$

$$a_{i,j}^{\mathrm{p2b}} \leftarrow a_{i,j}^{\mathrm{p2b}} + P(y_i^{\mathrm{pb}}|\mathrm{tag}_i^{\mathrm{pb}} \in \{\mathsf{B},\mathsf{E}\}) \cdot |i-j|^{-1}, \tag{19}$$

$$a_{i,j}^{\mathrm{o2b}} \leftarrow a_{i,j}^{\mathrm{o2b}} + P(y_i^{\mathrm{ob}}|\mathrm{tag}_i^{\mathrm{ob}} \in \{\mathsf{B},\mathsf{E}\}) \cdot |i-j|^{-1}, \tag{20}$$

$$\mathbf{h}_i^{\mathrm{p2b}} = \sum_{j=1}^{n} a_{i,j}^{\mathrm{p2b}}\mathbf{h}_j^{\mathrm{pe}}, \quad \mathbf{h}_i^{\mathrm{o2b}} = \sum_{j=1}^{n} a_{i,j}^{\mathrm{o2b}}\mathbf{h}_j^{\mathrm{oe}}, \tag{21}$$

$$\tilde{\mathbf{h}}_i^{\mathrm{ba}} = \mathbf{W}_{\mathrm{ba}}[\mathbf{h}_i^{\mathrm{p2b}} \oplus \mathbf{h}_i^{\mathrm{o2b}} \oplus \mathbf{h}_i^{\mathrm{ba}}] + \mathbf{b}_{\mathrm{ba}}. \tag{22}$$

Here Equations 17-22 are similar to 12-14, the difference is that we add the probability of the token being identified as the boundary of one predicate $P(y_i^{\mathrm{pb}}|\mathrm{tag}_i^{\mathrm{pb}} \in \{\mathsf{B},\mathsf{E}\})$ to the attention score between the BA and PE features $a_{i,j}^{\mathrm{p2b}}$, and add $P(y_i^{\mathrm{ob}}|\mathrm{tag}_i^{\mathrm{ob}} \in \{\mathsf{B},\mathsf{E}\})$ to $a_{i,j}^{\mathrm{o2b}}$. By doing this, the boundary tokens can get larger attention weights and contribute more to the boundary alignment process. $|i-j|^{-1}$ is a distance-relevant factor, which decreases with increasing distance between the i-th token and the j-token. Note that the newly introduced auxiliary tasks are only used to propagate collaborative signals and do not participate in the extraction of predicates and objects. Thus our collaborative learning strategy can comprehensively model the interactive relations without being affected by cascading errors. Finally, PE, OE, and BA modules will make predictions based on the task-specific enhanced features.

4.8 Workflow

In this subsection, we introduce the training and decoding procedure of our framework.

Algorithm 1 Decoding Precedure

```
Input: The predicted tag table of predicate extractor, object extractor and
    boundary alignment module, denoted as $P$, $O$, and $B$, respectively.
Define $n \leftarrow$ Sentence Length
Initialize $S_{\text{rel}} \leftarrow \{\}$, $S_{\text{beg}} \leftarrow \{\}$, $S_{\text{end}} \leftarrow \{\}$
Obtain the predicate set $S_{\text{pre}}$ by decoding $P$
Obtain the lexical fact set $S_{\text{lex}}$ and the object set $S_{\text{obj}}$ by decoding $O$
Construct the dictionary $D_{\text{obj}}$ that maps object start position to a set of
    objects that begin with this position
Construct the position mapping dictionary $D_{\text{pre}}$ for predicate
for $i \leftarrow 1$ to $n$ do
    for $j \leftarrow 1$ to $n$ do
        if $B[i][j] = PB\text{-}OB$ then
            $S_{\text{beg}} \leftarrow S_{\text{beg}} \cup \{(i, j)\}$
        if $B[i][j] = PE\text{-}OE$ then
            $S_{\text{end}} \leftarrow S_{\text{end}} \cup \{(i, j)\}$
for $(i, j) \in S_{\text{beg}}$ do
    for predicate $p \in D_{\text{pre}}[i]$ do
        for object $o \in D_{\text{obj}}[j]$ do
            if ($p$.endposition, $o$.endposition) $\in S_{\text{end}}$ then
                $S_{\text{rel}} \leftarrow S_{\text{rel}} \cup \{(p, o)\}$
return $S_{\text{lex}}$ and $S_{\text{rel}}$
```

4.8.1 Training procedure. All the components in our framework are differentiable; thus, the whole framework can be efficiently trained with gradient-based methods. Word-level cross-entropy error is employed as the loss function:

$$\mathcal{L}^{\mathcal{I}} = -\frac{1}{T}\sum_{t=1}^{T} \mathbb{I}(y_t^{\mathcal{I}}) \circ P(y_t^{\mathcal{I}}), \tag{23}$$

where $\mathcal{I} \in \{\text{PE}, \text{OE}, \text{BA}, \text{PB}, \text{OB}\}$ is the symbol of subtask indicator. $\mathbb{I}(y)$ represents the one-hot vector with the y-th component being 1 and $y_t^{\mathcal{I}}$ is the gold label for $\mathcal{I}$ at the t-th position. T stands for the length of tag sequence. For an n-word sentence, T is equal to n for PB and OB, $\frac{n^2+n}{2}$ for PE and OE, and n^2 for BA. Then, the losses from the subtasks of SOIE and the two auxiliary subtasks are aggregated to form the training objective $\mathcal{J}(\theta)$ of the framework:

$$\mathcal{J}(\theta) = \mathcal{L}^{\text{PE}} + \mathcal{L}^{\text{OE}} + \mathcal{L}^{\text{BA}} + \mathcal{L}^{\text{PB}} + \mathcal{L}^{\text{OB}}. \tag{24}$$

4.8.2 Decoding procedure. At the decoding stage, our goal is to recover the desired lexical facts and relational facts from the output tables. Formally, the decoding process is summarized in Algorithm 1. In the beginning, we extract all the detected objects, lexical facts and predicates from the tag table of object extractor and predicate extractor, denoted as object set S_{obj}, lexical fact set S_{lex} and predicate set S_{pre}, respectively. Considering that synonym and hyponymy may also be the candidate object, we add these two kinds of lexical facts to S_{obj}. Next, we construct a dictionary D_{pre}, which maps the beginning position of each predicate in S_{pre} to the corresponding predicates starting with this position. Similarly, we can obtain the dictionary D_{obj} for S_{obj}. After that, we start to traversing the boundary alignment tag table B to find all the positions with the PB-OB or PE-OE tag, and add them to a beginning set S_{beg} or a ending set S_{end}. For each entry (i, j) in S_{beg}, denoting that i and j may be respectively the beginning position of a valid (*predicate, object*) pair, we iterate all candidate pairs by pairwise combining $D_{\text{pre}}[i]$ and $D_{\text{obj}}[j]$ and checking whether their ending position tuples exists in S_{end}. If so, a new relational fact is extracted and added into the resulting set S_{rel}. Once all the elements in S_{beg} are iterated, this decoding function ends by returning S_{lex} and S_{rel}.

5 EXPERIMENTS

In this section, USE is evaluated against competing models, and we provide a comprehensive analysis of the results. Experiments are carried out on our SOIE dataset, which contains 20k samples for training, 2k samples for development, and 2k samples for test.

5.1 Evaluation Metrics

The evaluation metric measures the micro-average precision (P), recall (R), and F1 score over facts based on the exact match, where an extracted relational fact is considered to be correct only if its predicate and object are the same as that from the ground-truth facts, a lexical fact (description/synonym/hyponymy) is regarded as correct when it's text and type are both correct.

5.2 Implementation Details

We built our model upon the original base BERT model proposed in [9] and optimized it using BertAdam with a learning rate of 2e-5 and a weight decay of 10^{-2}. The max sequence length is set to 200. The window size of CNN for the task-individual encoder is set to 3, and the number of filters is 768, as the hidden layer embedding size of base BERT. Dropout is applied to shared hidden states and task-specific features with a rate of 0.5. All the hyper-parameters are tuned on the dev set. We used Tesla V100 to train the model for at most 40 epochs and chose the model with the best overall F1 score on the dev set to output results on the test set.

5.3 Baseline Models

In order to comprehensively evaluate our proposed model and access the challenges of SOIED, we adapted several state-of-the-art information extraction methods to SOIE, which can be categorized into three classes: rule-based, tagging-based, and generation-based.

Rule-based models use human designed patterns to extract facts. (1) **ZORE** [14] is a widely used Chinese open information extraction system, which identifies relational facts based on a set of pre-constructed syntactic patterns. We used the official program to generate the extraction results of the input sentence and selected the triples with the given entity as the subject as the output relational facts. For lexical fact extraction, we carefully designed several rules based on the sequential patterns and the results of ZORE; (2) **ZORE+** adds a post-processing step to ZORE by introducing a BERT-based binary classification model trained on our labeled dataset to filter incorrect facts.

Generation-based models frame this task as a sequence-to-sequence generation problem. (1) **NeuOIE** [5] is the first attempt to explore how to effectively extract open information based on the encoder-decoder framework. After adapted towards SOIE, the resulting NeuOIE model generates (*predicate, object*) tuples with placeholders in a sequential fashion. The type of lexical fact is considered a special predicate selected from a pre-defined set. More details please refer to [5] and [27]; (2) **Logician** [27] enhances

Table 4: Performance comparison on test set.

Model	Overall			Relational Fact			Lexical Fact		
	P	R	F1	P	R	F1	P	R	F1
ZORE	12.4	7.6	9.4	7.3	5.9	6.5	27.1	9.6	14.2
ZORE+	83.3	7.3	13.4	83.7	5.5	10.8	82.9	9.0	16.2
NeuOIE	72.7	66.1	69.2	69.2	59.8	64.2	76.6	73.8	75.1
Logician	75.2	69.0	72.0	73.0	64.1	68.2	77.8	75.1	76.4
RnnOIE	82.5	70.4	76.0	80.1	61.5	69.6	84.8	81.6	83.1
Pipeline	77.3	77.1	77.2	72.1	70.2	71.1	83.4	85.7	84.5
PATag	82.6	73.5	77.8	80.7	64.7	71.8	84.5	84.4	84.5
ETL	81.4	74.9	78.0	77.6	67.7	72.6	84.9	83.8	84.3
USE	81.7	79.0	**80.3**	79.3	73.3	**76.2**	84.4	86.0	**85.2**

Table 5: Performance comparison in different types of lexical facts on test set.

Model	Description			Synonym			Hyponymy		
	P	R	F1	P	R	F1	P	R	F1
ZORE	24.4	11.4	15.5	37.9	14.5	21.0	22.2	6.1	9.5
ZORE+	77.2	10.8	18.9	87.7	12.7	22.2	84.6	6.0	11.2
NeuOIE	64.0	63.1	63.6	85.9	80.9	83.3	79.5	76.6	78.0
Logician	68.8	65.8	67.3	82.1	80.1	81.1	80.9	78.0	79.4
RnnOIE	75.8	74.0	74.9	86.3	87.4	86.9	89.5	83.2	86.2
Pipeline	76.3	73.5	74.5	85.7	85.6	85.6	86.2	93.0	89.5
PATag	74.7	75.6	75.1	86.6	86.8	86.7	89.4	88.4	88.9
ETL	73.7	75.7	74.7	85.9	84.1	85.0	91.5	88.6	90.0
USE	74.7	75.7	**75.2**	85.7	89.1	**87.4**	89.6	90.7	**90.1**

Table 6: Ablation study of our USE model, evaluated on dev set. Numbers denote the corresponding F1 scores

Objective	Relational Fact	Lexical Fact
USE	76.6	85.0
– Subject Query	72.4	80.2
– Conditional LN	74.9	84.0
– Attentive Tagging	74.6	83.3
– Collaborative Learning	74.0	84.6

NeuralOIE with gated dependency attention and coverage mechanism to exploit syntactical information and solve the problem of under-extraction.

Tagging-based models operate on the word sequences. (1) **RnnOIE** [26] is a unified OIE tagging model, which annotates each word in the sentence to A1, A2, R or O. A1 and A2 denote the subject and object, R is the predicate phrase, and O represents all other words. We modified RnnOIE to fit the SOIE paradigm by removing A1 from the tagging scheme; (2) **Pipeline** first extracts all the candidate predicates and objects based on our proposed Multi-hop tagging scheme and table filling network, then enumerates all possible (*predicate, object*) pairs and classify whether each of these pair is valid or not. These modules are trained jointly by parameter sharing; (3) **PATag** [6] is originally proposed for joint entity and relation extraction, and we altered the tagging scheme towards SOIE. Specifically, it directly tags object according to a query word position and identifies predicates at other positions that have associations with the former; (4) **ETL** first distinguishes all objects and lexical facts that may be related to the target subject entity and then identifies corresponding predicates for each extracted object. This framework is designed by drawing inspiration from the SOTA joint extraction method [31, 35]. The predicate extractor and object extractor share the same structure with our USE model.

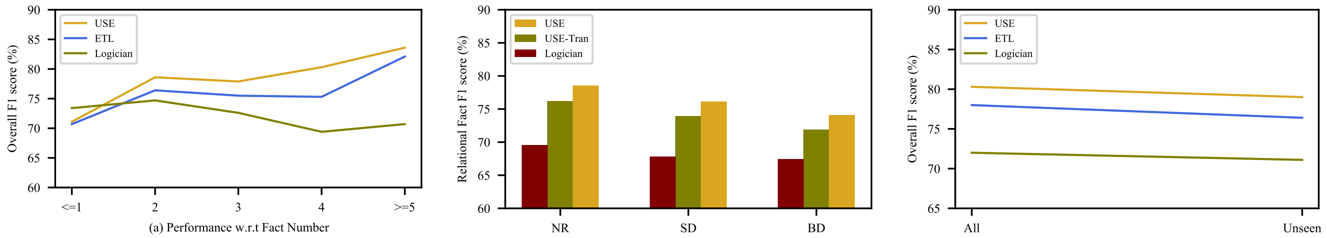
For a fair comparison, we re-implemented generation-based and tagging-based models based on our subject-guided encoder. And a LSTM network is deployed as the decoder of generation-based models. The hyper-parameters of baseline methods have been carefully tuned on the dev set. Considering that RnnOIE and PATagging can only extract continuous elements, we constructed their training set by deleting facts containing discontinuous predicates or objects from the standard set while keeping the test set unchanged. The source code of baseline models will be available to the public.

5.4 Main results

Table 4 summarizes the main results on the SOIED test set. We have the following observations. (1) ZORE performs worst in both precision and recall due to a large number of unmatched facts during extraction, which proves that the pre-defined rules or patterns are not general enough to handle the diverse open-domain corpus; (2) With the aid of post-filtering network, ZORE+ achieves higher precision than ZORE, but the recall is still far from satisfactory; (3) Bypassing hand-crafted patterns, NeuralOIE and Logician cast SOIE into a text generation task and achieves substantial improvements over rule-based methods, indicates that the generalization capability of the neural approach is better than pre-defined rules; (4) Compared with generation-based models, tagging-based approaches boost the overall performance by a considerable margin. We consider that it is because: generation-based models suffer from exposure bias between training and inference, which is the inevitable problem of sequence-to-sequence framework; and the decoding process of generation-based model is to freely select words from the input sentence as the output facts, which is difficult to guarantee the relative word order in the decoded fact elements. (5) For lexical extraction, tagging-based methods use a similar tagging scheme, so their performance is roughly the same in this part. Table 5 shows the detailed comparison results in different types of lexical fact extraction. (6) As Pipeline and ETL can extract discontinuous predicates and objects, they exhibit a remarkable gain in recall compared with RnnOIE and PATag in the relational fact extraction. (7) Our proposed model significantly outperforms competing baselines and achieves the best overall F1 performance. Over Pipeline and ETL, USE achieves an absolute improvement of 5.1% and 3.6% in the F1 score of relational fact extraction, respectively. We hypothesize that this is because Pipeline and ETL are two-stage models in essence, and limited by the error propagation between different extraction stages, while USE has no this issues; and USE introduces the collaborative learning strategy which comprehensively models the interactive relations between modules.

5.5 Model Ablation Study

5.5.1 Effect of Different Components. To demonstrate the effectiveness of each component, we remove one particular part at a



(a) Performance w.r.t Fact Number

(b) Performance w.r.t Discontinuous Type

(c) Performance w.r.t Unseen Fact

Figure 6: Performance on (a) extracting multiple facts, (b) identifying discontinuous facts, and (c) detecting unseen facts.

Table 7: Ablation study of collaborative learning for relational fact extraction on dev set.

Objective	Relational Fact		
	P	R	F1
USE	79.5	74.0	76.6
– R_1: two-way relation between PE and OE	78.6	73.2	75.8
– R_2: one-way relation between BA and PE	79.4	71.3	75.1
– R_3: one-way relation between BA and OE	79.7	71.8	75.5

time to understand its impact on the performance. Concretely, we investigated subject query (by replacing the subject-guided input with the original sentence, the conditional vector is then calculated by averaging the embeddings over the subject tokens in the sentence), conditional layer normalization, attentive table filling (by removing $\mathbf{c}_{i,j}$ from Equation 8) and collaborative learning. From these ablations shown in Table 6, we find that: (1) Removing the subject query hurts the results by 4.2% and 4.8% F1 score in relational and lexical fact extraction, respectively, which indicates that it is vital to let BERT aware of the semantic of the given subject in the encoding process to filter out the inner-sentence noise (irrelevant facts belonging to other subjects); (2) Normalizing the hidden states conditioned on the subject embedding seems an efficient way to give contextual vectors more guidance towards the target subject entity; (3) Benefiting from the attention mechanism, our table filling network can effectively collect useful clues related to the depended positions from the contextual representations, thus achieving better performance; (4) Collaborative learning strategy brings a remarkable improvement (2.6%) in relational fact extraction, which demonstrates that our predicate extractor, object extractor, and boundary alignment module work in the mutual promotion way, and exchanging informative clues among three subtasks is beneficial to capture their interactive relations; As for lexical fact extraction, the performance gain brought by collaborative learning is limited. We guess this is because the object extractor itself has been able to recognize lexical fact effectively;

5.5.2 Effect of Different Relations in Collaborative Learning. One can observe the performance gain boosted by collaborative learning, especially in relational fact extraction, from Table 6. To investigate the underlying reason, we conduct an ablation study by removing one interactive relation from collaborative learning at a time. As shown in Table 7, all three relations contribute to the final performance of relational fact extraction, which shows the effectiveness of collaborative learning. When removing the two-way relation R_1 between predicate extractor and object extractor, the relational fact extraction F1 score drops from 76.6 to 75.8, indicating the inherent association underlying the PE and OE subtasks. R_2 and R_3 help the most by greatly improving the recall, proving that predicate extractor and object extractor can work together to provide crucial boundary clues for boundary alignment tagging.

5.6 Performance Analysis

5.6.1 Performance on Multiple Facts. We compare the models'ability of extracting multiple facts in a sentence. We divide the samples of SOIE test set into 5 categories, which respectively indicate its number of facts is ≤ 1, 2, 3, 4 and ≥ 5. The results are shown in Figure 6(a). It can be observed that our USE gains great improvements compared with other models in extracting multiple facts. When extracting information from instances that contains ≤ 1 fact, Logician model achieve the best performance. However, when the number of facts increases, the performance of Logician model decreases significantly. In contrast, USE attains consistently strong performance over all five classes. This experiment fully demonstrates the advantages of our proposed model in dealing with complex extracting situation.

5.6.2 Performance on Discontinuous Relational Facts. To verify the ability of our model in handling the discontinuous problem, we conduct further experiments by dividing the test set into three categories: Normal (NR), SingleDiscontinuous (SD), and BothDiscontinuous (BD). Specifically, a sample belongs to the NR class if none of its facts has a discontinuous predicate or object. If one of the predicate and object in a relational fact is discontinuous, the sample will be added to the SD set. If both predicate and object in a relational fact are discontinuous, the sample will be added to the BD set. For a comprehensive evaluation, we also implement another competitive baseline by replacing our predicate extractor and object extractor with the latest discontinuous NER model Tran [7] based on the official code[5], named as USE-Tran. Note that Tran is restricted to processes one data at a time due to the transition-based architecture, which means it cannot be trained in the batch

[5] https://github.com/daixiangau/acl2020-transition-discontinuous-ner

Table 8: Computational cost of different methods, B/s refers to the number of batches can be processed per second.

	RnnOIE	NeuOIE	Logician	USE
Parameter Number	106.7M	109.6M	111.5M	121.8M
Test-time Speed	7.3 B/s	4.2 B/s	4.0 B/s	3.2 B/s

mode. So we train the different modules of USE-Tran individually using the same encoder as ours. From Figure 6(b), we find that: (1) Performance of all models on NR, SD, and BD presents a decreasing trend, reflecting the increasing difficulty of extracting relational facts with different discontinuous patterns. (2) Our proposed model is least sensitive to the discontinuous situation and consistently achieves better results than USE-Tran and Logician, demonstrating the effectiveness of our multi-hop tagging scheme and table filling network in addressing the discontinuous problem.

5.6.3 Performance on Unseen Facts. The ultimate objective of SOIE is to collect new facts for the given entity. So we propose to validate the models' capability in extracting unseen facts. In this experiment, we construct a new test set: if one relational fact or lexical fact exists in the training set, the sample will be removed from the original test set. Figure 6(c) gives the F1 score comparison on all and unseen facts. We can observe that the performance gap between two different settings is negligible (about 1.5%), which indicates that instead of just memorizing the frequent patterns, the models trained on SOIED indeed learns general knowledge about fact extraction and can generalize to unseen facts, demonstrating the openness and diversity of our annotated dataset.

5.6.4 Analysis on Computational Cost. We analyze time complexity for all neural baselines used in this paper. Pipeline, PATag, and ETL have O(n^2) time-complexity, which is similar to our USE model. Given a sentence of length n, RnnOIE runs sequence tagging only once to extract non-discontinuous facts bringing O(n) time-complexity. In NeuOIE and Logician, time complexity depends on the number of facts related to the given subject in the sentence, denoted as O(d). This means our model is more time-consuming than RnnOIE, NeuOIE and Logician in theory (O(n^2) vs. O(n) or O(d)). In order to evaluate their actual computational cost, we run them on the SOIE test set with the same batch size in a GPU server and present the results in Table 8. The test-time speed of our USE model is slower than other three models, but they are still of the same order of magnitude. We consider that it is because (1) these four models all use BERT as the backbone, which takes up the main part of all parameters and the main cost of time; (2) although the time complexity of NeuOIE and Logician are both $O(d)$, their decoding process can not be parallelized and requires beam search, so the actual running speed is not outstanding. Considering the distinct performance advantage of USE presented in previous subsections, we think its computational cost is acceptable.

5.7 Error analysis

Although the proposed USE model outperforms all baseline methods and achieves state-of-the-art results, the performance is still far from satisfactory, particularly in relational fact extraction. We provide insights into specific reasons for the mistakes made by USE by randomly selecting 100 incorrect facts and summarizing the prediction errors, which pinpoints the model limitation and the direction of improvement for future work. Specifically, we categorize the incorrect facts into six classes: (1) Entirely incorrect fact; (2) Incorrect subject: the fact corresponds to other entities in the sentence rather than the given subject; (3) Incorrect lexical fact: the boundary of lexical fact is wrong; (4) Correct predicate, incorrect object: the predicate is correct, but the object is false; (5) Correct object, incorrect predicate: the object is correct, but the predicate is false; (6) Incorrect pairing: the predicate and object are both correct, but they are mismatched.

Table 9: Prediction error by percentage.

Error Type	Percentage
Entirely incorrect fact	23%
Incorrect subject	32%
Incorrect lexical fact	17%
Correct predicate, incorrect object	14%
Correct object, incorrect predicate	12%
Incorrect pairing	2%

Table 9 shows the percentage of each category. From the results, one can observe that the fact with incorrect subject roughly accounts for 32% of the error set, indicating that our model fail to distinguish different facts for different subjects when the structure of the sentence is complicated. More linguistic knowledge, such as syntactical information, should be introduced to solve this problem. The incorrect lexical fact issue causes 17% of errors. We find most of the failures stem from the description boundary error. Intuitively, a description is usually long and complex, so its extraction is more challenging than other lexical facts. It would be interesting to see if designing a more precise tagging scheme, e.g., binary tagging [31, 35], can improve the performance. About 26% of the errors are due to the incorrect predicate or object. Most failure cases of this issue are caused by mistakenly merging separated spans as the discontinuous elements. We think enhancing the ability of our multi-hop tagging based table filling network to capture the semantic dependency between spans may be a promising improvement direction. Another encouraging observation is that there are few errors due to incorrect pairing, which again demonstrates the effectiveness of our boundary alignment module. Overall, the challenge of extracting target-specific facts is far from solved. How to accurately identify the related facts of the target entity in complex sentences is still an opening problem.

6 CONCLUSION

We present a new task named Semi-Open Information Extraction (SOIE) and an accompany annotated dataset named SOIED in this work. The new task requires the model to discover domain-independent facts towards a target entity from web text, which poses specific challenges given the diverse and complex corpus. We also propose a unified framework, USE, to provide some meaningful explorations for this task. USE first builds the subject-aware contextual representations based on subject-guided input and conditional layer normalization mechanism, and then transforms the SOIE task into three table filling problems with tailor-designed tagging schemes. Extensive experiments show that the proposed model achieves significant improvements on SOIED compared with competitive baseline methods, but the challenge still remains.

Interesting future directions including: N-ary SOIE, document-level SOIE, and aligning extraction results with knowledge bases. We believe the new task and new algorithm will innovate the research community on new research ideas and directions for information extraction.

REFERENCES

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

[2] Ermei Cao, Difeng Wang, Jiacheng Huang, and Wei Hu. 2020. Open Knowledge Enrichment for Long-tail Entities. In *Proceedings of The Web Conference 2020*. 384–394.

[3] Junkun Chen, Xipeng Qiu, Pengfei Liu, and Xuanjing Huang. 2018. Meta multi-task learning for sequence modeling. *Proceedings of 2018 AAAI Conference on Artificial Intelligence*.

[4] Zhuang Chen and Tieyun Qian. 2020. Relation-Aware Collaborative Learning for Unified Aspect-Based Sentiment Analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 3685–3694.

[5] Lei Cui, Furu Wei, and Ming Zhou. 2018. Neural Open Information Extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. 407–413.

[6] Dai Dai, Xinyan Xiao, Yajuan Lyu, Shan Dou, Qiaoqiao She, and Haifeng Wang. 2019. Joint extraction of entities and overlapping relations using position-attentive sequence labeling. In *Proceedings of 2019 AAAI Conference on Artificial Intelligence*. 6300–6308.

[7] Xiang Dai, Sarvnaz Karimi, Ben Hachey, and Cecile Paris. 2020. An Effective Transition-based Model for Discontinuous NER. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

[8] Harm de Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron Courville. 2017. Modulating early visual processing by language. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 6597–6607.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*. 4171–4186.

[10] Hady Elsahar, Elena Demidova, Simon Gottschalk, Christophe Gravier, and Frederique Laforest. 2017. Unsupervised open relation extraction. In *Proceedings of 2017 European Semantic Web Conference*. 12–16.

[11] Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. 2008. Open information extraction from the web. *Commun. ACM* 51, 12, 68–74.

[12] Xu Han, Tianyu Gao, Yankai Lin, Hao Peng, Yaoliang Yang, Chaojun Xiao, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2020. More Data, More Relations, More Context and More Openness: A Review and Outlook for Relation Extraction. *arXiv preprint arXiv:2004.03186* (2020).

[13] Tianxing He, Jingzhao Zhang, Zhiming Zhou, and James Glass. 2019. Quantifying exposure bias for neural language generation. *arXiv preprint arXiv:1905.10617* (2019).

[14] Shengbin Jia, Shijia E, Maozhen Li, and Yang Xiang. 2018. Chinese open relation extraction and knowledge base establishment. *ACM Transactions on Asian and Low-Resource Language Information Processing* (2018), 1–22.

[15] Keshav Kolluru, Samarth Aggarwal, Vipul Rathore, Soumen Chakrabarti, et al. 2020. IMoJIE: Iterative Memory-Based Joint Open Information Extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

[16] Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. 302–308.

[17] Guiliang Liu, Xu Li, Miningming Sun, and Ping Li. 2020. An Advantage Actor-Critic Algorithm with Confidence Exploration for Open Information Extraction. In *Proceedings of the 2020 SIAM International Conference on Data Mining*. SIAM, 217–225.

[18] Guiliang Liu, Xu Li, Jiakang Wang, Mingming Sun, and Ping Li. 2020. Extracting Knowledge from Web Text with Monte Carlo Tree Search. In *Proceedings of The Web Conference 2020*. 2585–2591.

[19] Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-BERT: Enabling Language Representation with Knowledge Graph.. In *Proceedings of 2020 AAAI Conference on Artificial Intelligence*. 2901–2908.

[20] Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, Tag, Realize: High-Precision Text Editing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. 5057–5068.

[21] Mausam Mausam. 2016. Open information extraction systems and downstream applications. In *Proceedings of the twenty-fifth international joint conference on artificial intelligence*. 4074–4077.

[22] Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2018. A Survey on Open Information Extraction. In *Proceedings of the 27th International Conference on Computational Linguistics*. 3866–3878.

[23] Xiang Ren, Zeqiu Wu, Wenqi He, Meng Qu, Clare R Voss, Heng Ji, Tarek F Abdelzaher, and Jiawei Han. 2017. Cotype: Joint extraction of typed entities and relations with knowledge bases. In *Proceedings of the 26th International Conference on World Wide Web*. 1015–1024.

[24] Arpita Roy, Youngja Park, Taesung Lee, and Shimei Pan. 2019. Supervising Unsupervised Open Information Extraction Models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. 728–737.

[25] Stephen Soderland, Brendan Roof, Bo Qin, Shi Xu, Mausam, and Oren Etzioni. 2010. Adapting Open Information Extraction to Domain-Specific Relations. *AI Magazine* 31, 3 (2010), 93–102. https://doi.org/10.1609/aimag.v31i3.2305

[26] Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics*. 885–895.

[27] Mingming Sun, Xu Li, Xin Wang, Miao Fan, Yue Feng, and Ping Li. 2018. Logician: A unified end-to-end neural approach for open-domain information extraction. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 556–564.

[28] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*. 4149–4158.

[29] Linlin Wang, Zhu Cao, Gerard De Melo, and Zhiyuan Liu. 2016. Relation classification via multi-level attention cnns. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. 1298–1307.

[30] Yu Wang, Yun Li, Ziye Zhu, Hanghang Tong, and Yue Huang. 2020. Adversarial Learning for Multi-Task Sequence Labeling With Attention Mechanism. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28 (2020), 2476–2488.

[31] Zhepei Wei, Jianlin Su, Yue Wang, Yuan Tian, and Yi Chang. 2020. A novel cascade binary tagging framework for relational triple extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 1476–1488.

[32] Ruidong Wu, Yuan Yao, Xu Han, Ruobing Xie, Zhiyuan Liu, Fen Lin, Leyu Lin, and Maosong Sun. 2019. Open relation extraction: Relational knowledge transfer from supervised data to unsupervised data. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. 219–228.

[33] Ji Xin, Hao Zhu, Xu Han, Zhiyuan Liu, and Maosong Sun. 2018. Put it back: Entity typing with language model enhancement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 993–998.

[34] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2369–2380.

[35] Bowen Yu, Zhenyu Zhang, Xiaobo Shu, Yubin Wang, Tingwen Liu, Bin Wang, and Sujian Li. 2020. Joint Extraction of Entities and Relations Based on a Novel Decomposition Strategy. In *Proceedings of the 24th European Conference on Artificial Intelligence*.

[36] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of the 25th International Conference on Computational Linguistics*. 2335–2344.

[37] Junlang Zhan and Hai Zhao. 2019. Span Model for Open Information Extraction on Accurate Corpus. *Proceedings of 2019 AAAI Conference on Artificial Intelligence*.

[38] Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. 2019. Bridging the Gap between Training and Inference for Neural Machine Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 4334–4343.

[39] Yuhao Zhang, Peng Qi, and Christopher D Manning. 2018. Graph Convolution over Pruned Dependency Trees Improves Relation Extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2205–2215.

[40] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced Language Representation with Informative Entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 1441–1451.