

Unified Structure Generation for Universal Information Extraction

Yaojie Lu^{1,4,*}, Qing Liu^{1,4,*}, Dai Dai³, Xinyan Xiao³, Hongyu Lin^{1,†},
Xianpei Han^{1,2,5}, Le Sun^{1,2,†}, Hua Wu³

¹Chinese Information Processing Laboratory ²State Key Laboratory of Computer Science
Institute of Software, Chinese Academy of Sciences, Beijing, China

³Baidu Inc., Beijing, China

⁴University of Chinese Academy of Sciences, Beijing, China

⁵Beijing Academy of Artificial Intelligence, Beijing, China

{yaojie2017, liuqing2020, hongyu, xianpei, sunle}@iscas.ac.cn

{daidai, xiaoxinyan, wu_hua}@baidu.com

Abstract

Information extraction suffers from its varying targets, heterogeneous structures, and demand-specific schemas. In this paper, we propose a unified text-to-structure generation framework, namely UIE, which can universally model different IE tasks, adaptively generate targeted structures, and collaboratively learn general IE abilities from different knowledge sources. Specifically, UIE uniformly encodes different extraction structures via a structured extraction language, adaptively generates target extractions via a schema-based prompt mechanism – structural schema instructor, and captures the common IE abilities via a large-scale pre-trained text-to-structure model. Experiments show that UIE achieved the state-of-the-art performance on 4 IE tasks, 13 datasets, and on all supervised, low-resource, and few-shot settings for a wide range of entity, relation, event and sentiment extraction tasks and their unification. These results verified the effectiveness, universality, and transferability of UIE¹.

1 Introduction

Information extraction (IE) aims to identify and structure user-specified information from unstructured texts (Andersen et al., 1992; Grishman, 2019). IE tasks are highly diversified due to its varying targets (entity, relation, event, sentiment, etc.), heterogeneous structures (spans, triplets, records, etc.), and demand-specific schemas (Grishman and Sundheim, 1996; Mitchell et al., 2005; Ji and Grishman, 2011).

Currently, most IE approaches are *task-specialized*, which leads to dedicated architectures, isolated models, and specialized knowl-

Task	Schema	Instance												
Entity	PER: _ ORG: _	In 1997, Steve was excited to become the CEO of Apple.												
Relation	(_, Work for, _)	In 1997, Steve was excited to become the CEO of Apple.												
Event	<table border="1"> <tr> <th>Type</th><th>Start</th><th>Position</th></tr> <tr> <td>employee</td><td></td><td></td></tr> <tr> <td>employer</td><td></td><td></td></tr> <tr> <td>...</td><td></td><td></td></tr> </table>	Type	Start	Position	employee			employer			...			In 1997, Steve was excited to become the CEO of Apple.
Type	Start	Position												
employee														
employer														
...														
Sentiment	Positive { Opinion: _; Target: _ }	In 1997, Steve was excited to become the CEO of Apple.												

(a) Task-specialized IE

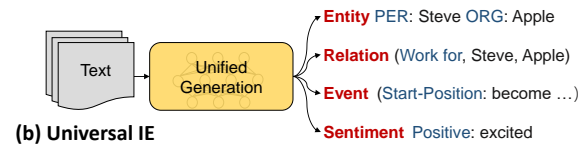


Figure 1: From (a) Task-specialized IE: different tasks, different structures, different schemas to (b) Universal IE: unified modeling via structure generation.

edge sources for different IE task. These task-specialized solutions greatly hinder the rapid architecture development, effective knowledge sharing, and quick cross-domain adaptation of IE systems. First, it is very complicated to develop dedicated architectures for a large amount of IE tasks/settings/scenarios. Second, learning isolated models severely restricts the knowledge sharing between related tasks and settings. Finally, it is costly and time-consuming to construct data sets and knowledge sources specialized for different IE tasks. Therefore, it will be of great benefit to develop a universal IE architecture that can uniformly model different IE tasks, adaptively predict heterogeneous structures and effectively learn from various resources, which we referred to as **Universal IE**.

Fundamentally, all IE tasks can be modeled as text-to-structure transformations, with different

*Part of this work was done when Yaojie Lu and Qing Liu interned at Baidu.

†Corresponding authors.

¹<https://universal-ie.github.io>

tasks correspond to different structures. For example, as shown in Figure 1, an entity is a named span structure, an event is a schema-defined record structure. These text-to-structure transformations in IE can be further decomposed into several atomic transformation operations: 1) *Spotting*, which locates the desirable spans concerning to given specific semantic types (Kripke and Munitz, 1971; Chen and Yuille, 2004). For example, locating span “Steve” as a *Person* entity and locating “excited” as a sentiment expression. 2) *Associating*, which connects spans by assigning them with semantic roles in pre-defined schemas (Onyshkevych, 1994; Milward and Thomas, 2000). For example, associating “Steve” and “Apple” by assigning them as the *Arg1* and the *Arg2* of a *Work-for* relation. In this way, different IE tasks can be decomposed into a sequence of atomic text-to-structure transformations, and all IE models share the same underlying spotting and associating abilities. For example, entity extraction can be viewed as spotting mention spans of corresponding entity types, while event detection can be reformulated as spotting triggers spans with event types. And the spotting abilities can be shared between these two tasks.

Based on the above observations, we propose UIE, a unified text-to-structure generation architecture that can universally model different IE tasks, adaptively generate targeted structures, and collaboratively learn general IE abilities from different knowledge sources. Specifically, to model heterogeneous IE structures, we design a structural extraction language (SEL) that can effectively encode different IE structures into a uniform representation, so that various IE tasks can be universally modeled in the same text-to-structure generation framework. To adaptively generate targeted structures for different IE tasks, we propose structural schema instructor (SSI), a schema-based prompt mechanism which controls what to spot, what to associate, and what to generate in UIE. To learn common IE abilities for UIE, we pre-train UIE on large-scale, heterogeneous datasets mined from easily accessible web sources. The large-scale pre-trained UIE model provides a solid foundation for knowledge sharing and quick adaptation to new IE settings, and significantly boosts the IE performance in all supervised, low-resource, and few-shot settings.

We conduct experiments on 13 datasets of 4 main IE tasks (entity/relation/event/sentiment ex-

traction and their unification), and supervised, low-resource, and few-shot settings. Experiment results show that UIE achieves significant improvements in all settings. On supervised settings, UIE achieved 1.42% F1 scores improvements over the state-of-the-art, task-specialized architectures on all datasets. On few-shot and low-resource settings, UIE exhibits strong on-demand adaptation ability: it outperforms baselines dramatically by a large margin. These results verified the effectiveness, universality, and transferability of UIE across different IE tasks, settings, and scenarios.

The main contributions of this paper are:

1) We propose UIE, a unified text-to-structure generation architecture that can universally model different IE tasks, adaptively generate targeted structures, and collaboratively learn general IE abilities from different knowledge sources.

2) We design a unified structure generation network, which encodes heterogeneous IE structures into a uniform representation via a structural extraction language, and controls the UIE model which to spot, which to associate, and which to generate via structural schema instructor mechanism.

3) We pre-train a large-scale text-to-structure generation model via a unified pre-training algorithm. To the best of our knowledge, this is the first text-to-structure pre-trained extraction model, which can benefit future IE studies.

2 Unified Structure Generation for Universal Information Extraction

Information extraction tasks can be formulated as text-to-structure problems, where different IE tasks correspond to different structures. This paper aims to uniformly model the text-to-structure transformations of different IE tasks via a single framework, i.e., different structure transformations will share the same underlying operations and different transformation abilities in a universal model. Formally, given a specific pre-defined schema s and texts x , a universal IE model needs to generate a structure that contains the desirable structural information in the text x indicated by the schema s .

Generally, there are two main challenges here. Firstly, due to the diversity of IE tasks, there are many different target structures to extract, e.g., entity, relation, event, etc. Secondly, IE tasks are often demand-specific which are defined using different schemas, therefore we need to adaptively control the extraction process.

```

(
  (Spot Name: Info Span
    (Asso Name: Info Span)
    (Asso Name: Info Span)
  )
)

```

(a) Structured extraction language (SEL) for Universal IE.

```

(
  (person: Steve
    (work for: Apple)
  )
  (start-position: became
    (employee: Steve)
    (employer: Apple)
    (time: 1997)
  )
  (organization: Apple)
  (time: 1997)
)

```

(b) The SEL representation of the extraction structure of “Steve became CEO of Apple in 1997.”, where the relation structure is marked **blue**, the event structure is marked **red**, and the rest are entities.

Figure 2: Illustrations of structured extraction language.

In this section, we describe how to jointly formulate, learn, and conduct various IE tasks in a unified text-to-structure generation architecture, named **UIE**. Specifically, we first design structured extraction language (SEL) to uniformly encode heterogeneous extraction structures, i.e., encode entity, relation, event into a unified representation. Then we describe structural schema instructor (SSI), a schema-based prompt mechanism that controls the UIE model which to spot, which to associate, and which to generate for different extraction settings. The details are as follows.

2.1 Structured Extraction Language for Uniform Structure Encoding

This section describes how to encode heterogeneous IE structures into a uniform representation. Based on the above discussions, IE structure generation can be decomposed into two atomic operations:

1. **Spotting** indicates locating target information pieces from the sentence, e.g., the entity and the trigger word in the event.
2. **Associating** indicates connecting different information pieces based on the desirable associations, e.g., the relation between entity pair or the role between event and its argument.

Then different IE structures can be represented as a combination of atomic structure generation operations.

Concretely, we design a unified structured ex-

traction language (SEL), which encodes different IE structures via the spotting-associating structure. As shown in Figure 2a, each SEL expression contains three types of semantic units: 1) **SPOTNAME** represents there is a specific information piece with the type of spot name existing in the source text; 2) **ASSONAME** indicates there exists a specific information piece in the source text that is with the AssoName association to its upper-level Spotted information in the structure; 3) **INFOSPAN** represents the text span corresponding to the specific spotting or associating information piece in the source text. Furthermore, “.” in the SEL indicates the mapping from InfoSpan to its spotting or associating names, and the two structure indicators “(” and “)” are used to form the hierarchical structure between the extracted information.

Using SEL, Figure 2b shows how to represent entity, relation, and event structures. There are three entities and each entity is represented as a spotting structure such as “person:Steve”, “organization:Apple”, and “time:1997”; one relation which is represented as an association structure between “Steve” and “Apple” with association name work for; and one event which is represented as an association structure, where the trigger is a spotting structure “start-position:became”, and its arguments are associated with the trigger: Steve as employee, Apple as employer, 1997 as time.

We can see that, SEL have the advantages that: 1) uniformly encodes varying IE structures, therefore different IE tasks can be modeled as the same text-to-structure generation process; 2) efficiently represents all extraction results of a sentence in the same structure, thus can perform joint extraction naturally; 3) the output structure of generation is very compact, which greatly reduce the complexity of decoding.

For example, the two different tasks entity recognition and event detection can be revisited using the same “(SpotName: InfoSpan)” grammar. While both relation extraction and event extraction can be formulated using the grammar “(SpotName: InfoSpan (AssoName: InfoSpan), ...)”, even they are with totally different binary “entity-relation-entity” and N-ary “event-arguments” structures. Such a unified structured extraction language enables UIE to learn from and adapt to different IE tasks without designing task-specialized architectures, because these IE tasks are all universally formulated as the transformation from texts to SEL representations.

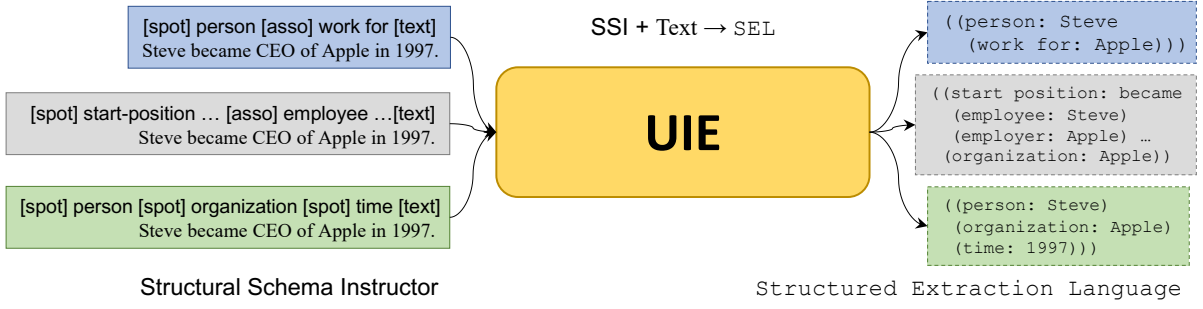


Figure 3: The overall framework of UIE.

2.2 Structural Schema Instructor for Controllable IE Structure Generation

Using SEL, UIE can uniformly generate different IE structures. However, because different IE tasks have different schemas, one challenge here is how to adaptively control which information we want to generate during extraction. For example, given a sentence “Steve became CEO of Apple in 1997.”, an entity recognition system will generate “((person: Steve) (organization: Apple) (Time: 1997))”, and an event extraction system will generate “((start position: became (employee: Steve) (employer: Apple)))”. To this end, we propose structural schema instructor (SSI), a schema-based prompt mechanism that controls which kinds of information need to be spotted and associated.

Figure 3 shows the overall framework of UIE. Formally, UIE takes the given structural schema instructor (s) and the text sequence (x) as input, and generates the linearized SEL (y) which contains the extracted information from x based on schema s :

$$y = \text{UIE}(s \oplus x) \quad (1)$$

where $x = [x_1, \dots, x_{|x|}]$ is the text sequence, $s = [s_1, \dots, s_{|s|}]$ is the structural schema instructor, and $y = [y_1, \dots, y_{|y|}]$ is a SEL sequence that can be easily converted into the extracted information record.

2.2.1 Structural Schema Instructor

To describe the extraction target of a task, the structural schema instructor constructs a schema-based prompt and uses it as a prefix during generation.

Specifically, corresponding to the spotting-association structure, the structural schema instructor contains three types of token segments: 1) SPOTNAME: the targeted spotting name in the specific information extraction task, such as “person” in the NER task; 2) ASSONAME: the targeted association name, such as “work for” in the relation extraction task; 3) Special Symbols ([spot], [asso],

[text]) which are added before each SPOTNAME, ASSONAME, and input text sequence. All tokens in SSI are concatenated and put before the original text sequences. As shown in Figure 3, the entire input for UIE is in the form of:

$$\begin{aligned} s \oplus x &= [s_1, s_2, \dots, s_{|s|}, x_1, x_2, \dots, x_{|x|}] \\ &= [[\text{spot}], \dots [\text{spot}], \dots, \\ &\quad [\text{asso}], \dots, [\text{asso}], \dots, \\ &\quad [\text{text}], x_1, x_2, \dots, x_{|x|}] \end{aligned} \quad (2)$$

For example, the SSI “[spot] person [spot] company [asso] work for [text]” indicates extracting records of the relation schema “the person works for the company” from the sentence. Given the SSI s , UIE first encodes the text x , then generates the target record y in linearized SEL using an encoder-decoder-style architecture.

We found that the schema-based prompt can: 1) effectively guide the SEL generation of UIE, so that the general IE ability can be transferred to new IE tasks; 2) adaptively control which to spot, which to associate, and which to generate, so that semantic knowledge across different labels and tasks can be better shared.

2.2.2 Structure Generation with UIE

Given SSI s and text x as input, UIE extracts targeted information by generating a linearized SEL. We formulate this text-to-SEL generation process using an encoder-decoder-style architecture. Given the raw text sequence x and the schema instructor s , UIE first compute the hidden representation $\mathbf{H} = [s_1, \dots, s_{|s|}, \mathbf{x}_1, \dots, \mathbf{x}_{|x|}]$ of each token:

$$\mathbf{H} = \text{Encoder}(s_1, \dots, s_{|s|}, x_1, \dots, x_{|x|}) \quad (3)$$

where $\text{Encoder}(\cdot)$ is a Transformer encoder. Then UIE will decode the input text into a linearized SEL in an auto-regressive style. At the step i of decoding, UIE generates the i -th token y_i in the SEL

sequence and the decoder state \mathbf{h}_i^d as following:

$$y_i, \mathbf{h}_i^d = \text{Decoder}([\mathbf{H}; \mathbf{h}_1^d, \dots, \mathbf{h}_{i-1}^d]) \quad (4)$$

$\text{Decoder}(\cdot)$ is a transformer decoder, which predicts the conditional probability $p(y_i|y_{<i}, x, s)$ of token y_i . Finally, $\text{Decoder}(\cdot)$ finishes prediction when outputting the end symbol $\langle \text{eos} \rangle$, then we convert the predicted SEL expression into the extracted information record.

Compared with previous IE studies which treat labels as specific symbols, the text-to-structure generation paradigm treats labels as natural language tokens. By verbalizing and generating labels and structures, our method can effectively transfer knowledge from pre-trained language models such as BART (Lewis et al., 2020), T5 (Raffel et al., 2020), and related tasks can easily share knowledge because their labels have similar semantics (e.g., *location* and *place*) and share common label-text associations (e.g., *victim* for different event types).

3 Pre-training and Fine-tuning for UIE

In this section, we describe: 1) how to pre-train a large-scale UIE model which captures common IE abilities for different IE tasks; 2) how to adapt UIE to different IE tasks in different settings via quick fine-tuning. Specifically, we first collect several large-scale datasets from the Web, including structured (e.g., knowledge bases), unstructured (e.g., raw texts), and parallel (e.g., Wikipedia-Wikidata links) data, then we uniformly pre-train our UIE model on these heterogeneous datasets. Finally, we adapt the pre-trained UIE model to the specific downstream IE tasks via on-demand fine-tuning. We found that the pre-trained UIE model provides a solid foundation for capturing, sharing, and transferring knowledge between different IE tasks, and new IE tasks can be effectively solved because UIE learns general IE ability.

3.1 Pre-training Corpus Construction

UIE needs to encode the text, map text to structure, and decode valid structure. Therefore, we collect a large-scale pre-training corpus from easily accessible web data sources (more details are in the appendix):

$\mathcal{D}_{\text{pair}}$ is the text-structure parallel data, where each instance is a parallel pair (token sequence x , structured record y). We collect large-scale parallel text-structure pairs by aligning Wikidata with

English Wikipedia. $\mathcal{D}_{\text{pair}}$ is used to pre-train the text-to-structure transformation ability of UIE.

$\mathcal{D}_{\text{record}}$ is the structure dataset where each instance is structured record y . We collect structured records from ConceptNet (Speer et al., 2017) and Wikidata. $\mathcal{D}_{\text{record}}$ is used to pre-train the structure decoding ability of UIE.

$\mathcal{D}_{\text{text}}$ is the unstructured text dataset, and we use all plain texts in English Wikipedia. $\mathcal{D}_{\text{text}}$ is used to pre-train the semantic encoding ability of UIE.

3.2 Pre-training

We pre-train UIE using three sequence generation tasks with above mentioned pre-training datasets.

Text-to-Structure Pre-training using $\mathcal{D}_{\text{pair}}$. To capture the fundamental text-to-structure mapping ability, we pre-train UIE using $\mathcal{D}_{\text{pair}} = \{(x, y)\}$. Specifically, for each parallel pair (x, y) , we extract the spot type s_{s+} and the associating type s_{a+} in the record y as the positive schema $s_+ = s_{s+} \cup s_{a+}$. However, we found that if we only feed UIE with a positive schema, it will only simply remember the triplet in the pre-training data. To learn general mapping ability, we also automatically construct negative schemas for each pair, i.e., we first sample negative spots s_{s-} and negative association set s_{a-} , then concatenate meta-schema $s_{\text{meta}} = s_+ \cup s_{s-} \cup s_{a-}$, and construct the final extraction target. For example, *person* and *work for* is the positive schema in the record “((person: Steve (work for: Apple)))”, and we sample *vehicle* and *located in* as the negative schema to construct meta-schema. Finally, the objective of text-to-structure pre-training is:

$$\mathcal{L}_{\text{Pair}} = \sum_{(x,y) \in \mathcal{D}_{\text{pair}}} -\log p(y|x, s_{\text{meta}}; \theta_e, \theta_d) \quad (5)$$

where θ_e and θ_d are the parameter of encoder and decoder, respectively.

Structure Generation Pre-training with $\mathcal{D}_{\text{record}}$.

To pre-train the ability of generating valid structures defined by SEL and schemas, we pre-train UIE on $\mathcal{D}_{\text{record}}$. We pre-train UIE decoder as an structured language model, where each record in $\mathcal{D}_{\text{record}}$ is an expression of SEL:

$$\mathcal{L}_{\text{Record}} = \sum_{y \in \mathcal{D}_{\text{record}}} -\log p(y_i|y_{<i}; \theta_d) \quad (6)$$

By pre-training for structure generation, the decoder can capture the regularity of SEL and the interactions between different labels.

Retrofitting Semantic Representation using $\mathcal{D}_{\text{text}}$

During text-to-structure pre-training, we continually pre-train UIE also with the masked language model tasks (Raffel et al., 2020) on $\mathcal{D}_{\text{text}}$ to retrofit semantic representations of UIE. Specifically, we add span corruption based mask language modeling objective in the pre-training stage:

$$\mathcal{L}_{\text{Text}} = \sum_{x \in \mathcal{D}_{\text{text}}} -\log p(x''|x'; \theta_e, \theta_d) \quad (7)$$

where x' is the corrupted source text and x'' is corrupted target spans. We found this pre-training can effectively alleviate the catastrophic forgetting of token semantics especially on SPOTNAME and ASSONAME tokens.

Final Pre-training Criteria. We initialize UIE-base and UIE-large with T5-v1.1-base and T5-v1.1-large (Raffel et al., 2020), and the model architectures are shown in Table 7. The final objective is the combine of the above tasks:

$$\mathcal{L} = \mathcal{L}_{\text{Pair}} + \mathcal{L}_{\text{Record}} + \mathcal{L}_{\text{Text}} \quad (8)$$

For implementation, we uniformly represent all pre-training data as triplets. For text data (x) in $\mathcal{D}_{\text{text}}$, we build a triplet (None, x' , x'') where x' is the corrupted source text and x'' is corrupted spans. For text-record data (x, y) in $\mathcal{D}_{\text{pair}}$, we construct (s, x, y) by sampling the meta-schema s for each text-record pair. For record data (y) in $\mathcal{D}_{\text{record}}$, we take (None, None, y) as the input triplet. We randomly pack instances for different tasks in one batch, and details are shown in Algorithm 1 in the appendix.

3.3 On-Demand Fine-tuning

Given the pre-trained UIE model, we can quickly adapt it to different IE tasks and settings through model fine-tuning. Given a labeled corpus $\mathcal{D}_{\text{task}} = \{(s, x, y)\}$, we fine-tune the UIE model using teacher-forcing cross-entropy loss:

$$\mathcal{L}_{\text{FT}} = \sum_{(s, x, y) \in \mathcal{D}_{\text{Task}}} -\log p(y|x, s; \theta_e, \theta_d) \quad (9)$$

To alleviate the exposure bias (Ranzato et al., 2016; Zhang et al., 2020) of the auto-regressive model during decoding, we also design a **Rejection Mechanism** for effective fine-tuning. Specifically, given an instance (s, x, y), we first encode y using SEL language, then we randomly insert several [NULL] unit with negative SPOTNAME and ASSONAME: (SPOTNAME, [NULL]) and (ASSONAME, [NULL]) into the ground-truth SEL with the

SSI	<spot> person ... <spot> facility <asso> ... <text>
Text	Steve became CEO of Apple in 1997.
SEL	((person: Steve (work for: Apple)) (start-position: ...
+ RM	((person: Steve (work for: Apple)) (<i>facility: [NULL]</i>) ...

Table 1: An example of rejection mechanism (RM), here “(facility: [NULL])” is the injected rejection noise during learning stage, and the [NULL]-valued span will be ignored during inference stage.

probability of p_e . For example, in Table 1, *facility* is the negative spot in the schema prompt, i.e., there is no *facility* entity in the sentence “Steve became CEO of Apple in 1997”. Therefore, we randomly inject the noise of “(facility: [NULL])” into the target record during model learning. In this way, the UIE can effectively learn to reject misleading generation by generating [NULL] token.

4 Experiments

To verify the effectiveness of UIE, we conducted experiments on different IE tasks and settings.

4.1 Experimental Settings

Datasets. We conduct experiments on 13 IE benchmarks across 4 well-representative IE tasks (including entity extraction, relation extraction, event extraction, structured sentiment extraction) and their combinations (e.g., joint entity-relation extraction). The used datasets includes ACE04 (Mitchell et al., 2005), ACE05 (Walker et al., 2006); CoNLL03 (Tjong Kim Sang and De Meulder, 2003), CoNLL04 (Roth and Yih, 2004), SciERC (Luan et al., 2018), NYT (Riedel et al., 2010), CASIE (Satyapanich et al., 2020), SemEval-14 (Pontiki et al., 2014), SemEval-15 (Pontiki et al., 2015), SemEval-16 (Pontiki et al., 2016), see Table 8 for detail. We employ the end-to-end setting for all extraction tasks, which takes the raw text as input and directly generates the target structure.

Evaluation. We use the same evaluation metrics as all previous methods, and details of metrics are shown in the appendix. For each fine-tuning experiment, we report the average performance on 3 random seeds. Because UIE only generates text spans, we map spans to offsets by finding the first matched offsets that are not already matched in the same SEL hierarchical level (details in appendix). We found this simple heuristic rule is very effective (<0.5% error offsets) and more complicated mapping approaches (such as attention-weight guided span mapping) are left as the future work.

Dataset	Domain	Metric	Comparable SOTA	SEL	UIE
ACE04	News, Speech	Entity F1	(Yan et al., 2021b)	86.84	86.52
ACE05-Ent	News, Speech	Entity F1	(Yan et al., 2021b)	84.74	85.78
CoNLL03	News	Entity F1	(Wang et al., 2021a)	93.21	92.17
ACE05-Rel	News, Speech	Relation Strict F1	(Zhong and Chen, 2021)	65.60	64.68
CoNLL04	News	Relation Strict F1	(Wang and Lu, 2020)	73.60	73.07
NYT	News	Relation Triplet F1	(Zheng et al., 2021)	92.70	93.54
SciERC	Scientific	Relation Strict F1	(Zhong and Chen, 2021)	35.60	33.36
ACE05-Evt	News, Speech	Event Trigger F1	(Lin et al., 2020)	72.80	72.63
		Event Argument F1	(Lin et al., 2020)	54.80	54.67
CASIE	Cybersecurity	Event Trigger F1	(Lu et al., 2021)	67.51	68.98
		Event Argument F1	(Lu et al., 2021)	59.45	60.37
14-res	Reviews	Sentiment Triplet F1	(Zhang et al., 2021)	72.16	73.78
14-lap	Reviews	Sentiment Triplet F1	(Zhang et al., 2021)	60.78	63.15
15-res	Reviews	Sentiment Triplet F1	(Xu et al., 2021)	63.27	66.10
16-res	Reviews	Sentiment Triplet F1	(Xu et al., 2021)	70.26	73.87

Table 2: Overall results of UIE-large on different datasets. SEL refers to UIE without pre-training by directly using T5-v1.1-large as the backbone. Because NYT overlaps with pre-training data, we didn’t conduct UIE on NYT for fair comparison. More results of UIE-base and the details of evaluation metric are shown in the appendix.

4.2 Experiments on Supervised Settings

UIE provides a universal backbone for IE tasks. This section assesses the UIE performance in supervised settings. We compare UIE with the state-of-the-art, task-specific supervised models. For a fair comparison, we only compare the state-of-the-art without leveraging additional dataset-specific knowledge or larger-scale contexts. These extensions are good complementary of UIE, and can be left for further improvement. Table 2 shows the performance of UIE on the 13 IE datasets across 4 tasks. We can observe that:

1) *By modeling IE as text-to-structure generation and encoding with an effective SEL language, UIE provides an effective universal architecture for IE.* The UIE model achieves state-of-the-art performance on nearly all datasets and tasks, even without pre-training (SEL). 2) *The large-scale pre-trained model provides a solid foundation for universal IE.* Compared with baselines, the pre-trained model achieves the performance of the state-of-the-art in most datasets and improves 1.42% F1 on average. 3) *By universally modeling IE tasks and pre-training using large-scale datasets, UIE can effectively capture, share, and transfer IE abilities.* Pre-training improves all tasks at the same time, especially events and sentiment knowledge rarely appear in the pre-train dataset. It proves that SEL is a unified and cross-task transferable structured

representation for IE, which allows UIE to share learned capabilities and information across different and various information extraction tasks.

4.3 Experiments on Low-resource Settings

To verify the quick adaptation ability of UIE, we conducted low-resource experiments on six different partitions of the original training sets (1/5/10-shot, 1/5/10% ratio) across 4 tasks. For the few-shot experiments, we sample 1/5/10 sentences for each entity/relation/event/sentiment type in the training set. To avoid the influence of random sampling, we repeated each experiment 10 times with different samples and reported their averaged results as previous works (Huang et al., 2021).

We compare UIE with the following pre-trained model: 1) **T5-v1.1-base** is an initial model of UIE-base; 2) **Fine-tuned T5-base** is fine-tuned with sequence generation tasks such as summarization, which have been shown effective in many low-resource NLP tasks (Paolini et al., 2021); 3) **UIE-base w/o SSI** is the distant supervised version of UIE without SSI in the pre-training stage, which is used to verify the necessity of SSI when adapting UIE in low-resource settings. Table 3 shows the performance of 4 IE tasks under 6 low-resource settings. We observe that: 1) *By guiding the generation using schema-based prompts, SSI is an effective way for adaptively controlling which to ex-*

	Model	1-Shot	5-Shot	10-Shot	AVE-S	1%	5%	10%	AVE-R
Entity (CoNLL03) Ent-F1	T5-v1.1-base	12.73	30.17	58.89	33.93	75.74	85.71	87.70	83.05
	Fine-tuned T5-base	24.93	54.85	65.31	48.36	78.51	87.67	88.91	85.03
	UIE-base w/o SSI	43.52	64.76	72.47	60.25	81.91	88.41	89.84	86.72
	UIE-base	46.43	67.09	73.90	62.47	82.84	88.34	89.63	86.94
Relation (CoNLL04) Rel-S F1	T5-v1.1-base	2.35	7.99	25.98	12.11	6.08	32.38	41.87	26.78
	Fine-tuned T5-base	4.24	28.16	41.44	24.61	12.89	37.75	49.95	33.53
	UIE-base w/o SSI	13.21	40.35	49.47	34.34	24.21	48.70	56.59	43.17
	UIE-base	22.05	45.41	52.39	39.95	30.77	51.72	59.18	47.22
Event Trigger (ACE05-Evt) Evt Tri F1	T5-v1.1-base	19.40	43.35	50.57	37.77	25.59	49.47	57.18	44.08
	Fine-tuned T5-base	30.18	48.31	51.27	43.25	31.08	51.16	57.76	46.67
	UIE-base w/o SSI	32.07	48.11	51.00	43.73	32.71	53.20	59.26	48.39
	UIE-base	38.14	51.21	53.23	47.53	41.53	55.70	60.29	52.51
Event Argument (ACE05-Evt) Evt Arg F1	T5-v1.1-base	2.75	20.21	27.53	16.83	3.59	21.53	30.90	18.67
	Fine-tuned T5-base	6.96	25.07	30.96	21.00	7.39	24.97	33.90	22.09
	UIE-base w/o SSI	9.31	23.99	30.31	21.20	9.57	27.25	34.18	23.67
	UIE-base	11.88	27.44	33.64	24.32	12.80	30.43	36.28	26.50
Sentiment (16res) Rel-S F1	T5-v1.1-base	0.04	2.11	12.66	4.94	3.50	27.08	45.97	25.52
	Fine-tuned T5-base	6.55	21.06	29.92	19.18	18.72	39.63	51.65	36.67
	UIE-base w/o SSI	7.79	17.77	32.07	19.21	19.14	42.76	53.44	38.45
	UIE-base	10.50	26.24	39.11	25.28	24.24	49.31	57.61	43.72

Table 3: Low-resource results on end-to-end IE tasks, where **AVE-S**(hot) and **AVE-R**(atio) are the averaged performance across 3 few-shot settings and 3 low-resource settings respectively.

tract. Compared with the UIE model w/o SSI, UIE equipped with SSI achieves improvements of 4.16 and 3.30 on average for n-shot and n-ratio experiments. 2) *Our pre-training algorithms can learn general IE ability rather than capture task-specific information.* Even the pre-training of UIE didn't include event and sentiment knowledge, UIE still achieved significantly better performance on these tasks compared to the baseline with only a small number of samples.

4.4 Ablations on Pre-training Tasks

Task	Entity	Relation	Event		Sent.
F1	Ent	Rel-S	Evt-Tri	Evt-Arg	Rel-S
UIE-base	95.89	75.97	72.63	57.27	74.73
w/o $\mathcal{L}_{\text{Pair}}$	95.83	75.07	71.20	55.79	74.27
w/o $\mathcal{L}_{\text{Record}}$	95.69	75.68	71.99	57.60	74.43
w/o $\mathcal{L}_{\text{Text}}$	95.66	75.70	70.89	54.16	74.28
T5-v1.1-base	95.29	72.12	70.50	54.42	72.03

Table 4: Experiment results of UIE-base with different learning tasks on the development set of four downstream datasets: entity (CoNLL03), relation (CoNLL04), event (ACE05-Evt) and sentiment (16res).

To investigate the effect of different pre-training tasks, Table 4 shows ablation experiment results of UIE-base on four downstream tasks. We can

	ΔP	P	R	F
UIE-base		79.54	72.63	75.91
w/o rejection	+11.41	68.13	67.85	66.13
UIE-base w/o SSI		78.96	70.50	74.49
w/o rejection	+9.41	69.55	63.69	66.44
T5-base		74.12	61.72	67.33
w/o rejection	+17.95	56.17	56.00	55.94
T5-v1.1		71.88	51.23	59.67
w/o rejection	+13.88	58.00	45.04	50.38

Table 5: Experiment results of 10-shot setting on the CoNLL 03 development set.

see that: (1) *The pre-training of SEL ($\mathcal{L}_{\text{Record}}$) and sequence-to-structure mapping ($\mathcal{L}_{\text{Pair}}$) is crucial for UIE, and such a structure generation pre-training is especially useful for small-scale datasets.* In small datasets CoNLL04 and 16res, adding structure generation pre-training (from T5-v1.1-base to UIE-base w/o $\mathcal{L}_{\text{Text}}$), the performance significantly increases from 72.12 to 75.70 and 72.03 to 74.28. (2) *Retrofitting semantic using the mask language model task ($\mathcal{L}_{\text{Text}}$) is more important for the complex extraction task.* In the tasks with more semantic types such as event extraction (33 types), the performance drops significantly after removing the $\mathcal{L}_{\text{Text}}$ task, e.g., 72.63 \rightarrow 70.89 and

57.27→54.16. (3) *The mapping pre-training with \mathcal{L}_{Pair} enables the model to learn the ability of extraction.* After ablating \mathcal{L}_{Pair} , the extraction ability of UIE is significantly decreased, i.e., the performance on the relation (-0.90), event (-1.43/-1.48), and sentiment (-0.46) tasks all see large decline.

4.5 Effects of Rejection Noise

This section investigates the effect of the proposed rejection noise. Table 5 shows the results of the different pre-trained models on the development set of CoNLL 03 under the 10-shot setting. The mis-generated label has a negative influence on the precision of the proposed generation method leading to a large number of error extraction results. The proposed rejection noise is useful for the generation method, which leads to improvements of 13.16 precision (P) on average.

5 Related Work

Building and pre-training universal models of NLP tasks has attracted a lot of attention in recent years, e.g., contextualized representation (Devlin et al., 2019; Liu et al., 2019), text generation (Lewis et al., 2020; Raffel et al., 2020), multi-modal (Li et al., 2021b; Cho et al., 2021), and multi-lingual (Conneau et al., 2020; Xue et al., 2021). This paper proposes and pre-trains the first universal model for information extraction.

IE is a long-researched area and many classical neural architectures have been proposed, such as sequence tagging (Lample et al., 2016; Zheng et al., 2017; Lin et al., 2019), span classification (Sohrab and Miwa, 2018; Lin et al., 2018; Wadden et al., 2019), and MRC (Levy et al., 2017; Li et al., 2020; Du and Cardie, 2020). And several task-specific pre-training techniques are proposed on these architectures (Mengge et al., 2020; Wang et al., 2021b; Qin et al., 2021). More relevant to our work are generation-based IE methods, which generate text spans via tagging (Straková et al., 2019; Ma et al., 2019), index pointer (Ren et al., 2021; Yan et al., 2021b) or copy mechanism (Zeng et al., 2018), and these methods usually employ specific classifiers to represent labels. The generation can be enhanced using label templates (Li et al., 2021a; Liu et al., 2021; Cui et al., 2021), schema (Lu et al., 2021; Ahmad et al., 2021), and augmented language methods (Paolini et al., 2021).

Compared with previous IE studies which focus on developing more effective task-specialized mod-

els, this paper aims to universally model various IE tasks in a unified text-to-structure framework, which can greatly benefit the rapid development, effective knowledge sharing, and quick adaptation of IE systems.

6 Conclusion

In this paper, we propose a unified text-to-structure generation framework – UIE, which can universally model different IE tasks, adaptively generate targeted structures, and unifiedly learn general IE abilities from different knowledge sources. Experimental results show that UIE achieves very competitive performance in both supervised and low-resource settings, which verified its universality, effectiveness, and transferability. A large-scale pre-trained text-to-structure model is also released, which will benefit future studies. For future work, we want to extend UIE to KB-aware IE tasks such as entity linking (Cao et al., 2021), and document-aware IE tasks such as co-reference (Lee et al., 2017; Lu et al., 2022).

Acknowledgements

We sincerely thank the reviewers for their insightful comments and valuable suggestions. This research work is supported by the National Natural Science Foundation of China under Grants no. U1936207, 62122077 and 62106251, the Project of the Chinese Language Committee under Grant no. YB2003C002.

References

- Wasi Ahmad, Jianfeng Chi, Tu Le, Thomas Norton, Yuan Tian, and Kai-Wei Chang. 2021. [Intent classification and slot filling for privacy policies](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4402–4417, Online. Association for Computational Linguistics.
- Peggy M. Andersen, Philip J. Hayes, Steven P. Weinstein, Alison K. Huettner, Linda M. Schmandt, and Irene B. Nirenburg. 1992. [Automatic extraction of facts from press releases to generate news stories](#). In *Third Conference on Applied Natural Language Processing*, pages 170–177, Trento, Italy. Association for Computational Linguistics.
- Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. [Autoregressive entity retrieval](#). In *International Conference on Learning Representations*.

- Xiangrong Chen and Alan L. Yuille. 2004. [Detecting and reading text in natural scenes](#). In *2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004), with CD-ROM, 27 June - 2 July 2004, Washington, DC, USA*, pages 366–373. IEEE Computer Society.
- Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. 2021. [Unifying vision-and-language tasks via text generation](#). In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 1931–1942. PMLR.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.
- Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. [Template-based named entity recognition using BART](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1835–1845, Online. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Xinya Du and Claire Cardie. 2020. [Event extraction by answering \(almost\) natural questions](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683, Online. Association for Computational Linguistics.
- Ralph Grishman. 2019. [Twenty-five years of information extraction](#). *Natural Language Engineering*, 25(6):677–692.
- Ralph Grishman and Beth Sundheim. 1996. [Message Understanding Conference- 6: A brief history](#). In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.
- Jiaxin Huang, Chunyuan Li, Krishan Subudhi, Damien Jose, Shobana Balakrishnan, Weizhu Chen, Baolin Peng, Jianfeng Gao, and Jiawei Han. 2021. [Few-shot named entity recognition: An empirical baseline study](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10408–10423, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Heng Ji and Ralph Grishman. 2011. [Knowledge base population: Successful approaches and challenges](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1148–1158, Portland, Oregon, USA. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *The Third International Conference on Learning Representations*, San Diego.
- Saul Kripke and Milton K Munitz. 1971. [Identity and necessity](#). 1971, pages 135–164.
- Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. [Neural architectures for named entity recognition](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.
- Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. [End-to-end neural coreference resolution](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.
- Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. [Zero-shot relation extraction via reading comprehension](#). In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada. Association for Computational Linguistics.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Peng Li, Jing Jiang, and Yinglin Wang. 2010. [Generating templates of entity summaries with an entity-aspect model and pattern mining](#). In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 640–649, Uppsala, Sweden. Association for Computational Linguistics.
- Sha Li, Heng Ji, and Jiawei Han. 2021a. [Document-level event argument extraction by conditional generation](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 894–908, Online. Association for Computational Linguistics.

- Wei Li, Can Gao, Guocheng Niu, Xinyan Xiao, Hao Liu, Jiachen Liu, Hua Wu, and Haifeng Wang. 2021b. [UNIMO: Towards unified-modal understanding and generation via cross-modal contrastive learning](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2592–2607, Online. Association for Computational Linguistics.
- Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. [A unified MRC framework for named entity recognition](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5849–5859, Online. Association for Computational Linguistics.
- Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. 2018. [Nugget proposal networks for Chinese event detection](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1565–1574, Melbourne, Australia. Association for Computational Linguistics.
- Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. 2019. [Sequence-to-nuggets: Nested entity mention detection via anchor-region networks](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5182–5192, Florence, Italy. Association for Computational Linguistics.
- Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. [A joint neural model for information extraction with global features](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. Association for Computational Linguistics.
- Qing Liu, Hongyu Lin, Xinyan Xiao, Xianpei Han, Le Sun, and Hua Wu. 2021. [Fine-grained entity typing via label reasoning](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4611–4622, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized BERT pretraining approach](#). *CoRR*, abs/1907.11692.
- Yaojie Lu, Hongyu Lin, Jialong Tang, Xianpei Han, and Le Sun. 2022. [End-to-end neural event coreference resolution](#). *Artificial Intelligence*, 303:103632.
- Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. [Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2795–2806, Online. Association for Computational Linguistics.
- Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. [Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.
- Dehong Ma, Sujian Li, Fangzhao Wu, Xing Xie, and Houfeng Wang. 2019. [Exploring sequence-to-sequence learning in aspect term extraction](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3538–3547, Florence, Italy. Association for Computational Linguistics.
- Xue Mengge, Bowen Yu, Zhenyu Zhang, Tingwen Liu, Yue Zhang, and Bin Wang. 2020. [Coarse-to-Fine Pre-training for Named Entity Recognition](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6345–6354, Online. Association for Computational Linguistics.
- David Milward and James Thomas. 2000. [From information retrieval to information extraction](#). In *ACL-2000 Workshop on Recent Advances in Natural Language Processing and Information Retrieval*, pages 85–97, Hong Kong, China. Association for Computational Linguistics.
- Alexis Mitchell, Stephanie Strassel, Shudong Huang, and Ramez Zakhary. 2005. [Ace 2004 multilingual training corpus](#).
- Boyan Onyshkevych. 1994. [Issues and methodology for template design for information extraction](#). In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.
- Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, RISHITA ANUBHAI, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. [Structured prediction as translation between augmented natural languages](#). In *International Conference on Learning Representations*.
- Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. [SemEval-2016 task 5: Aspect based sentiment analysis](#). In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, San Diego, California. Association for Computational Linguistics.

- Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. [SemEval-2015 task 12: Aspect based sentiment analysis](#). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 486–495, Denver, Colorado. Association for Computational Linguistics.
- Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. [SemEval-2014 task 4: Aspect based sentiment analysis](#). In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.
- Yujia Qin, Yankai Lin, Ryuichi Takanobu, Zhiyuan Liu, Peng Li, Heng Ji, Minlie Huang, Maosong Sun, and Jie Zhou. 2021. [ERICA: Improving entity and relation understanding for pre-trained language models via contrastive learning](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3350–3363, Online. Association for Computational Linguistics.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Marc’Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. [Sequence level training with recurrent neural networks](#). In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.
- Liliang Ren, Chenkai Sun, Heng Ji, and Julia Hockenmaier. 2021. [HySPA: Hybrid span generation for scalable text-to-graph extraction](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4066–4078, Online. Association for Computational Linguistics.
- Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*, pages 148–163, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Dan Roth and Wen-tau Yih. 2004. [A linear programming formulation for global inference in natural language tasks](#). In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 1–8, Boston, Massachusetts, USA. Association for Computational Linguistics.
- Taneeya Satyapanich, Francis Ferraro, and Tim Finin. 2020. [Casie: Extracting cybersecurity event information from text](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8749–8757.
- Mohammad Golam Sohrab and Makoto Miwa. 2018. [Deep exhaustive model for nested named entity recognition](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2843–2849, Brussels, Belgium. Association for Computational Linguistics.
- Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. [Conceptnet 5.5: An open multilingual graph of general knowledge](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, pages 4444–4451.
- Jana Straková, Milan Straka, and Jan Hajic. 2019. [Neural architectures for nested NER through linearization](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5326–5331, Florence, Italy. Association for Computational Linguistics.
- Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, Xianrong Zeng, and Shengping Liu. 2020. [Joint entity and relation extraction with set prediction networks](#). *CoRR*, abs/2011.01675.
- Bruno Taillé, Vincent Guigue, Geoffrey Scoutheeten, and Patrick Gallinari. 2020. [Let’s Stop Incorrect Comparisons in End-to-end Relation Extraction!](#) In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3689–3701, Online. Association for Computational Linguistics.
- Erik F. Tjong Kim Sang and Fien De Meulder. 2003. [Introduction to the conll-2003 shared task: Language-independent named entity recognition](#). In *Proceedings of CoNLL-2003*, pages 142–147. Edmonton, Canada.
- David Wadden, Ulme Wennberg, Yi Luan, and Hananeh Hajishirzi. 2019. [Entity, relation, and event extraction with contextualized span representations](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789, Hong Kong, China. Association for Computational Linguistics.
- Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. [Ace 2005 multilingual training corpus](#).
- Jue Wang and Wei Lu. 2020. [Two are better than one: Joint entity and relation extraction with table-sequence encoders](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1706–1721, Online. Association for Computational Linguistics.
- Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu.

- 2021a. [Improving named entity recognition by external context retrieving and cooperative learning](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1800–1812, Online. Association for Computational Linguistics.
- Yucheng Wang, Bowen Yu, Yueyang Zhang, Tingwen Liu, Hongsong Zhu, and Limin Sun. 2020. [TPLinker: Single-stage joint extraction of entities and relations through token pair linking](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1572–1582, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Ziqi Wang, Xiaozhi Wang, Xu Han, Yankai Lin, Lei Hou, Zhiyuan Liu, Peng Li, Juanzi Li, and Jie Zhou. 2021b. [CLEVE: Contrastive Pre-training for Event Extraction](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6283–6297, Online. Association for Computational Linguistics.
- Lu Xu, Yew Ken Chia, and Lidong Bing. 2021. [Learning span-level interactions for aspect sentiment triplet extraction](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4755–4766, Online. Association for Computational Linguistics.
- Lu Xu, Hao Li, Wei Lu, and Lidong Bing. 2020. [Position-aware tagging for aspect sentiment triplet extraction](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2339–2349, Online. Association for Computational Linguistics.
- Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. [mT5: A massively multilingual pre-trained text-to-text transformer](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.
- Hang Yan, Junqi Dai, Tuo Ji, Xipeng Qiu, and Zheng Zhang. 2021a. [A unified generative framework for aspect-based sentiment analysis](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2416–2429, Online. Association for Computational Linguistics.
- Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021b. [A unified generative framework for various NER subtasks](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5808–5822, Online. Association for Computational Linguistics.
- 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 *Proc. of ECAI*.
- Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2018. [Extracting relational facts by an end-to-end neural model with copy mechanism](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 506–514, Melbourne, Australia. Association for Computational Linguistics.
- Ranran Haoran Zhang, Qianying Liu, Aysa Xuemo Fan, Heng Ji, Daojian Zeng, Fei Cheng, Daisuke Kawahara, and Sadao Kurohashi. 2020. [Minimize exposure bias of Seq2Seq models in joint entity and relation extraction](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 236–246, Online. Association for Computational Linguistics.
- Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2021. [Towards generative aspect-based sentiment analysis](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 504–510, Online. Association for Computational Linguistics.
- Hengyi Zheng, Rui Wen, Xi Chen, Yifan Yang, Yunyan Zhang, Ziheng Zhang, Ningyu Zhang, Bin Qin, Xu Ming, and Yefeng Zheng. 2021. [PRGC: Potential relation and global correspondence based joint relational triple extraction](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6225–6235, Online. Association for Computational Linguistics.
- Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 2017. [Joint extraction of entities and relations based on a novel tagging scheme](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1227–1236, Vancouver, Canada. Association for Computational Linguistics.
- Zexuan Zhong and Danqi Chen. 2021. [A frustratingly easy approach for entity and relation extraction](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 50–61, Online. Association for Computational Linguistics.

A Experiment Details

This section describes the details of experiments, including pre-training and fine-tuning on downstream tasks.

A.1 Pre-training Details

Data Construction We use the 20210401 version of Wikipedia² and Wikidata³ dump and ConceptNet⁴ to construct the pre-train dataset.

For Wikidata and Wikipedia, we use them to collect the tuples $\mathcal{T}_w = \{ \langle T_h, e_h, r, e_t, X \rangle \}$, where T_h is head entity type, e_h is head entity, r is relation, e_t is tail entity, X is sentence, and the \mathcal{T}_w can be used to construct $\mathcal{D}_{\text{pair}}$, $\mathcal{D}_{\text{record}}$ and $\mathcal{D}_{\text{text}}$. Firstly, we construct entity type dictionary \mathcal{L} and relation dictionary \mathcal{P} from Wikidata. Wikidata has more than 40M entity items and each item has its corresponding properties which indicate the association between entities. For type dictionary \mathcal{L} , we regard each item as an entity, use the “instance of” and “subclass of” property values as its corresponding entity types and consider other properties as the relation of the entity with others. To learn general knowledge, all entity types will be retained except those whose instances are < 5 . For the type whose name is longer than 3 tokens, we use its headwords as the final type for simplicity, e.g., “state award of the Republic of Moldova” is converted to “state award”. For relation dictionary \mathcal{P} , Wikidata has more than 9K kinds of properties⁵, we filter out the properties of external-id, URL, and math types. In this way, we obtain a collection of 31K types and retained 1535 properties which can serve as a solid foundation for universal IE. Secondly, we collect the mentions of each entity by using its anchor texts in Wikipedia and the top 3 frequent noun phrase occurrences of its entry page (Li et al., 2010). Then for each mention, we identify its entity types by linking it to its Wikidata item’s types. For each Wikipedia page, we split the text into sentences⁶ and filter out sentences that have no entities. Thirdly, we regard each entity as a head entity and find the associated entities according to its properties. The associated entity will set as as tail entity, and the property value will set as association type. If a head entity has no type, T_h will be blank or

²<https://www.wikipedia.org/>

³<https://www.wikidata.org/>

⁴<https://conceptnet.io/>

⁵https://www.wikidata.org/wiki/Wikidata:List_of_properties

⁶nltk.tokenize.punkt

Algorithm 1 The pre-training process of UIE in a Python-like style.

```

# The training details of UIE
function pretraining_process
  for step in all_steps do
    batch = []
    # load  $n_{\text{text}}$  unstructured text samples
    texts = get_data( $\mathcal{D}_{\text{text}}$ ,  $n_{\text{text}}$ )
    # construct corrupted source text  $x'$  and
    # corrupted spans  $x''$  for each text sample
    for x in texts do
       $x', x'' = \text{span\_corrupt}(x)$ 
      batch.extend((None,  $x'$ ,  $x''$ ))
    end for
    # load  $n_{\text{record}}$  structured record samples
    records = get_data( $\mathcal{D}_{\text{record}}$ ,  $n_{\text{record}}$ )
    for y in records do
      batch.extend((None, None, y))
    end for
    # load  $n_{\text{pair}}$  text-record parallel pairs
    text_record_pairs = get_data( $\mathcal{D}_{\text{pair}}$ ,  $n_{\text{pair}}$ )
    # construct meta-schema  $s_{\text{meta}}$ 
    # for each text-record pair ( $x$ ,  $y$ )
    for ( $x, y$ ) in text_record_pairs do
       $s = \text{meta\_schema\_sample}(y)$ 
      batch.extend(( $s, x, y$ ))
    end for
    # compute loss and backward
     $\mathcal{L}_{\text{pair}}, \mathcal{L}_{\text{record}}, \mathcal{L}_{\text{text}} = \text{UIE}(batch)$ 
    loss =  $\mathcal{L}_{\text{pair}} + \mathcal{L}_{\text{record}} + \mathcal{L}_{\text{text}}$ 
    loss.backward()
  end for
end function

# The meta sample of UIE
function meta_schema_sample(y)
  # get positive spots and associations
  # in the record y
   $s_{s+}, s_{a+} = \text{get\_schema\_from\_record}(y)$ 
  # sample negative spots
   $s_{s-} = \text{sample\_negative\_spot}(s_{s+})$ 
  # sample negative associations
   $s_{a-} = \text{sample\_negative\_association}(s_{a+})$ 
  return  $s_{s+} \cup s_{s-} \cup s_{a+} \cup s_{a-}$ 
end function

```

has no associated tail entity, r and e_t will be blank. To this end, given a sentence, we can construct instances based on the collected tuples \mathcal{T}_w by setting e_h and e_t as INFOSPAN, and assigning T_h as SPOT-NAME, r as ASSONAME. Finally, from Wikipedia and Wikidata, we construct $\mathcal{D}_{\text{pair}}$, $\mathcal{D}_{\text{record}}$ and $\mathcal{D}_{\text{text}}$ with 65M instances, respectively. And we keep 50K as the development dataset.

To add common sense knowledge to structured extraction language (SEL), we extract the tuples \mathcal{T}_c from ConceptNet. ConceptNet contains 48 associations and has no context or entity types. So we leave the T_h, T_t, X blank and finally construct 1M instances.

Hyper-parameter	UIE-base				UIE-large		
	Pre-training	Fine-tuning			Pre-training	Fine-tuning	
		Ent/Rel/Evt	Sentiment	Low-resource		Ent/Rel/Evt	Sentiment
Learning Rate	1e-4	1e-4, 3e-4, 5e-4		1e-4	1e-4	5e-5, 1e-4, 3e-4	
Rejection Noise p_e	0.0	0, 0.1, 0.2		0.1	0.0	0, 0.1, 0.2	
Global Batch Size	512	64	16	16	512	32	8
Schedule	linear	linear	linear	constant	linear	linear	linear
Warmup Rate	0.06	0.06	0.06	0.0	0.06	0.06	0.06
Epoch/Step	500K step	50 epoch	50 epoch	200 epoch	500K step	50 epoch	50 epoch

Table 6: Hyper-parameters pre-training and fine-tuning for UIE-base and UIE-large.

Hyper-parameter	UIE-base	UIE-large
# Layers of Encoder	12	24
# Layers of Decoder	12	24
Hidden Dimension	768	1,024
FF hidden size	2,048	2,816
Layer Normalize ϵ	1e-6	1e-6
# Attention head	12	16
Attention head size	64	64

Table 7: Model architectures.

	lEntl	lRel	lEvtl	#Train	#Val	#Test
ACE04	7	-	-	6,202	745	812
ACE05-Ent	7	-	-	7,299	971	1,060
CoNLL03	4	-	-	14,041	3,250	3,453
ACE05-Rel	7	6	-	10,051	2,420	2,050
CoNLL04	4	5	-	922	231	288
NYT	3	24	-	56,196	5,000	5,000
SciERC	6	7	-	1,861	275	551
ACE05-Evt	-	-	33	19,216	901	676
CASIE	21	-	5	11,189	1,778	3,208
14res	2	3	-	1,266	310	492
14lap	2	3	-	906	219	328
15res	2	3	-	605	148	322
16res	2	3	-	857	210	326

Table 8: Detailed datasets statistics. |*| indicates the number of categories, and # is the number of sentences in the specific subset. We take sentiment types as special relation type: positive, negative, and neutral; and each sentiment triplet holds a aspect and a opinion.

Training Details We first initialize UIE-base and UIE-large with T5-v1.1-base and T5-v1.1-large checkpoints (Raffel et al., 2020), and the model architectures are shown in Table 7. We employ Adam optimizer (Kingma and Ba, 2015) as the optimizer with learning rate=1e-4, and use linear scheduling with a warming up proportion 6%. For negative spots and associations in the $\mathcal{L}_{\text{Pair}}$, we randomly select negative spots and associations up to 10 for each instance, respectively. For $\mathcal{L}_{\text{Text}}$, we set the corruption rate to 15% and the average

corrupting span length to 3, following Raffel et al. (2020). We truncate the concatenated overall length of schema prompt s and raw text x , as well as the length of SEL expression y , together to 128 during pre-training. We train our base model and large model for both 500K steps with batch size 512 on 8 NVIDIA A100 GPUs.

The detailed pre-training process in a python-like style is shown in Algorithm 1. In each batch of pre-training processes for UIE, we construct a batch of triplets (s, x, y) containing text-record pairs, text instances, and record instances. In practice, since 8 GPUs could only run the large model with an overall batch of 128 (batch=16 on each GPU), we update the model parameters after accumulating 4 gradients.

A.2 Details of Downstream Tasks

We conduct downstream tasks on 4 IE tasks, 13 datasets, and the detailed statistic of each dataset is shown in Table 8.

Entity We conduct entity extraction experiments on three entity datasets: ACE04⁷ (Mitchell et al., 2005), ACE05-Ent⁸ (Walker et al., 2006), and CoNLL03 (Tjong Kim Sang and De Meulder, 2003). For nested entity extraction datasets ACE04 and ACE05-Ent, we follow the pre-processing steps and data split of previous works (Li et al., 2020).

Relation We conduct experiments on four wide-used end-to-end relation extraction datasets across several languages and domains: ACE05-Rel (Walker et al., 2006), CoNLL04⁹ (Roth and Yih, 2004), NYT¹⁰ (Riedel et al., 2010), and SciERC¹¹ (Luan et al., 2018). We follow the preprocessing

⁷<https://catalog.ldc.upenn.edu/LDC2005T09>

⁸<https://catalog.ldc.upenn.edu/LDC2006T06>

⁹<https://github.com/btaille/sincere>

¹⁰<https://github.com/yubowen-ph/JointER>

¹¹<http://nlp.cs.washington.edu/sciIE/>

steps and data split of previous works (Taillé et al., 2020; Yu et al., 2020; Wadden et al., 2019).

Event For ACE05-Evt, we follow the same types, data splits, and pre-processing steps as Lin et al. (2020). For CASIE (Satyapanich et al., 2020), we first remove three incomplete annotated documents (999, 10001, 10002), then split the remaining documents into three sets: train/val/test=697/100/200 according to the time order of each document. We employ the state-of-the-art generation-based event extraction method TEXT2EVENT (Lu et al., 2021) as the comparable state-of-the-art system.

Sentiment We conduct sentiment extraction experiments on the sentiment triplet extraction (Xu et al., 2020) of SemEval 14/15/16 aspect sentiment analysis datasets. We employ the pre-processing datasets of the previous work (Yan et al., 2021a)¹².

Evaluation We use span-based offset Micro-F1 as the primary metric to evaluate the model:

- **Entity**: an entity mention is correct if its offsets and type match a reference entity.
- **Relation Strict**: relation with strict match, a relation is correct if its relation type is correct and the offsets and entity types of the related entity mentions are correct.
- **Relation Triplet**: relation with boundary match, a relation is correct if its relation type is correct and the string of the subject/object are correct.
- **Event Trigger**: an event trigger is correct if its offsets and event type matches a reference trigger.
- **Event Argument**: an event argument is correct if its offsets, role type, and event type match a reference argument mention.
- **Sentiment Triplet**: a correct triplet requires the offsets boundary of the target, the offsets boundary of the opinion span, and the target sentiment polarity to be all correct at the same time.

To make a fair comparison with baseline systems, we mapped the generated string-level extraction results to offset-level for model evaluation. In detail, we reconstructed the offset of predicted entity/trigger mentions by finding the matched utterance in the input sequence one by one. For argument mentions in relation and event tasks, we found the nearest matched utterance to the predicted entity/trigger mention as the predicted offset. This simple heuristic offset strategy achieves high accuracy. Compared to the string level evaluation,

Methods	PLM	14res	14lap	15res	16res
(Xu et al., 2020)	BERT-base	62.40	51.04	57.53	63.83
(Yan et al., 2021a)	BART-base	65.25	58.69	59.26	67.62
(Xu et al., 2021)	BERT-base	71.85	59.38	63.27	70.26
(Zhang et al., 2021)	T5-base	72.16	60.78	62.10	70.10
SSI + SEL	UIE-base	72.55	62.94	64.41	72.86
	T5-v1.1-base	71.27	58.69	59.60	70.24

Table 9: Experiment results of UIE-base on the sentiment triplet extraction tasks.

Methods	PLM	P	R	F
(Wang et al., 2020)	BERT-base	91.40	92.60	92.00
(Sui et al., 2020)	BERT-base	92.50	92.20	92.30
(Zheng et al., 2021)	BERT-base	93.50	91.90	92.70
SSI + SEL	T5-v1.1-base	91.94	93.28	92.60

Table 10: Experiment results of SSI and SEL on the NYT (the joint entity and relation extraction setting).

the error rate of the reported offset level evaluation is less than 0.5%. More complicated mapping approaches are left as future work.

Table 6 shows the detailed hyper-parameters for downstream tasks.

A.3 Comparison of UIE-base

This section introduces detailed experiment results of UIE-base.

Table 9 shows the performance of UIE-base and the state-of-the-art systems on the four aspect-based sentiment analysis datasets. As shown in Table 9, the proposed SEL and SSI also have strong portability to sentiment triplets extraction, which achieves the competitive performance with the state-of-the-art with task-specific architectures. With the unified pre-training, UIE-base achieves an improvement of 3.24 on average over T5-v1.1-base across four datasets. This verifies the proposed unified pre-training algorithms can learn general IE ability even the sentiment knowledge is rarely in the pre-training stage.

Table 10 shows the performance of SEL-SSI with the T5-v1.1-base for NYT. Due to the high overlapping of NYT and pre-trained data, we didn’t conduct the experiment of UIE on NYT. Even without pre-training, SSI + SEL still achieved the state-of-the-art performance on NYT. This is because of the flexible generation architecture and the universal SEL expression, UIE can naturally handle entity overlap problems.

¹²<https://github.com/yhcc/BARTABSA>

Task	Dataset	Structural Schema Instructor
Entity	ACE04/05-Ent	<spot> facility <spot> geographical social political <spot> location <spot> organization <spot> person <spot> vehicle <spot> weapon
Entity	CoNLL03	<spot> location <spot> miscellaneous <spot> organization <spot> person
Relation	ACE05-Rel	<spot> facility <spot> geographical social political <spot> location <spot> organization <spot> person <spot> vehicle <spot> weapon <asoc> agent artifact <asoc> general affiliation <asoc> organization affiliation <asoc> part whole <asoc> personal social <asoc> physical
Relation	CoNLL04	<spot> location <spot> organization <spot> other <spot> people <asoc> kill <asoc> live in <asoc> located in <asoc> organization in <asoc> work for
Relation	NYT	<spot> location <spot> organization <spot> person <asoc> administrative divisions <asoc> advisors <asoc> capital <asoc> children <asoc> company <asoc> contains <asoc> country <asoc> ethnicity <asoc> founders <asoc> geographic distribution <asoc> industry <asoc> location <asoc> major shareholder of <asoc> major shareholders <asoc> nationality <asoc> neighborhood of <asoc> people <asoc> place founded <asoc> place lived <asoc> place of birth <asoc> place of death <asoc> profession <asoc> religion <asoc> teams
Relation	SciERC	<spot> generic <spot> material <spot> method <spot> metric <spot> other scientific term <spot> task <asoc> compare <asoc> conjunction <asoc> evaluate for <asoc> feature of <asoc> hyponym of <asoc> part of <asoc> used for
Event	ACE05-Evt	<spot> acquit <spot> appeal <spot> arrest jail <spot> attack <spot> born <spot> charge indict <spot> convict <spot> declare bankruptcy <spot> demonstrate <spot> die <spot> divorce <spot> elect <spot> end organization <spot> end position <spot> execute <spot> extradite <spot> fine <spot> injure <spot> marry <spot> meet <spot> merge organization <spot> nominate <spot> pardon <spot> phone write <spot> release parole <spot> sentence <spot> start organization <spot> start position <spot> sue <spot> transfer money <spot> transfer ownership <spot> transport <spot> trial hearing <asoc> adjudicator <asoc> agent <asoc> artifact <asoc> attacker <asoc> beneficiary <asoc> buyer <asoc> defendant <asoc> destination <asoc> entity <asoc> giver <asoc> instrument <asoc> organization <asoc> origin <asoc> person <asoc> place <asoc> plaintiff <asoc> prosecutor <asoc> recipient <asoc> seller <asoc> target <asoc> vehicle <asoc> victim
Event	CASIE	<spot> capabilities <spot> common vulnerabilities and exposures <spot> data <spot> databreach <spot> device <spot> discover vulnerability <spot> file <spot> geopolitical entity <spot> malware <spot> money <spot> number <spot> organization <spot> patch <spot> patch vulnerability <spot> payment method <spot> person <spot> personally identifiable information <spot> phishing <spot> purpose <spot> ransom <spot> software <spot> system <spot> time <spot> version <spot> vulnerability <spot> website <asoc> attack pattern <asoc> attacker <asoc> capabilities <asoc> common vulnerabilities and exposures <asoc> compromised data <asoc> damage amount <asoc> discoverer <asoc> issues addressed <asoc> number of data <asoc> number of victim <asoc> patch <asoc> patch number <asoc> payment method <asoc> place <asoc> price <asoc> purpose <asoc> releaser <asoc> supported platform <asoc> time <asoc> tool <asoc> trusted entity <asoc> victim <asoc> vulnerability <asoc> vulnerable system <asoc> vulnerable system owner <asoc> vulnerable system version
Sentiment	14/15/16-res	<spot> aspect <spot> opinion <asoc> negative <asoc> neutral <asoc> positive
Sentiment	14-lap	<spot> aspect <spot> opinion <asoc> negative <asoc> neutral <asoc> positive

Table 11: Structured schema instructor for each dataset (we use <spot> and <asoc> rather than [spot] and [asoc] for better visualization).

Task	Dataset	Structured Extraction Language
Entity	ACE04/ACE05-Ent	((geographical social political: Filipino) (person: Filipino President) (person: Filipino President Ramos) (person: the six people awarded Magasaysay award) (person: Magasaysay))
Entity	CoNLL03	((organization: EU) (miscellaneous: German) (miscellaneous: British))
Relation	ACE05-Rel	((geographical social political: European) (geographical social political: troika (part whole: European)) (geographical social political: itself) (geographical social political: Washington))
Relation	CoNLL04	((location: Rome (located in: Lazio)) (location: Lazio) (location: Naples (located in: Campania)) (location: Campania))
Relation	NYT	((person: William F. Weld (place lived: New York)) (location: New York))
Relation	SciERC	((method: HMMs) (other scientific term: weak duration constraints (feature of: HMMs)))
Event	ACE05-Evt	((transport: heading (artifact: family) (destination: new hampshire) (origin: lakeland) (vehicle: plane)))
Event	CASIE	((phishing: email scam (trusted entity: a Netflix notification) (victim: subscribers) (trusted entity: the streaming service)) (file: a Netflix notification) (person: subscribers) (system: the streaming service))
Sentiment	14/15/16-res	((aspect: staff (negative: horrible)) (opinion: horrible))
Sentiment	14lap	((opinion: good) (aspect: battery life (positive: good)))

Table 12: Structured extraction language expressions for each dataset.