A Joint Neural Model for Information Extraction with Global Features

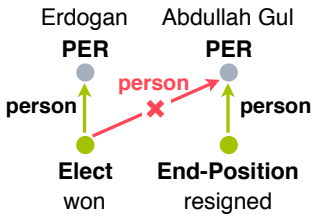
Ying Lin[1], **Heng Ji**[1], **Fei Huang**[2], **Lingfei Wu**[3]
[1]University of Illinois at Urbana-Champaign
[2]Alibaba DAMO Academy [3]IBM Research
{yinglin8,hengji}@illinois.edu,
f.huang@alibaba-inc.com, wuli@us.ibm.com

Abstract

Most existing joint neural models for Information Extraction (IE) use local task-specific classifiers to predict labels for individual instances (e.g., trigger, relation) regardless of their interactions. For example, a VICTIM of a DIE event is likely to be a VICTIM of an ATTACK event in the same sentence. In order to capture such cross-subtask and cross-instance inter-dependencies, we propose a joint neural framework, ONEIE, that aims to extract the globally optimal IE result as a graph from an input sentence. ONEIE performs end-to-end IE in four stages: (1) Encoding a given sentence as contextualized word representations; (2) Identifying entity mentions and event triggers as nodes; (3) Computing label scores for all nodes and their pairwise links using local classifiers; (4) Searching for the globally optimal graph with a beam decoder. At the decoding stage, we incorporate global features to capture the cross-subtask and cross-instance interactions. Experiments show that adding global features improves the performance of our model and achieves new state-of-the-art on all subtasks. As ONEIE does not use any language-specific feature, we prove it can be easily applied to new languages or trained in a multilingual manner. Our code and models for English, Spanish and Chinese are publicly available for research purpose [1].

1 Introduction

Information Extraction (IE) aims to extract structured information from unstructured texts. It is a complex task comprised of a wide range of subtasks, such as named, nominal, and pronominal mention extraction, entity linking, entity coreference resolution, relation extraction, event extraction, and event coreference resolution. Early efforts typically perform IE in a pipelined fashion, which leads to the error propagation problem and disallows interactions among components in the pipeline. As a solution, some researchers propose joint inference and joint modeling methods to improve local prediction (Roth and Yih, 2004; Ji and Grishman, 2005; Ji et al., 2005; Sil and Yates, 2013; Li et al., 2014; Durrett and Klein, 2014; Miwa and Sasaki, 2014; Lu and Roth, 2015; Yang and Mitchell, 2016; Kirschnick et al., 2016). Due to the success of deep learning, neural models have been widely applied to various IE subtasks (Collobert et al., 2011; Chiu and Nichols, 2016; Chen et al., 2015; Lin et al., 2016). Recently, some efforts (Wadden et al., 2019; Luan et al., 2019) revisit global inference approaches by designing neural networks with embedding features to jointly model multiple subtasks. However, these methods use separate local task-specific classifiers in the final layer and do not explicitly model the inter-dependencies among tasks and instances. Figure 1 shows a real example where the local argument role classifier predicts a redundant PERSON edge. The model should be able to avoid such mistakes if it is capable of learning and leveraging the fact that it is unusual for an ELECT event to have two PERSON arguments.



Example: Prime Minister **Abdullah Gul** *resigned* earlier Tuesday to make way for **Erdogan**, who *won* a parliamentary seat in by-elections Sunday.

Figure 1: A typical error made by local classifiers without global constraints.

To address this issue, we propose a joint neu-

[1] http://blender.cs.illinois.edu/software/oneie

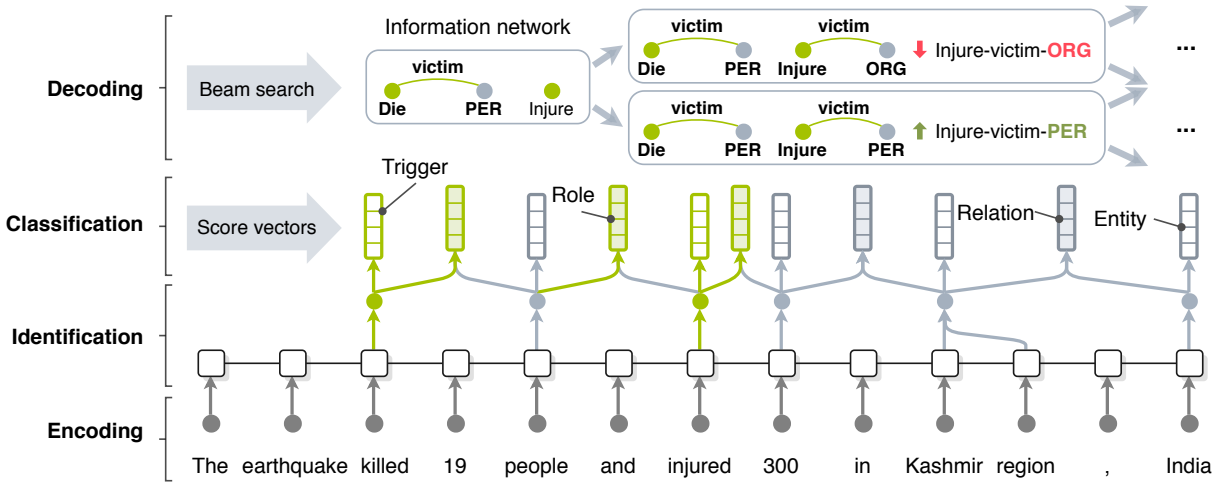


Figure 2: An illustration of our end-to-end joint information extraction framework ONEIE at the test stage. We do not show all pairwise links for simplicity purposes.

ral framework, ONEIE, to perform end-to-end IE with global constraints. As Figure 2 shows, instead of predicting separate knowledge elements using local classifiers, ONEIE aims to extract a globally optimal information network for the input sentence. When comparing candidate information networks during the decoding process, we not only consider individual label scores for each knowledge element, but evaluate cross-subtask and cross-instance interactions in the network. In this example, a graph with the INJURE-VICTIM-ORG (the VICTIM of an INJURE event is an ORG entity) structure is demoted. Experiments show that our framework achieves comparable or better results compared to the state-of-the-art end-to-end architecture (Wadden et al., 2019).

To the best of our knowledge, ONEIE is the first end-to-end neural IE framework that explicitly models cross-subtask and cross-instance interdependencies and predicts the result as a unified graph instead of isolated knowledge elements. Because ONEIE does not rely on language-specific features, it can be rapidly applied to new languages. Furthermore, global features in our framework are highly explainable and can be explicitly analyzed.

2 Task

Given a sentence, our ONEIE framework aims to extract an *information network* representation (Li et al., 2014), where entity mentions and event triggers are represented as nodes, and relations and event-argument links are represented as edges. In other words, we perform entity, relation, and event extraction within a unified framework. In this section, we will elaborate these tasks and involved terminologies.

Entity Extraction aims to identify entity mentions in text and classify them into pre-defined entity types. A mention can be a name, nominal, or pronoun. For example, "Kashmir region" should be recognized as a location (LOC) named entity mention in Figure 2.

Relation Extraction is the task of assigning a relation type to an ordered pair of entity mentions. For example, there is a PART-WHOLE relation between "Kashmir region" and "India".

Event Extraction entails identifying *event triggers* (the words or phrases that most clearly express event occurrences) and their *arguments* (the words or phrases for participants in those events) in unstructured texts and classifying these phrases, respectively, for their types and roles. An argument can be an entity, time expression, or value (e.g., MONEY, JOB-TITLE, CRIME). For example, in Figure 2, the word "injured" triggers an INJURE event and "300" is the VICTIM argument.

We formulate the task of extracting information networks as follows. Given an input sentence, our goal is to predict a graph $G = (V, E)$, where V and E are the node and edge sets respectively. Each node $v_i = \langle a_i, b_i, l_i \rangle \in V$ represents an entity mention or event trigger, where a and b are the start and end word indices, and l is the node type label. Each edge $e_{ij} = \langle i, j, l_{ij} \rangle \in E$ is represented similarly, whereas i and j denote the indices of involved nodes. For example, in Figure 2, the trigger "injured" is represented as $\langle 7, 7, \text{INJURE} \rangle$, the entity mention "Kashmir region" is represented as $\langle 10,$

11, LOC⟩, and the event-argument edge between them is ⟨2, 3, PLACE⟩.

3 Approach

As Figure 2 illustrates, our ONEIE framework extracts the information network from a given sentence in four steps: encoding, identification, classification, and decoding. We encode the input sentence using a pre-trained BERT encoder (Devlin et al., 2019) and identify entity mentions and event triggers in the sentence. After that, we compute the type label scores for all nodes and pairwise edges among them. During decoding, we explore possible information networks for the input sentence using beam search and return the one with the highest global score.

3.1 Encoding

Given an input sentence of L words, we obtain the contextualized representation $\boldsymbol{x}_i$ for each word using a pre-trained BERT encoder. If a word is split into multiple word pieces (e.g., Mondrian → `Mon`, `##dr`, `##ian`), we use the average of all piece vectors as its word representation. While previous methods typically use the output of the last layer of BERT, our preliminary study shows that enriching word representations using the output of the third last layer of BERT can substantially improve the performance on most subtasks.

3.2 Identification

At this stage, we identify entity mentions and event triggers in the sentence, which will act as nodes in the information network. We use a feed-forward network FFN to compute a score vector $\hat{\boldsymbol{y}}_i = \text{FFN}(\boldsymbol{x}_i)$ for each word, where each value in $\hat{\boldsymbol{y}}_i$ represents the score for a tag in a target tag set[2]. After that, we use a conditional random fields (CRFs) layer to capture the dependencies between predicted tags (e.g., an I-PER tag should not follow a B-GPE tag). Similar to (Chiu and Nichols, 2016), we calculate the score of a tag path $\hat{\boldsymbol{z}} = \{\hat{z}_1, ..., \hat{z}_L\}$ as

$$s(\boldsymbol{X}, \hat{\boldsymbol{z}}) = \sum_{i=1}^{L} \hat{y}_{i,\hat{z}_i} + \sum_{i=1}^{L+1} A_{\hat{z}_{i-1},\hat{z}_i},$$

where $\boldsymbol{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_L\}$ is the contextualized representations of the input sequence, $\hat{y}_{i,\hat{z}_i}$ is the $\hat{z}_i$-th component of the score vector $\hat{\boldsymbol{y}}_i$, and $A_{\hat{z}_{i-1},\hat{z}_i}$ is the $(\hat{z}_{i-1}, \hat{z}_i)$ entry in matrix $\boldsymbol{A}$ that indicates the transition score from tag $\hat{z}_{i-1}$ to $\hat{z}_i$. The weights in $\boldsymbol{A}$ are learned during training. We append two special tags `<start>` and `<end>` to the tag path as $\hat{z}_0$ and $\hat{z}_{L+1}$ to denote the start and end of the sequence. At the training stage, we maximize the log-likelihood of the gold-standard tag path as

$$\log p(\boldsymbol{z}|\boldsymbol{X}) = s(\boldsymbol{X}, \boldsymbol{z}) - \log \sum_{\hat{\boldsymbol{z}} \in Z} e^{s(\boldsymbol{X}, \hat{\boldsymbol{z}})},$$

where Z is the set of all possible tag paths for a given sentence. Thus, we define the identification loss as $\mathcal{L}^{\text{I}} = -\log p(\boldsymbol{z}|\boldsymbol{X})$.

In our implementation, we use separate taggers to extract entity mentions and event triggers. Note that we do not use types predicted by the taggers. Instead, we make a joint decision for all knowledge elements at the decoding stage to prevent error propagation and utilize their interactions to improve the prediction of node type.

3.3 Classification

We represent each identified node as $\boldsymbol{v}_i$ by averaging its word representations. After that, we use separate task-specific feed-forward networks to calculate label scores for each node as $\hat{\boldsymbol{y}}_i^t = \text{FFN}^t(\boldsymbol{v}_i)$, where t indicates a specific task. To obtain the label score vector for the edge between the i-th and j-th nodes, we concatenate their span representations and calculate the vector as $\hat{\boldsymbol{y}}_k^t = \text{FFN}^t(\boldsymbol{v}_i, \boldsymbol{v}_j)$.

For each task, the training objective is to minimize the following cross-entropy loss

$$\mathcal{L}^{\text{t}} = -\frac{1}{N^t} \sum_{i=1}^{N^t} \boldsymbol{y}_i^t \log \hat{\boldsymbol{y}}_i^t,$$

where $\boldsymbol{y}_i^t$ is the true label vector and N^t is the number of instances for task t.

If we ignore the inter-dependencies between nodes and edges, we can simply predict the label with the highest score for each knowledge element and thus generate the locally best graph $\hat{G}$. The score of $\hat{G}$ can be calculated as

$$s'(\hat{G}) = \sum_{t \in T} \sum_{i=1}^{N^t} \max \hat{\boldsymbol{y}}_i^t,$$

where T is the set of tasks. We refer to $s'(\hat{G})$ as the local score of $\hat{G}$.

[2] We use the BIO tag scheme, in which the prefix B- marks the beginning of a mention, and I- means inside of a mention. A token not belonging to any mention is tagged with O.

Categary	Description
Role	1. The number of entities that act as <role$_i$> and <role$_j$> arguments at the same time.
	2. The number of <event_type$_i$> events with <number> <role$_j$> arguments.
	3. The number of occurrences of <event_type$_i$>, <role$_j$>, and <entity_type$_k$> combination.
	4. The number of events that have multiple <role$_i$> arguments.
	5. The number of entities that act as a <role$_i$> argument of an <event_type$_j$> event and a <role$_k$> argument of an <event_type$_l$> event at the same time.
Relation	6. The number of occurrences of <entity_type$_i$>, <entity_type$_j$>, and <relation_type$_k$> combination.
	7. The number of occurrences of <entity_type$_i$> and <relation_type$_j$> combination.
	8. The number of occurrences of a <relation_type$_i$> relation between a <role$_j$> argument and a <role$_k$> argument of the same event.
	9. The number of entities that have a <relation_type$_i$> relation with multiple entities.
	10. The number of entities involving in <relation_type$_i$> and <relation_type$_j$> relations simultaneously.
Trigger	11. Whether a graph contains more than one <event_type$_i$> event.

Table 1: Global feature categories.

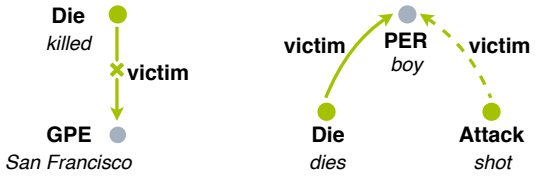
3.4 Global Features

A limitation of local classifiers is that they are incapable of capturing inter-dependencies between knowledge elements in an information network. We consider two types of inter-dependencies in our framework.

The first type of inter-dependency is **Cross-subtask interactions** between entities, relations, and events. Consider the following sentence. *"A civilian aid worker from **San Francisco** was **killed** in an attack in Afghanistan."* A local classifier may predict "San Francisco" as a VICTIM argument because an entity mention preceding "was killed" is usually the victim despite the fact that a GPE is unlikely to be a VICTIM. To impose such constraints, we design a global feature as shown in Figure 3(a) to evaluate whether the structure DIE-VICTIM-GPE exists in a candidate graph.

Another type of inter-dependency is **Cross-instance interactions** between multiple event and/or relation instances in the sentence. Take the following sentence as an example. "*South Carolina **boy**, 9, dies during hunting trip after his father accidentally **shot** him on Thanksgiving Day.*" It can be challenging for a local classifier to predict "boy" as the VICTIM of the ATTACK event triggered by "shot" due to the long distance between these two words. However, as shown in Figure 3(b), if an entity is the VICTIM of a DIE event, it is also likely to be the VICTIM of an ATTACK event in the same sentence.

Motivated by these observations, we design a set of global feature templates (event schemas) as listed in Table 1 to capture cross-subtask and cross-instance interactions, while the model fills in all possible types to generate features and learns the



(a) Cross-subtask Interaction (b) Cross-instance Interactions

Figure 3: Examples of inter-dependencies between elements in information networks. (a) A VICTIM edge is unlikely to exist between a GPE entity and a DIE event trigger. (b) The VICTIM of a DIE event is likely to be the VICTIM of an ATTACK event in the same sentence.

weight of each feature during training. Given a graph G, we represent its global feature vector as $\boldsymbol{f}_G = \{f_1(G), ..., f_M(G)\}$, where M is the number of global features and $f_i(\cdot)$ is a function that evaluates a certain feature and returns a scalar. For example,

$$f_i(G) = \begin{cases} 1, & G \text{ has multiple ATTACK events} \\ 0, & \text{otherwise.} \end{cases}$$

Next, ONEIE learns a weight vector $\boldsymbol{u} \in \mathbb{R}^M$ and calculates the global feature score of G as the dot product of $\boldsymbol{f}_G$ and $\boldsymbol{u}$. We define the global score of G as the sum of its local score and global feature score, namely

$$s(G) = s'(G) + \boldsymbol{u}\boldsymbol{f}_G,$$

We make the assumption that the gold-standard graph for a sentence should achieve the highest global score. Therefore, we minimize the following loss function

$$\mathcal{L}^{\mathrm{G}} = s(\hat{G}) - s(G),$$

where $\hat{G}$ is the graph predicted by local classifiers and G is the gold-standard graph.

Finally, we optimize the following joint objective function during training

$$\mathcal{L} = \mathcal{L}^I + \sum_{t \in T} \mathcal{L}^t + \mathcal{L}^G$$

3.5 Decoding

As we have discussed, because local classifiers ignore interactions among elements in an information network, they may predict contradictory results or fail to predict difficult edges that require information from other elements. In order to address these issues, ONEIE makes a joint decision for all nodes and their pairwise edges to obtain the globally optimal graph. The basic idea is to calculate the global score for each candidate graph and select the one with the highest score. However, exhaustive search is infeasible in many cases as the size of search space grows exponentially with the number of nodes. Therefore, we design a beam search-based decoder as Figure 4 depicts.

Given a set of identified nodes V and the label scores for all nodes and their pairwise links, we perform decoding with an initial beam set $B = \{K_0\}$, where K_0 is an order-zero graph. At each step i, we expand each candidate in B in node step and edge step as follows.

Node step: We select $v_i \in V$ and define its candidate set as $V_i = \{\langle a_i, b_i, l_i^{(k)} \rangle | 1 \leq k \leq \beta_v\}$, where $l_i^{(k)}$ denotes the label with the k-th highest local score for v_i, and β_v is a hyper-parameter that controls the number of candidate labels to consider. We update the beam set by

$$B \leftarrow \{G + v | (G, v) \in B \times V_i\},$$

Edge step: We iteratively select a previous node $v_j \in V, j < i$ and add possible edges between v_j and v_i. Note that if v_i is a trigger, we skip v_j if it is also a trigger. At each iteration, we construct a candidate edge set as $E_{ij} = \{\langle j, i, l_{ij}^{(k)} \rangle | 1 \leq k \leq \beta_e\}$, where $l_{ij}^{(k)}$ is the label with k-th highest score for e_{ij} and β_e is a threshold for the number of candidate labels. Next, we update the beam set by

$$B \leftarrow \{G + e | (G, e) \in B \times E_{ij}\},$$

At the end of each edge step, if $|B|$ is larger than the beam width θ, we rank all candidates by global score in descending order and keep the top θ ones. After the last step, we return the graph with the highest global score as the information network for the input sentence.

4 Experiments

4.1 Data

We perform our experiments on the Automatic Content Extraction (ACE) 2005 dataset[3], which provides entity, value, time, relation, and event annotations for English, Chinese, and Arabic. Following Wadden et al. (2019)'s pre-processing[4], we conduct experiments on two datasets, ACE05-R that includes named entity and relation annotations, and ACE05-E that includes entity, relation, and event annotations. We keep 7 entity types, 6 coarse-grained relation types, 33 event types, and 22 argument roles.

In order to reinstate some important elements absent from ACE05-R and ACE05-E, we create a new dataset, ACE05-E$^+$, by adding back the order of relation arguments, pronouns, and multi-token event triggers, which have been largely ignored in previous work. We also skip lines before the `<text>` tag (e.g., headline, datetime) as they are not annotated.

In addition to ACE, we derive another dataset, ERE-EN, from the Entities, Relations and Events (ERE) annotation task created under the Deep Exploration and Filtering of Test (DEFT) program because it covers more recent articles. Specifically, we extract 458 documents and 16,516 sentences from three ERE datasets, LDC2015E29, LDC2015E68, and LDC2015E78. For ERE-EN, we keep 7 entity types, 5 relation types, 38 event types, and 20 argument roles.

To evaluate the portability of our model, we also develop a Chinese dataset from ACE2005 and a Spanish dataset from ERE (LDC2015E107). We refer to these datasets as ACE05-CN and ERE-ES respectively.

4.2 Experimental Setup

We optimize our model with BertAdam for 80 epochs with a learning rate of 5e-5 and weight decay of 1e-5 for BERT, and a learning rate of 1e-3 and weight decay of 1e-3 for other parameters. We use the bert-base-multilingual-cased model[5] for

[3] https://www.ldc.upenn.edu/collaborations/past-projects/ace

[4] https://github.com/dwadden/dygiepp

[5] https://huggingface.co/transformers/pretrained_models.html

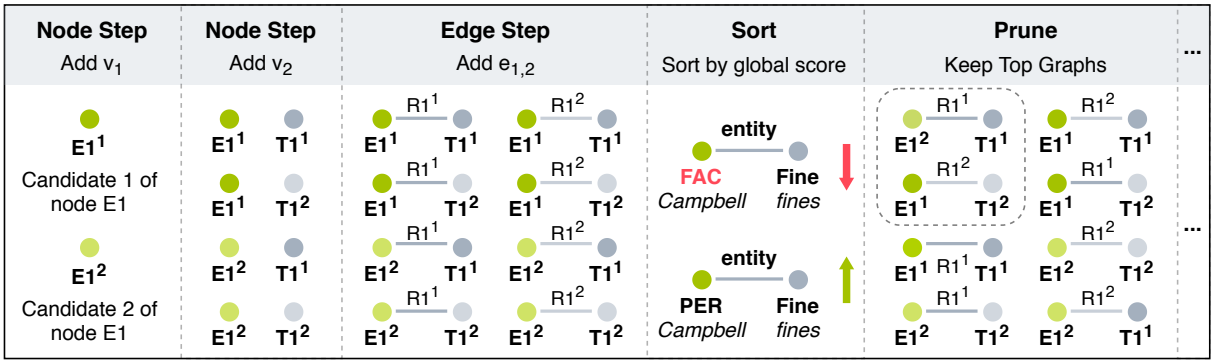


Figure 4: An illustration of our decoding algorithm. At each step, we expand each candidate graph by adding a new node and possible edges between it and existing nodes. After that, we rank all expanded graphs and keep the top ones.

Dataset	Split	#Sents	#Entities	#Rels	#Events
ACE05-R	Train	10,051	26,473	4,788	-
	Dev	2,424	6,362	1,131	-
	Test	2,050	5,476	1,151	-
ACE05-E	Train	17,172	29.006	4,664	4,202
	Dev	923	2,451	560	450
	Test	832	3,017	636	403
ACE05-CN	Train	6,841	29,657	7,934	2,926
	Dev	526	2,250	596	217
	Test	547	2,388	672	190
ACE05-E$^+$	Train	19,240	47,525	7,152	4,419
	Dev	902	3,422	728	468
	Test	676	3,673	802	424
ERE-EN	Train	14,219	38,864	5,045	6,419
	Dev	1,162	3,320	424	552
	Test	1,129	3,291	477	559
ERE-ES	Train	7,067	11,839	1,698	3,272
	Dev	556	886	120	210
	Test	546	811	108	269

Table 2: Dataset statistics.

ACE05-CN and ERE-ES, and use the bert-large-cased model for other datasets. Following (Wadden et al., 2019), we use two-layer FFNs with a dropout rate of 0.4 for local classifiers. We use 150 hidden units for entity and relation extraction, and 600 hidden units for event extraction. For global features, we set β_v and β_e to 2 and set θ to 10. In our experiments, we use random seeds and report averaged scores across runs. We use the same criteria as (Zhang et al., 2019; Wadden et al., 2019) for evaluation as follows.

- **Entity**: An entity mention is correct if its offsets and type match a reference entity.
- **Relation**: A relation is correct if its relation type is correct and the offsets of the related entity mentions are correct.
- **Trigger**: A trigger is correctly identified (Trig-I) if its offsets match a reference trigger. It is correctly classified (Trig-C) if its event type also matches the reference trigger.
- **Argument**: An argument is correctly identified (Arg-I) if its offsets and event type match a reference argument mention. It is correctly classified (Arg-C) if its role label also matches the reference argument mention.

4.3 Overall Performance

In Table 3, we compare our results with two models: (1) DyGIE++ (Wadden et al., 2019), the state-of-the-art end-to-end IE model that utilizes multi-sentence BERT encodings and span graph propagation; (2) Baseline that follows the architecture of OneIE but only uses the output of the last layer of BERT and local classifiers. We can see that our model consistently outperforms DyGIE++ and Baseline on ACE05-R and ACE05-E.

In (Wadden et al., 2019), the authors show that combining triggers predicted by a four-model ensemble optimized for trigger detection can improve the performance of event extraction. While we also report our results using a four-model ensemble in Table 4 for fair comparison, we hold the opinion that the single-model scores in Table 3 better reflect the actual performance of OneIE and should be used for future comparison.

Table 5 shows the performance of OneIE on two new datasets, ACE05-E$^+$ and ERE-EN.

In Table 6 we list salient global features learned by the model. Take feature #9 as an example, if a candidate graph contains multiple ORG-AFF edges incident to the same node, the model will demote this graph by adding a negative value into its global

Dataset	Task	DyGIE++	Baseline	OneIE
ACE05-R	Entity	88.6	-	**88.8**
	Relation	63.4	-	**67.5**
ACE05-E	Entity	89.7	**90.2**	**90.2**
	Trig-I	-	76.6	**78.2**
	Trig-C	69.7	73.5	**74.7**
	Arg-I	53.0	56.4	**59.2**
	Arg-C	48.8	53.9	**56.8**

Table 3: Results on ACE2005 datasets (F-score, %).

Dataset	Task	DyGIE++*	OneIE*
ACE05-E	Entity	**90.7**	90.3
	Trig-I	76.5	**78.6**
	Trig-C	73.6	**75.2**
	Arg-I	55.4	**60.7**
	Arg-C	52.5	**58.6**

Table 4: Experiment results on ACE05-E (F-score, %). DyGIE++* and OneIE* use a four-model ensemble optimized for trigger detection.

Dataset	Entity	Trig-I	Trig-C	Arg-I	Arg-C	Relation
ACE05-E$^+$	89.6	75.6	72.8	57.3	54.8	58.6
ERE-EN	87.0	68.4	57.0	50.1	46.5	53.2

Table 5: New benchmark results (F-score, %).

score. We also observe that the weights of about 9% global features are almost not updated, which indicates that they are barely found in both gold-standard and predicted graphs. In Table 8, we perform qualitative analysis on concrete examples.

4.4 Porting to Another Language

As Table 7, we evaluate the proposed framework on ACE05-CN and ERE-ES. The results show that OneIE works well on Chinese and Spanish data without any special design for the new language. We also observe that adding English training data can improve the performance on Chinese and Spanish.

4.5 Remaining Challenges

We have analyzed 75 of the remaining errors and in Figure 5 we present the distribution of various error types which need more features and knowledge acquisition to address in the future. In this section, we will discuss some main categories with examples.

Need background knowledge. Most of current IE methods ignore external knowledge such as entity attributes and scenario models. For example, in the following sentence, "*And Putin's media aide, Sergei Yastrzhembsky, told Kommersant Russia would not forgive the Iraqi debt*", our model

	Positive Feature	Weight
1	A Transport event has only one Destination argument	2.61
2	An Attack event has only one Place argument	2.31
3	A Transport event has only one Origin argument	2.01
4	An End-Position event has only one Person argument	1.51
5	A Per-Soc relation exists between two Per entities	1.08
6	A Gen-Aff relation exists between Org and Loc entities	0.96
7	A Beneficiary argument is a Per entity	0.93
8	A Gen-Aff relation exists between Org and Gpe entities	0.90

	Negative Feature	Weight
9	An entity has an Org-Aff relation with multiple entities	-3.21
10	An entity has an Part-Whole relation with multiple entities	-2.49
11	An event has two Place arguments	-2.47
12	A Transport event has multiple Destination arguments	-2.25
13	An entity has a Gen-Aff relation with multiple entities	-2.02
14	An Attack event has multiple Place arguments	-1.86
15	An entity has a Phys relation with multiple entities	-1.69
16	An event has multiple Victim arguments	-1.61

Table 6: Salient positive and negative global features.

Dataset	Training	Entity	Relation	Trig-C	Arg-C
ACE05-CN	CN	88.5	62.4	65.6	52.0
	CN+EN	89.8	62.9	67.7	53.2
ERE-ES	ES	81.3	48.1	56.8	40.3
	ES+EN	81.8	52.9	59.1	42.3

Table 7: Results on ACE05-CN and ERE-ES (F-score, %). For ACE05-CN, EN refers to ACE05-E$^+$. For ERE-ES, EN refers to ERE-EN.

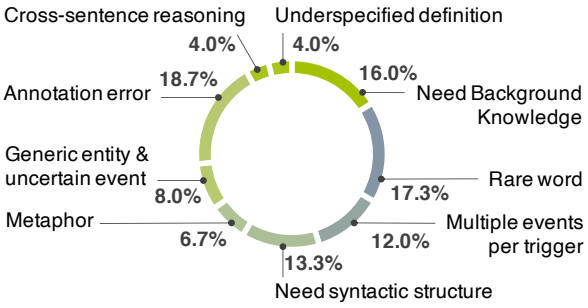


Figure 5: Distribution of remaining errors.

mistakenly identifies "Kommersan" as a person instead of organization. With entity linking, we can correct this error based on the first sentence in its Wikipedia page "*Kommersant is a nationally distributed daily newspaper published in Russia*

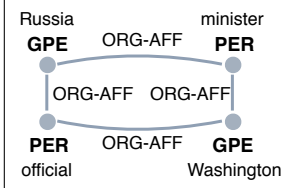
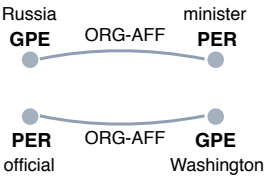
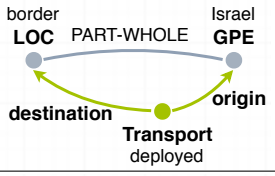
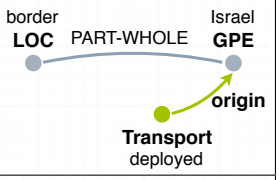
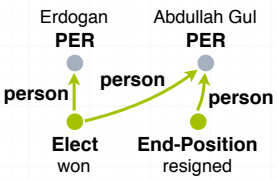
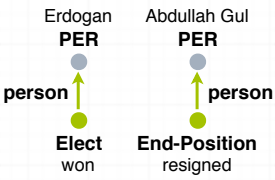
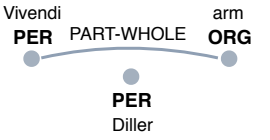
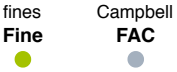
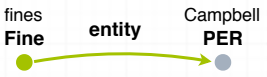
Sentence & Analysis	Baseline	+Global Features
#1: **Russia**'s foreign **minister** expressed outrage at suggestions from a top **Washington official** last week that Moscow should forgive the eight billion dollars in Soviet-era debt that Baghdad owes it, as a gesture of good will. ⋆ Global feature category: 8 ⋆ Analysis: It is unlikely for a person to have an ORG-AFF relation with multiple entities.		
#2: They also **deployed** along the **border** with **Israel**. ⋆ Global feature category: 9 ⋆ Analysis: It is uncommon that an ORIGIN argument and a DESTINATION argument have a PART-WHOLE relation.		
#3: Prime Minister **Abdullah Gul resigned** earlier Tuesday to make way for **Erdogan** , who **won** a parliamentary seat in by-elections Sunday. ⋆ Global feature categories: 2 and 5 ⋆ Analysis: 1. An ELECT usually has only one PERSON argument; 2. An entity is unlikely to act as a PERSON argument for END-POSITION and ELECT events at the same time.		
#4: Diller will continue to play a critical role in the future of **Vivendi** 's entertainment **arm**. ⋆ Global feature category: 6 ⋆ Analysis: A PART-WHOLE relation should not exist between PER and ORG entities.		
#5: He also brought a check from **Campbell** to **pay** the fines and fees. ⋆ Global feature category: 3 ⋆ Analysis: As "Campbell" is likely to be an ENTITY argument of a FINE event, the model corrects its entity type from FAC to PER.		

Table 8: Examples showing how global features improve the quality of extracted information networks. For some sentences, we do not draw the whole information network.

mostly devoted to politics and business".

Rare words. The second challenge is the famous long-tail problem: many triggers, entity mentions (e.g., "caretaker", "Gazeta.ru") and contextual phrases in the test data rarely appear in the training data. While most event triggers are verbs or nouns, some adverbs and multi-word expressions can also serve as triggers.

Multiple types per trigger. Some trigger words may indicate both the procedure and the result status of an action. For example, "named" may indicate both NOMINATE and START-POSITION events; "killed" and "eliminate" may indicate both ATTACK and DIE events. In these cases the human ground truth usually only annotates the procedure types, whereas our system produces the resultant event types.

Need syntactic structure. Our model may benefit from deeper syntactic analysis. For example, in the following sentence "*As well as previously holding senior positions at Barclays Bank, BZW and Kleinwort Benson, McCarthy was formerly a top civil servant at the Department of Trade and Industry*", our model misses all of the employers "Barclays Bank", "BZW" and "Kleinwort Benson" for "McCarthy" probably because they appear in a previous sub-sentence.

Uncertain events and metaphors. Our model mistakenly labels some future planned events as specific events because its lacking of tense prediction and metaphor recognition. For example, START-ORG triggered by "formation" does not happen in the following sentence "*The statement did not give any reason for the move, but said Lahoud would begin consultations Wednesday aimed at the formation of a new government*". Our model also mistakenly identifies "camp" as a facility, and a DIE event triggered by "dying" in the following sentence "*Russia hints 'peace camp' alliance with Germany and France is dying by Dmitry Zaks.*".

The IE community is lacking of newer data sets with end-to-end annotations. Unfortunately, the annotation quality of the ACE data set is not perfect due to some long-term debates on the annotation guideline; e.g., Should "government" be tagged as a GPE or an ORG? Should "dead" be both an entity and event trigger? Should we consider designator word as part of the entity mention or not?

5 Related Work

Previous work (Roth and Yih, 2004; Li et al., 2011) encodes inter-dependency among knowledge elements as global constraints in an integer linear programming framework to effectively remove extraction errors. Such integrity verification results can be used to find knowledge elements that violate the constraints and identify possible instances of detector errors or failures. Inspired by these previous efforts, we propose a joint neural framework with global features in which the weights are learned during training. Similar to (Li et al., 2014)'s method, ONEIE also uses global features to capture cross-subtask and cross-instance interdependencies, while our features are language-independent and do not rely on other NLP tools such as dependency parsers. Our methods also differ in local features, optimization methods, and decoding procedures.

Some recent efforts develop joint neural models to perform extraction of two IE subtasks, such as entity and relation extraction (Zheng et al., 2017; Katiyar and Cardie, 2017; Bekoulis et al., 2018; Fu et al., 2019; Luan et al., 2019; Sun et al., 2019) and event and temporal relation extraction (Han et al., 2019). Wadden et al. (2019) design a joint model to extract entities, relations and events based on BERT and dynamic span graphs. Our framework extends (Wadden et al., 2019) by incorporating global features based on cross-subtask and cross-instance constraints. Unlike (Wadden et al., 2019) that uses a span-based method to extract mentions, we adopt a CRF-based tagger in our framework because it can extract mentions of any length, not restricted by the maximum span width.

6 Conclusions and Future Work

We propose a joint end-to-end IE framework that incorporates global features to capture the inter-dependency between knowledge elements. Experiments show that our framework achieves better or comparable performance compared to the state of the art and prove the effectiveness of global features. Our framework is also proved to be language-independent and can be applied to other languages, and it can benefit from multi-lingual training.

In the future, we plan to incorporate more comprehensive event schemas that are automatically induced from multilingual multimedia data and external knowledge to further improve the quality of IE. We also plan to extend our framework to more IE subtasks such as document-level entity coreference resolution and event coreference resolution.

Acknowledgement

This research is based upon work supported in part by U.S. DARPA KAIROS Program No. FA8750-19-2-1004, U.S. DARPA AIDA Program No. FA8750-18-2-0014, Air Force No. FA8650-17-C-7715, the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via contract No. FA8650-17-C-9116. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA, ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018. Adversarial training for multi-context joint entity and relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP2015)*.

Jason Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association of Computational Linguistics (TACL)*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep

bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT2019)*.

Greg Durrett and Dan Klein. 2014. A joint model for entity analysis: Coreference, typing, and linking. In *Transactions of the Association for Computational Linguistics (TACL)*.

Tsu-Jui Fu, Peng-Hsuan Li, and Wei-Yun Ma. 2019. GraphRel: Modeling text as relational graphs for joint entity and relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL2019)*.

Rujun Han, Qiang Ning, and Nanyun Peng. 2019. Joint event and temporal relation extraction with shared representations and structured prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP2019)*.

Heng Ji and Ralph Grishman. 2005. Improving name tagging by reference resolution and relation detection. In *In Proceedings of ACL 05, Ann Arbor, USA*.

Heng Ji, David Westbrook, and Ralph Grishman. 2005. Using semantic relations to refine coreference decisions. In *In Proceedings of HLT/EMNLP 05, Vancouver, B.C., Canada*.

Arzoo Katiyar and Claire Cardie. 2017. Going out on a limb: Joint extraction of entity mentions and relations without dependency trees. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL2017)*.

Johannes Kirschnick, Holmer Hemsen, and Volker Markl. 2016. JEDI: Joint entity and relation detection using type inference. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics System Demonstrations (ACL2016)*.

Qi Li, Sam Anzaroot, Wen-Pin Lin, Xiang Li, and Heng Ji. 2011. Joint inference for cross-document information extraction. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM2011)*.

Qi Li, Heng Ji, HONG Yu, and Sujian Li. 2014. Constructing information networks using one single model. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP2014)*.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL2016)*.

Wei Lu and Dan Roth. 2015. Joint mention extraction and classification with mention hypergraphs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP2015)*.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT2019)*.

Makoto Miwa and Yutaka Sasaki. 2014. Modeling joint entity and relation extraction with table representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP2014)*.

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 (CoNLL2004)*.

Avirup Sil and Alexander Yates. 2013. Re-ranking for joint named-entity recognition and linking. In *Proceedings of the 22nd ACM international conference on Conference on Information & Knowledge Management (CIKM2013)*.

Changzhi Sun, Yeyun Gong, Yuanbin Wu, Ming Gong, Daxin Jiang, Man Lan, Shiliang Sun, and Nan Duan. 2019. Joint type inference on entities and relations via graph convolutional networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL2019)*.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh 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-IJCNLP2019)*.

Bishan Yang and Tom M. Mitchell. 2016. Joint extraction of events and entities within a document context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT2016)*.

Tongtao Zhang, Heng Ji, and Avirup Sil. 2019. Joint entity and event extraction with generative adversarial imitation learning. *Data Intelligence*.

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 (ACL2017)*.