DEGREE: A Data-Efficient Generative Event Extraction Model

I-Hung Hsu†‡, Kuan-Hao Huang†‡, Elizabeth Boschee†, Scott Miller†,
Prem Natarajan‡, Kai-Wei Chang‡, Nanyun Peng†‡

†Information Science Institute, University of Southern California
‡Computer Science Department, University of California, Los Angeles

{iunghsu, boschee, smiller, pntarat}@isi.edu
{khhuang, kwchang, violetpeng}@cs.ucla.edu

Abstract
Event extraction (EE) aims to identify structured events, including event triggers and their corresponding arguments, from unstructured text. Most of the existing works rely on a large number of labeled instances to train models, while the labeled data could be expensive to be obtained. In this work, we present a data-efficient event extraction method by formulating event extraction as a natural language generation problem. The formulation allows us to inject knowledge of label semantics, event structure, and output dependencies into the model. Given a passage and an event type, our model learns to summarize this passage into a templated sentence in a predefined structure. The template is event-type-specific, manually created, and contains event trigger and argument information. Lastly, a rule-based algorithm is used to derive the trigger and argument predictions from the generated sentence. Our method inherently enjoys the following benefits: (1) The pretraining of the generative language models help incorporate the semantics of the labels for generative EE. (2) The autoregressive generation process and our end-to-end design for extracting triggers and arguments force the model to capture the dependencies among the output triggers and their arguments. (3) The predefined templates form concrete yet flexible rules to hint the models about the valid patterns for each event type, reducing the models’ burden to learn structures from the data. Empirical results show that our model achieves superior performance over strong baselines on EE tasks in the low data regime and achieves competitive results to the current state-of-the-art when more data becomes available.

1 Introduction
Event extraction (EE) recognizes events of specific types from texts. Each type of event contains an expressive structure formed by the role of the participants that involve in the event. For example, in Figure 1, the structure for a Justice:Execute type event contains three components, including an Agent who carries out the execution, the Person who been executed, and the Place where the event occurs. In literature, EE is often divided into two subtasks: (1) event detection, which identifies event triggers (execution) and their types (Justice:Execute), and (2) event argument extraction, which extracts the entities that are the participants of the event and their corresponding roles (e.g. convicts as Person role). EE has been shown to benefit a wide range of applications, e.g., building knowledge graphs (Zhang et al. 2020), question answering (Berant et al. 2014), and supporting decision making (Hogenboom et al. 2016).

Most prior studies in EE require a large number of high-quality annotated data (Li, Ji, and Huang 2013; Nguyen and Grishman 2013; Nguyen, Cho, and Grishman 2016), which is expensive to be obtained. For example, the ACE 2005 English corpus (Doddington et al. 2004) requires two rounds of annotation by people with linguistic backgrounds. This hinders the current techniques to be widely used for new domains and new event types.

To build data-efficient EE models, we propose a generative approach that models three important aspects of the EE task. (1) Label semantics: the semantic of a label (either an event type or a role type) contains rich information for models to efficiently characterize the attribute of that label. For example, the role type “Adjudicator” indicates a person or group that makes an official decision about something, usually being a judge or court, could help model identify “Supreme Court” in the example in Figure 1. (2) Output dependencies: although EE can be decomposed into several individual prediction problems, any prediction made without the consideration of other predictions could lose some crucial information. For example, when Indonesia has been predicted as an Agent for the Justice:Execute event,
it is less possible for Indonesia to be the Person been executed. (3) Event structure: each event type has its predefined structure. For instance, Figure 1 shows that event type Justice:Appeal contains four valid roles; each role has only a few possible entity types that can match. For example, only person (PER), organizations (ORG), or geographical entities (GPE) can be Prosecutor in a Justice:Appeal event. This structural information provides an expressive prior that could reduce the amount of required data for learning.

In this work, we frame EE as a natural language generation task and propose DEGREE (Data-Efficient GeneRative Event Extraction model) that can capture event structures, label semantics, and output dependencies simultaneously. DEGREE is an encoder-decoder generative model initialized by pretrained sequence-to-sequence models (Lewis et al. 2020). Given a passage, DEGREE learns to summarize the passage into a sentence following a predefined template, from where the final triggers and arguments can be extracted. The predefined templates, one for each event type, are created by referencing the event structure to provide structural clues to the model. The textual form and the definition of the event and argument types are leveraged to incorporate label semantics during generation. Finally, our method models complex output dependencies because of (1) its autoregressive decoding process—each new word prediction depends on the words that have been generated, and (2) its end-to-end model design—the triggers and their corresponding arguments are predicted at once.

We evaluate our method on two EE datasets: ACE 2005 (English) (Doddington et al. 2004) and ERE-EN (Song et al. 2015). The experimental results show that DEGREE outperforms prior works by a large margin when only extremely low training data is available. The proposed approach also achieves competitive or even better performance than the current state-of-the-art (Lin et al. 2020) when more training data is available. The results demonstrate the effectiveness of our method for the EE task.

2 Related Work

Classification-based event extraction. Event extraction (EE) has been studied for over a decade (Ahn 2006; Ji and Grishman 2008) and is usually formulated as a classification problem (Nguyen, Cho, and Grishman 2016; Wang et al. 2019; Yang et al. 2019; Wadden et al. 2019). Most of them (Nguyen, Cho, and Grishman 2016; Yang et al. 2019; Wadden et al. 2019) follow a pipelined design, which leads to the error propagation problem and disallows the interactions between each local prediction. Consequently, some works propose to incorporate global features to capture the dependencies among each local classifier and apply joint inference (Lin et al. 2020; Li, Ji, and Huang 2013; Yang and Mitchell 2016) so as to exploit output dependencies.

Despite improvements, these models usually lack the exploitation of label semantics, since they mostly use a numeric label to represent event types or role types.

Event extraction as MRC. Recent work attempts to cast EE as a machine reading comprehension (MRC) problem (Liu et al. 2020; Du and Cardie 2020; Li et al. 2020). They use different strategies to frame questions for querying event triggers and arguments. Despite differences, most of them follow a pipelined question answering paradigm, which first asks a trigger question to query trigger tokens in the text, and then for each valid argument role, they separately ask designed questions to get the argument predictions. This multistep procedure hinders their model to capture output dependencies.

Generative model for event extraction. TANL (Paolini et al. 2021), which treats EE tasks as translation tasks between augmented natural languages, is a generative model that considers output dependencies by its autoregressive generation process. Their predicted target—augmented language embed labels into the input passage via using brackets and vertical bar symbols to mark the tokens for corresponding labels. Compared to our method that have event type specific output structure, the predicted targets for TANL are in an unified format, which resulting in their stronger dependency on data to learn the valid patterns for each event type. That is, our method explicitly exploits event structure, which TANL needs to learn from data.

Another proposed generative method, BART-Gen (Li, Ji, and Han 2021), focuses on document-level event argument extraction. The work shares a similar design with us in using templates to extract event arguments, while we focus on a more challenging setting—dealing with end-to-end event extraction. Additionally, their templates for event argument extraction contain various special tokens as placeholders. Conversely, in our design, all placeholders are natural words, which is shown to be useful in Section 6. The results in Section 5 also demonstrate the superiority of our method in both end-to-end event extraction and argument extraction.

Low-resource event extraction. It has been a rising interest working on event extraction under less data scenario. (Liu et al. 2020) utilize the MRC formulation and leverage label semantics to conduct event extraction in low-resource regime. Text2Event (Lu et al. 2021), a sequence-to-structure generation paradigm, first presents events in a linearized format, and then train a generative model to generate the linearized event sequence. This paradigm get success in both high-resource and low-resource EE tasks, however, their unnatural output format hinders the model to fully leverage pretrained knowledge. Hence, their model falls short on the cases with only extremely low-data being available (as shown in Section 5).

Another thread of works are using meta-learning to deal with the less label challenge (Deng et al. 2020; Shen et al. 2021; Cong et al. 2021). However, their method can only be applied on event detection, which differs from our main focus on end-to-end event extraction.

3 Method

We formulate event extraction (EE) as a natural language generation task and propose DEGREE, a Data-Efficient GeneRative Event Extraction model, to conduct end-to-end event extraction. The key idea is to generate a sentence that contains triggers and arguments from the given passage. The
Earlier Monday, a 19-year-old Palestinian riding a bicycle detonated a 30-kilo (66-pound) bomb near a military jeep in the Gaza Strip, injuring three soldiers.

Figure 2: An illustration of our method. Given an event type, we feed a sequence containing a passage and a set of prompts to the model. The prompts vary depending on the queried type and the task (ED, EAE, or end-to-end). The model is trained to generate output texts following event-type-specific templates, which contain placeholders for filling triggers or arguments. The final event predictions can be made by comparing the template and the output text.

generated sentence needs to be structural so that the key information about an event (event trigger and arguments) is all retained and the final derivation of triggers and arguments from the generated sentence can be done efficiently.

To achieve this goal, we design event-type-specific templates to encode the unique structure of each event type. Given a passage and an event type, we provide the template of the queried event type to the model in order to guide DEGREE to generate output in the desired format so that the final derivation can be conducted by comparing the template with the generated sentence. To be more specific, the input to the model is the input passage concatenated with a carefully designed prompt with a special separator [SEP] in between. The prompt includes elements that provide semantic information about the queried event and a predefined template for the queried event type. Then, the generative model is trained to output a sentence adhering to the predefined template. The whole process is illustrated in Figure 2.

While our focus is end-to-end event extraction, the design of our framework is general and can be applicable to sub-tasks of event extraction – event trigger detection and event argument extraction by simply tweaking the prompts that are supplied to the model. We use DEGREE(ED) and DEGREE(EAE) to represent the two variations. For the ease of understanding, we first introduce DEGREE(ED) (Section 3.1) and DEGREE(EAE) (Section 3.2) and then detail our end-to-end DEGREE model subsequently (Section 3.3).

3.1 Event Detection Model

DEGREE(ED) is designed to extract triggers given an event type. The prompt for DEGREE(ED) contains:

- **Event type definition** provides a definition for the given event type\(^2\). For example, “The event is related to conflict and some violent physical act.” describes a “Conflict:Attack” event.

- **Event keywords** presents some words that are semantically related to the given event type. In practice, we collect three words that appear as triggers in the example sentences from the annotation guidelines.

- **ED template** is the predefined output format for event detection tasks, which is designed as “Event trigger is <Trigger>.”, where “<Trigger>” is a special token. It is included in the prompt in order to guide the generation.

The goal of DEGREE(ED) is to generate an output that replaces “<Trigger>” in the ED template with one or more triggers of the appropriate type from the given passage, if they are present. For example, the golden output text for DEGREE(ED) in Figure 2 is “Event trigger is detonated.”.

\(^2\)The definition can be derived from the annotation guidelines for each dataset, e.g. the guideline for ACE.
where “detonated” is the trigger for the queried “Conflict:Attack” type. When there are multiple triggers, we connect the triggers with the word “and”. For cases containing no triggers for the given event type, the model learns to copy the ED template, i.e., “Event trigger is <Trigger>.”.

During inference, DEGREE(ED) must run once for each event type in the target ontology, since it only considers one event type at a time. During training, we randomly select m event types as queries for each example to avoid the training signal being too sparse, where m is a hyper-parameter.

3.2 Event Argument Extraction Model
DEGREE(EAE) is designed to identify arguments and their corresponding roles for a given event trigger. Similar to DEGREE(ED), we design three components of prompts for DEGREE(EAE), as illustrated in Figure 2:

- **Event type definition** is the same as the one for DEGREE(ED).
- **Query trigger** indicates the trigger word whose arguments are being sought, e.g. “detonated” in Figure 2.
- **EAE Template** is an event-type-specific natural language style template. Distinct templates are devised for each type to represent the unique structure of each event type. For example, in Figure 2, “some attacker attacked some facility, someone, or some organization by some way in somewhere” is the EAE template tailored for “Conflict:Attack” events. Each underlined part starting with “some-” serves as a placeholder corresponding to an argument role in an “Conflict:Attack” event. The full list of EAE templates can be found in the Appendix D.

The strategy for creating an EAE template is to first identify all valid argument roles for the event type, such as Attacker, Target, Instrument, and Place roles. Then, for each argument role, according to the semantics of the role type, we select natural and fluent words to form its placeholder (e.g., some way for Instrument). This design aims to provide a simple way to help the model learn both the roles’ label semantics and the event structure. Finally, we create a natural language sentence that connects all these placeholders. More examples can refer to Appendix C and Appendix D.

Notice that we intentionally write templates in natural sentences. This encourages the model to fully leverage the pre-training knowledge from pretrained models. In the latter section (Section 6), we will provide experiments to justify the effectiveness of such design.

The objective for DEGREE(EAE) is to generate output texts that replace placeholders with arguments in EAE template whenever possible (e.g., substitute some where with Gaza Strip).

3.3 DEGREE
To fully capture the output dependencies within events for end-to-end event extraction, we introduce DEGREE, which, for a given event type, generates an output text containing all triggers and their corresponding arguments for that type at once. The input prompt design of DEGREE is similar to that of DEGREE(ED) except for the creation of a new E2E template, which is a combination of ED template and EAE template, as shown in Figure 2. Then, DEGREE learns to generate texts that replace the placeholders in the E2E template with real triggers and arguments.

To perform event extraction task, another alternative is to cascading DEGREE(ED) with DEGREE(EAE), forming DEGREE(ED+EAE). Yet, compared to the pipelined approach, DEGREE can enforce the integration of the output dependencies from arguments to triggers, which is especially important when only extremely low training data is available. We will justify this end-to-end design in Section 4.

3.4 Decode Predictions from Output Text
Since DEGREE is a text generation model, to express our model’s predictions in span-based format, we apply a deterministic algorithm to the generated output text. There are two steps to get the final span offset for each prediction— (1) We compare our predefined templates with the output text to detect the placeholders being replaced. The words that used for substitution are our predictions in textual form. (2) To transform the textual form predictions to their span offsets in the text, we first use a tokenizer to tokenize the passage and then look for a continuous exact match span that is the same as our tokenized predicted text. The matched span is then our final output. Our experiments are evaluated with the span offsets we generated via this procedure.

4 Data-Scarce Event Extraction
4.1 Experimental Settings
Datasets. We consider ACE 2005 (Doddington et al. 2004), a widely used EE dataset which provides entity, value, time, relation, and event annotations. We follow the pre-processing in Wadden et al. (2019) and Lin et al. (2020), resulting in two variants: ACE05-E and ACE05-E+. The two variants differ because ACE05-E filters out multi-token event triggers, but both of them contain 33 event types and 22 argument roles. In addition, we consider ERE-EN (Song et al. 2015) and adopt the pre-processing in Lin et al. (2020), which keeps 38 event types and 21 argument roles.

Data split. There is no standard data splits for prior works that have studied end-to-end EE in low-data regime (Liu et al. 2020; Lu et al. 2021) to follow. To fairly compare our method with baselines, we create standard data splits with different-sized training set and run all models based on these standard splits. We split the training data based on documents, which is a more realistic setup compared to splitting data by instance. Appendix A lists detailed statistics.

Evaluation. We consider the same criteria in previous work on EE (Wadden et al. 2019; Lin et al. 2020) and report F1-scores in our experiments.

- **Trigger F1**: an trigger is correctly identified (Tri-I) if its offset matches the gold trigger’s; and it is correctly classified (Tri-C) if its event type also matches the gold one.

We release splits that contain {1%, 2%, 3%, 5%, 10%, 20%, 30%, 50%, and 100%} of training data.
### Table 1: End-to-end event extraction results in low-data regime. The reported number is argument classification F1-scores (%). Highest scores are in bold and the second best scores are underlined.

| Model                  | ACE05-E | ACE05-E⁺ | ERE-EN |
|------------------------|---------|----------|--------|
| OneIE                  | 9.4     | 26.8     | 26.8   |
| BERT_QA                | 4.7     | 26.9     | 27.6   |
| TANL                   | 8.5     | 24.7     | 29.0   |
| Text2Event             | 3.9     | 19.1     | 24.9   |
| Ours-DEGREE (ED+EAE)   | 13.1    | 27.4     | 41.6   |
| Ours-DEGREE            | 21.7    | 38.6     | 41.6   |

Figure 3: The comparison between models for data-scarce event extraction tasks: The upper three figures demonstrate models’ performance on event detection, and the lower three figures show the end-to-end argument classification results. We report detailed values for the end-to-end argument classification F1-scores in Table 1.

- **Argument F1**: an argument is correctly identified (Arg-I) if its offset and event type matches the gold argument’s; and it is correctly classified (Arg-C) if its role label also matches the gold argument’s.

#### 4.2 Compared Baselines

We consider prior works on EE and list our implementation details for all these baselines and our models in Appendix B.

- **OneIE** *(Lin et al. 2020)*, the current state-of-the-art on EE, is a classification-based IE system trained with global features and multi-tasking. The training of OneIE requires not only event annotations but also entity and relation labels, leading to its label-eager nature.

- **BERT_QA** *(Du and Cardie 2020)*, which views EE tasks as a sequence of extractive question answering problems. It takes multi-turns of separated QAs to extract triggers and arguments. It is worthy noting that BERT_QA assumes a trigger being a single token, hence this method is not applicable for ACE05-E⁺ and ERE-EN.

- **TANL** *(Paolini et al. 2021)*, which treats EE tasks as translation tasks between augmented natural languages. It lacks the design of prompting event structures to the model, hence, needs to discover valid patterns for each event types via learning from data.

- **Text2Event** *(Lu et al. 2021)*, a sequence-to-structure generation paradigm that is claimed to be data-efficient. Text2Event, which also lacks the design of prompting, overcomes the missing event structure issue via constrained decoding. However, their unnatural output formats hinders the exploitation of pretrained knowledge.

#### 4.3 Experimental Results

We illustrate the comparison between our methods and the baselines in Figure 3, where the evaluations on event detection are shown on the upper figure and the results of end-to-end EE evaluation (measured in argument classification F1-score) are demonstrated on the lower figure. From the figure, we can observe that both our DEGREE(ED+EAE) and DE-
GREE significantly outperform all baselines under extremely low data situation, i.e., whenever only less than 20% of training documents are available.

Table 1 shows the corresponding scores for the lower figure in Figure 3. We can observe that OneIE becomes more competitive when over 30% of training data is exposed. However, OneIE can benefit from more label resources since it requires not only event labels but also entity and relation annotations, leading to a slightly biased comparison.

By capturing output dependencies, our model obtain a better performance than BERT_QA. The importance of output dependencies is also shown when comparing DEGREE(ED+EAE) to DEGREE. DEGREE directly extracts events in an end-to-end manner, explicitly capturing the dependencies from arguments to triggers.

Taking advantages of importing event structures to the model for generation, our method outperforms TANL by a large margin. Text2Event perform poorly under extremely low data situation because of the complex output sequence design, which prevents the model from transferring the pre-trained knowledge in generative model to its generation process. Conversely, our model summarize the input passage into a natural language, facilitating the exploitation of the pre-trained knowledge.

5 High-Resource Event Extraction

While we are focusing on data-efficient learning for low-resource EE, in this section, we want to understand our proposed approach’s effectiveness when more data is available. This is to verify whether the three aspects of knowledge DEGREE encoded is generally helpful for the EE tasks.

Similar to the experiments reported in Section 4, we consider ACE-E, ACE05-E+, and ERE-EN datasets. Then, we train all baselines and our models on the full training set.

We compare our method to the baselines that we have mentioned in Section 4 and three additional baselines. Specifically, we compare with: (1) OneIE [Lin et al. 2020], (2) BERT_QA [Du and Cardie 2020], (3) TANL [Pasolin et al. 2021], (4) Text2Event [Lu et al. 2021], (5) DyGIE++ [Wadden et al. 2019], which is a classification-based model with span graph propagation technique (6) MQAEE [Li et al. 2020], which performs EE in a pipelined question answering fashion, including first asking for trigger identification, then, querying the corresponding trigger type, and finally, a series of argument extraction QAs, and (7) BART-Gen [Li, Ji, and Han 2021], which is a conditional generative model working on event argument extraction.

To better understand our models, we studies two evaluation settings—(1) event argument extraction, where the gold triggers are already provided during test time, and (2) end-to-end event extraction, where the models need to predict triggers and its corresponding arguments.

5.1 Results for Event Argument Extraction

In the discussion here, models are expected to predict their event arguments and the corresponding roles for the given gold triggers and their event types. Table 2 shows Arg-I and Arg-C for the three datasets. The results demonstrate that DEGREE(EAE) achieves improvements over the state-of-the-art model (OneIE) on all three datasets by an absolute F1 of 4.2%, 2.4%, and 6.3% when measuring on Arg-C. We even conduct additional experiments that equip OneIE with BART as its pretrained language model, which carry slightly improvements upon OneIE. However, OneIE (w/ BART) still fall behind DEGREE(EAE) for a large margin. By leveraging event structure and the output dependencies among arguments, DEGREE(EAE) show its effectiveness on event argument extraction task in high-data regime.

5.2 Results for Event Extraction

Next, we study our methods’ performances on high-resource end-to-end EE. As demonstrated in Table 3, both our DEGREE(ED+EAE) and DEGREE are superior to most baselines (DyGIE++, BERT_QA, MQAEE, TANL, Text2Event, BART-Gen). We observe that the improvements of our methods on event detection tasks (Tri-I and Tri-C) over baselines are less significant compared to the improvements on event extraction tasks. The relatively poor performance on trigger prediction is also seen in other generative-based models (TANL, Text2Event). We hypothesize the reason being that the output dependencies between triggers are weaker than the dependencies between arguments. Therefore, when trained with enough data, generative models which capture output dependencies are less advantageous over other models. Despite the slightly worse event detection results, both of our methods still show competitive results compared to the current state-of-the-art model, OneIE. Our models outperform OneIE and OneIE (w/ BART) on ERE-EN dataset, demonstrating our models’ capabilities.

6 Ablation Study

In this section, we conduct comprehensive studies to justify the design of our method. Specifically, we consider DEGREE(EAE) and DEGREE(ED), conduct ablation studies under both data-scarce and high-resource situation, and study on the input prompt choices and the template design.

Effects of different elements in prompts. We ablate each element in the the prompt of DEGREE(ED) and DEGREE(EAE) and report the scores in Table 4 and 5.

In Table 4, we can observe that ablating query trigger drops the model performance of DEGREE(EAE) the most.

| Model                | ACE05-E Arg-I | ACE05-E Arg-C | ERE-EN Arg-I | ERE-EN Arg-C |
|----------------------|---------------|---------------|--------------|--------------|
| DyGIE++              | 66.2          | 60.7          | -            | -            |
| BERT_QA*             | 68.2          | 65.4          | -            | -            |
| BART-Gen*            | 69.9          | 66.7          | -            | -            |
| OneIE                | 73.2          | 69.3          | 73.3         | 70.6         |
| OneIE (w/ BART)      | 74.2          | 70.3          | 73.5         | 70.8         |
| DEGREE(EAE)          | 76.0          | 73.5          | 75.2         | 73.0         |

Table 2: Results for event argument extraction. Models predict arguments for the given gold triggers. Best scores are in bold. *We report the numbers from the original paper.
| Model                  | ACE05-E    | ACE05-E* | ERE-EN |
|-----------------------|------------|----------|--------|
|                       | Trig-I     | Trig-C   | Arg-I  | Arg-C  | Trig-I     | Trig-C   | Arg-I  | Arg-C  | Trig-I     | Trig-C   | Arg-I  | Arg-C  |
| DyGIE++               | 74.3       | 70.0     | 53.2   | 50.0   | -         | -        | -      | -      | -         | -        | -      | -      |
| BERT-QA*              | 75.8       | 72.4     | 55.3   | 53.3   | -         | -        | -      | -      | -         | -        | -      | -      |
| MQAEE*                | 74.5       | 71.7     | 55.2   | 53.4   | -         | -        | -      | -      | -         | -        | -      | -      |
| TANL                  | 72.9       | 68.4     | 50.1   | 47.6   | 71.5      | 68.4     | 48.9   | 45.3   | 63.0      | 54.7     | 46.6   | 43.2   |
| Text2Event*           | -          | 69.2     | -      | 49.8   | -         | 71.8     | -      | 54.4   | -         | 59.4     | -      | 48.3   |
| BART-Gen*             | 74.4       | 71.1     | 55.2   | 53.7   | -         | -        | -      | -      | -         | -        | -      | -      |
| OneIE*                | 78.2       | 74.7     | 59.2   | 56.8   | 75.6      | 72.8     | 57.3   | 54.8   | 68.4      | 57.0     | 50.1   | 46.5   |
| OneIE (w/ BART)       | 75.1       | 71.3     | 58.1   | 55.5   | 76.6      | 72.6     | 60.3   | 58.3   | 67.3      | 58.6     | 52.2   | 48.8   |
| DEGREE(ED+EAE) (ours) | 75.7       | 72.2     | 57.6   | 56.0   | 74.3      | 71.7     | 59.6   | 58.0   | 63.6      | 56.6     | 53.5   | 51.1   |
| DEGREE (ours)         | 74.7       | 70.9     | 56.5   | 54.4   | 76.7      | 72.7     | 57.1   | 55.0   | 65.8      | 57.1     | 52.0   | 49.6   |

Table 3: Results for end-to-end event extraction evaluation with full training data. Highest scores are in bold and the second best scores are underlined. *We report the numbers from the original paper.

| Input Format | 10% Data | 100% data |
|--------------|----------|-----------|
|              | Arg-I | Arg-C | Arg-I | Arg-C | Arg-I | Arg-C | Arg-I | Arg-C |
| Full DEGREE(EAE) | 63.3  | 57.3  | 76.0  | 73.5  |
| - w/o Event type definition | 60.3  | 54.4  | 74.5  | 71.1  |
| - w/o EAE template | 57.0  | 51.9  | 73.8  | 70.4  |
| - w/o Query trigger | 55.2  | 49.9  | 71.4  | 69.0  |

Table 4: Input ablation study for event argument extraction with gold triggers provided on ACE05-E.

| Input Format | 10% Data | 100% data |
|--------------|----------|-----------|
|              | Trig-I | Trig-C | Tri-I | Tri-C | Trig-I | Trig-C | Tri-I | Tri-C |
| Full DEGREE(ED) | 72.0  | 70.0  | 75.7  | 72.2  |
| - w/o Event keywords | 69.8  | 67.6  | 73.5  | 69.1  |
| - w/o ED template | 69.8  | 67.5  | 74.0  | 70.5  |
| - w/o Event type definition | 68.8  | 65.4  | 73.5  | 70.1  |

Table 5: Input ablation study for event detection on the ACE05-E.

Triggers help the model identify the given event and provide potential information about the event type. The removal of EAE template also hurts the performance a lot, especially when replacing different role tokens or tags with an unified one. This confirms that label semantics is important.

### Template Design

From the ablation studies on the input prompt, we have verified the effect of providing template to the model. Then, we are now curious about the effectiveness about our template design. Specifically, we consider three variants of EAE templates:

1. **Natural sentence**: our proposed templates described in Section 3.2, e.g., "*somebody was born in somewhere*.", where "somebody" and "somewhere" are placeholders that can be replaced whenever there are corresponding arguments.

2. **Sentence w/different role tokens**: natural sentence templates with placeholders replaced by role-specific special tokens, e.g., "*<Person> was born in <Place>.*"

3. **Sentence w/an unified role token**: natural sentence templates with placeholders replaced by a unified special token, e.g., "*<Role> was born in <Role>.*"

Notice that since pretrained language models have never seen those special tokens, the two latter templates are considered as partially natural sentences.

The results of all variants of EAE templates on ACE05-E are shown in Table 6. We notice that writing templates in a natural language style get better performance, especially when only low data is available. This shows our design’s capability to leverage pretrained knowledge in the generation process. Additionally, there are large performance drops when replacing different role tokens or tags with an unified one. This confirms that label semantics is important.

### 7 Conclusion

In this paper, we formulate event extraction as a natural language generation task. We propose a framework that encodes the given information into carefully designed prompts and generates a natural sentence, from where the final triggers and arguments can be decoded. It successfully models event structures and exploits label semantics, resulting in promising performance on event extraction tasks under both data scarce and high-resource scenarios. In the future, we plan to extend the framework to a more general paradigm to address more structured prediction problems.
References

Ahn, D. 2006. The stages of event extraction. In Proceedings of the Workshop on Annotating and Reasoning about Time and Events.

Berant, J.; Srikumar, V.; Chen, P.; Linden, A. V.; Harding, B.; Huang, B.; Clark, P.; and Manning, C. D. 2014. Modeling Biological Processes for Reading Comprehension. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).

Cong, X.; Cui, S.; Yu, B.; Liu, T.; Wang, Y.; and Wang, B. 2021. Few-Shot Event Detection with Prototypical Amortized Conditional Random Field. In Findings of the Association for Computational Linguistics: ACL/IJCNLP.

Deng, S.; Zhang, N.; Kang, J.; Zhang, Y.; Zhang, W.; and Chen, H. 2020. Meta-Learning with Dynamic-Memory-Based Prototypical Network for Few-Shot Event Detection. In The Thirteenth ACM International Conference on Web Search and Data Mining (WSDM).

Doddington, G. R.; Mitchell, A.; Przybocki, M. A.; Ramshaw, L. A.; Strassel, S. M.; and Weischedel, R. M. 2004. The Automatic Content Extraction (ACE) Program - Tasks, Data, and Evaluation. In Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC).

Du, X.; and Cardie, C. 2020. Event Extraction by Answering (Almost) Natural Questions. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP).

Hogenboom, F.; Frasincar, F.; Kaymak, U.; de Jong, F.; and Caron, E. 2016. A Survey of event extraction methods from text for decision support systems. Decis. Support Syst., 85: 12–22.

Ji, H.; and Grishman, R. 2008. Refining Event Extraction through Cross-Document Inference. In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL).

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 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 (ACL).

Li, F.; Peng, W.; Chen, Y.; Wang, Q.; Pan, L.; Lyu, Y.; and Zhu, Y. 2020. Event Extraction as Multi-turn Question Answering. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings (EMNLP).

Li, Q.; Ji, H.; and Huang, L. 2013. Joint Event Extraction via Structured Prediction with Global Features. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL).

Li, S.; Ji, H.; and Han, J. 2021. 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 (NAACL-HLT).

Lin, Y.; Ji, H.; Huang, F.; and Wu, L. 2020. A Joint Neural Model for Information Extraction with Global Features. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL).

Liu, J.; Chen, Y.; Liu, K.; Bi, W.; and Liu, X. 2020. Event Extraction as Machine Reading Comprehension. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP).

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019.

Lu, Y.; Lin, H.; Xu, J.; Han, X.; Tang, J.; Li, A.; Sun, L.; Liao, M.; and Chen, S. 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 (ACL/IJCNLP).

Nguyen, T. H.; Cho, K.; and Grishman, R. 2016. Joint Event Extraction via Recurrent Neural Networks. In The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT).

Nguyen, T. H.; and Grishman, R. 2015. Event Detection and Domain Adaptation with Convolutional Neural Networks. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL).

Paolini, G.; Athiwaratkun, B.; Krone, J.; Ma, J.; Achille, A.; Anubhai, R.; dos Santos, C. N.; Xiang, B.; and Soatto, S. 2021. Structured Prediction as Translation between Augmented Natural Languages. In 9th International Conference on Learning Representations (ICLR).

Shen, S.; Wu, T.; Qi, G.; Li, Y.; Haffari, G.; and Bi, S. 2021. Adaptive Knowledge-Enhanced Bayesian Meta-Learning for Few-shot Event Detection. In Findings of the Association for Computational Linguistics: ACL/IJCNLP.

Song, Z.; Bies, A.; Strassel, S. M.; Riese, T.; Mott, J.; Ellis, J.; Wright, J.; Kulick, S.; Ryant, N.; and Ma, X. 2015. From Light to Rich ERE: Annotation of Entities, Relations, and Events. In Proceedings of the The 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation, (EVENTS@HLP-NAACL).

Wadden, D.; Wennberg, U.; Luan, Y.; and Hajishirzi, H. 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).

Wang, X.; Wang, Z.; Han, X.; Liu, Z.; Li, J.; Li, P.; Sun, M.; Zhou, J.; and Ren, X. 2019. HMEAE: Hierarchical Modular Event Argument Extraction. 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).

Yang, B.; and Mitchell, T. M. 2016. Joint Extraction of Events and Entities within a Document Context. In The
Yang, S.; Feng, D.; Qiao, L.; Kan, Z.; and Li, D. 2019. Exploring Pre-trained Language Models for Event Extraction and Generation. In Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL).

Zhang, H.; Liu, X.; Pan, H.; Song, Y.; and Leung, C. W. 2020. ASER: A Large-scale Eventuality Knowledge Graph. In The Web Conference 2020 (WWW).
A Dataset Statistics

Table 7 lists the statistics of ACE05-E, ACE05-E+, and ERE-EN.

B Implementation Details

The implementation details of all baselines and are as follows:

- **DyGIE++**: we use their released pretrained model for evaluation.
- **OneIE**: we use their provided code to train the model with default parameters.
- **BERT_QA**: we use their provided code to train the model.
- **TANL**: we use their provided code to train the model. We conduct the experiments with two variations: (1) using their default parameters, and (2) using their default parameters but with more training epochs. We observe that the second variant works better. As a result, we report the number obtained from the second setting.
- **Text2Event**: we use their official code to train the model with the provided parameter setting.
- **MQAEE**: we directly report the experimental results from their paper.
- **BART-Gen**: we report the experimental results from their released appendix.

For all of DEGREE(ED), DEGREE(EAE), and DEGREE, we finetune the pretrained BART-large (Lewis et al. 2020). We consider adamw optimizer (Loshchilov and Hutter 2019) with learning rate set to $10^{-5}$ and the weight decay is set to $10^{-5}$. The number of training epochs is 45. We set the batch size to 6 for DEGREE(EAE) and 32 for DEGREE(ED) and DEGREE. We set the number of negative examples $m$ to 15.

C Exemplification for Templates

Table 8 demonstrates two examples of input prompts and output texts we designed for our method. To interpret the table, the key is to first focus on the “Argument Roles” row, which represents the valid argument roles for the given event type. Based on the valid roles, we create the EAE template, which becomes the targeted structure that the output text needs to follow.

D List of EAE Templates

Table 9 and 10 lists all EAE templates for ACE05-E, ACE05-E+, and ERE-EN.

---

6 https://github.com/dwadden/dygiepp
7 http://blender.cs.illinois.edu/software/oneie/
8 https://github.com/xinyadu/eeqa
9 https://github.com/amazon-research/tanl
10 https://github.com/luyaojie/Text2Event
11 https://github.com/raspberryice/gen-arg/blob/main/NAACL_2021_Appendix.pdf
| Dataset | Split | #Docs | #Sents | #Events | #Event Types | #Args | #Arg Types |
|---------|-------|-------|--------|---------|--------------|-------|------------|
| ACE05-E | Train (full) | 529 17172 4202 | 33 | 4859 | 22 |
|         | Train (1%) | 5 103 47 | 14 | 65 | 16 |
|         | Train (2%) | 10 250 77 | 17 | 104 | 16 |
|         | Train (3%) | 15 451 119 | 23 | 153 | 17 |
|         | Train (5%) | 25 649 212 | 27 | 228 | 21 |
|         | Train (10%) | 50 1688 412 | 28 | 461 | 21 |
|         | Train (20%) | 110 3467 823 | 33 | 936 | 22 |
|         | Train (30%) | 160 5429 1368 | 33 | 1621 | 22 |
|         | Train (50%) | 260 8985 2114 | 33 | 2426 | 22 |
|         | Dev | 28 923 450 | 21 | 605 | 22 |
|         | Test | 40 832 403 | 31 | 576 | 20 |
|         | Train (full) | 529 19216 4419 | 33 | 6607 | 22 |
|         | Train (1%) | 5 92 49 | 15 | 75 | 16 |
|         | Train (2%) | 10 243 82 | 19 | 129 | 16 |
|         | Train (3%) | 15 434 124 | 24 | 203 | 19 |
|         | Train (5%) | 25 628 219 | 27 | 297 | 21 |
|         | Train (10%) | 50 1915 428 | 29 | 629 | 21 |
|         | Train (20%) | 110 3834 878 | 33 | 1284 | 22 |
|         | Train (30%) | 160 6159 1445 | 33 | 2212 | 22 |
|         | Train (50%) | 260 10104 2231 | 33 | 3293 | 22 |
|         | Dev | 28 901 468 | 22 | 739 | 22 |
|         | Test | 40 676 424 | 31 | 689 | 21 |
| ERE-EN  | Train (full) | 396 14736 6208 | 38 | 8924 | 21 |
|         | Train (1%) | 4 109 61 | 14 | 78 | 16 |
|         | Train (2%) | 8 228 128 | 21 | 183 | 19 |
|         | Train (3%) | 12 419 179 | 26 | 272 | 19 |
|         | Train (5%) | 20 701 437 | 31 | 640 | 21 |
|         | Train (10%) | 40 1536 618 | 37 | 908 | 21 |
|         | Train (20%) | 80 2848 1231 | 38 | 1656 | 21 |
|         | Train (30%) | 120 4382 1843 | 38 | 2632 | 21 |
|         | Train (50%) | 200 7690 3138 | 38 | 4441 | 21 |
|         | Dev | 31 1209 525 | 34 | 730 | 21 |
|         | Test | 31 1163 551 | 33 | 822 | 21 |

Table 7: Dataset statistics. Our experiments are conducted in sentences, which were split from documents. In the table, “#Docs” means the number of documents; “#Sents” means the number of sentences, “#Events” means the number of events; “#Event Types” means the number of event types in total; “#Args” means the number of argument in total; “#Arg Types” means the number of argument role types in total.

| Conflict: Demonstrate | Justice: Sue |
|-----------------------|-------------|
| Event Type Description | The event is related to a large number of people coming together to protest. | The event is related to a court proceeding has been initiated and someone sue the other. |
| Event Keywords | rally; protest; demonstration | sue; lawsuit; suit |
| ED Template | Event trigger is <Trigger>. | Event trigger is <Trigger>. |
| Output Text for ED | Event trigger is demonstrate. | Event trigger is lawsuits. |
| Argument Roles (Role: Argument) | Entity: Iraqis; Place: fallujah | Defendant: gunmakers, dealers; Plaintiff: victims, families; Place: none; Adjudicator: none |
| EAE Template | some people or some organization protest at somewhere | somebody was sued by some other in somewhere. The adjudication was judged by some adjudicator. |
| Output Text for DEGREE(EAE) | Iraqis protest at fallujah. | gunmakers and dealers was sued by victims and families in somewhere. The adjudication was judged by some adjudicator. |

Table 8: Examples of input prompts and output texts for our model. The events in the table follow the guidelines from the ACE 2005 dataset. Underlined parts of the templates represent placeholders, which can be replaced with triggers or arguments.
| Event Type               | EAE Template                                                                 |
|-------------------------|-----------------------------------------------------------------------------|
| Life:Be-Born            | somebody was born in somewhere.                                             |
| Life:Marry              | somebody got married in somewhere.                                          |
| Life:Divorce            | somebody divorced in somewhere.                                             |
| Life:Injure             | somebody or some organization led to some victim injured by some way in somewhere. |
| Life:Die                | somebody or some organization led to some victim died by some way in somewhere. |
| Movement:Transport      | something was sent to somewhere from some place by some vehicle. somebody or some organization was responsible for the transport. |
| Transaction:Transfer-Ownership | someone got something from some seller in somewhere.                          |
| Transaction:Transfer-Money | someone paid some other in somewhere.                                        |
| Business:Start-Org      | somebody or some organization launched some organization in somewhere.       |
| Business:Merge-Org      | some organization was merged.                                                |
| Business:Declare-Bankruptcy | some organization declared bankruptcy.                                     |
| Business:End-Org        | some organization dissolved.                                                |
| Conflict:Attack         | some attacker attacked some facility, someone, or some organization by some way in somewhere. |
| Conflict:Demonstrate    | some people or some organization protest at somewhere.                       |
| Contact:Meet            | some people or some organization met at somewhere.                          |
| Contact:Phone-Write     | some people or some organization called or texted messages at somewhere.     |
| Personnel:Start-Position | somebody got new job and was hired by some people or some organization in somewhere. |
| Personnel:End-Position  | somebody stopped working for some people or some organization at somewhere.  |
| Personnel:Nominate      | somebody was nominated by somebody or some organization to do a job.         |
| Personnel:Elect         | somebody was elected a position, and the election was voted by some people or some organization in somewhere. |
| Justice:Arrest-Jail     | somebody was sent to jailed or arrested by somebody or some organization in somewhere. |
| Justice:Release-Parole  | somebody was released by some people or some organization from somewhere.    |
| Justice:Charge-Indict   | somebody was charged by some other in somewhere. The adjudication was judged by some adjudicator. |
| Justice:Sue             | somebody was sued by some other in somewhere. The adjudication was judged by some adjudicator. |
| Justice:Convict         | somebody was convicted of a crime in somewhere. The adjudication was judged by some adjudicator. |
| Justice:Sentence        | somebody was sentenced to punishment in somewhere. The adjudication was judged by some adjudicator. |
| Justice:Fine            | some people or some organization in somewhere was ordered by some adjudicator to pay a fine. |
| Justice:Execute         | somebody was executed by somebody or some organization at somewhere.         |
| Justice:Extradite       | somebody was extradicted to somewhere from some place. somebody or some organization was responsible for the extradition. |
| Justice:Acquit          | somebody was acquitted of the charges by some adjudicator.                  |
| Justice:Pardon          | somebody received a pardon from some adjudicator.                           |
| Justice:Appeal          | some other in somewhere appealed the adjudication from some adjudicator.     |

Table 9: All EAE templates for ACE05-E and ACE05-E+.
| Event Type                              | EAE Template                                                                 |
|----------------------------------------|-----------------------------------------------------------------------------|
| Life:Be-Born                           | somebody was born in somewhere.                                             |
| Life:Marry                             | somebody got married in somewhere.                                          |
| Life:Divorce                           | somebody divorced in somewhere.                                             |
| Life:Injure                            | somebody or some organization led to some victim injured by some way in somewhere. |
| Life:Die                               | somebody or some organization led to some victim died by some way in somewhere. |
| Movement:Transport-Person              | somebody was moved to somewhere from some place by some way. somebody or some organization was responsible for the movement. |
| Movement:Transport-Artifact            | something was sent to somewhere from some place. somebody or some organization was responsible for the transport. |
| Business:Start-Org                    | somebody or some organization launched some organization in somewhere.      |
| Business:Merge-Org                     | some organization was merged.                                               |
| Business:Declare-Bankruptcy           | some organization declared bankruptcy.                                      |
| Business:End-Org                      | some organization dissolved.                                                |
| Conflict:Attack                        | some attacker attacked some facility, someone, or some organization by some way in somewhere. |
| Conflict:Demonstrate                  | some people or some organization protest at somewhere.                      |
| Conflict:Meet                          | some people or some organization met at somewhere.                          |
| Conflict:Correspondence               | some people or some organization contacted each other at somewhere.         |
| Conflict:Broadcast                    | some people or some organization made announcement to some publicity at somewhere. |
| Conflict:Contact                      | some people or some organization talked to each other at somewhere.         |
| Manufacture:Artifact                  | something was built by somebody or some organization in somewhere.          |
| Personnel:Start-Position               | somebody got new job and was hired by some people or some organization in somewhere. |
| Personnel:End-Position                 | somebody stopped working for some people or some organization at somewhere.  |
| Personnel:Nominate                    | somebody was nominated by somebody or some organization to do a job.        |
| Personnel:Elect                       | somebody was elected a position, and the election was voted by somebody or some organization in somewhere. |
| Transaction:Transfer-Ownership        | The ownership of something from someone was transferred to some other at somewhere. |
| Transaction:Transfer-Money            | someone paid some other in somewhere.                                      |
| Transaction:Transaction               | someone give some things to some other in somewhere.                       |
| Justice:Arrest-Jail                   | somebody was sent to jailed or arrested by somebody or some organization in somewhere. |
| Justice:Release-Parole                 | somebody was released by somebody or some organization from somewhere.      |
| Justice:Trial-Hearing                  | somebody, prosecuted by some other, faced a trial in somewhere. The hearing was judged by some adjudicator. |
| Justice:Charge-Indict                  | somebody was charged by some other in somewhere. The adjudication was judged by some adjudicator. |
| Justice:Sue                            | somebody was sued by some other in somewhere. The adjudication was judged by some adjudicator. |
| Justice:Convict                        | somebody was convicted of a crime in somewhere. The adjudication was judged by some adjudicator. |
| Justice:Sentence                      | somebody was sentenced to punishment in somewhere. The adjudication was judged by some adjudicator. |
| Justice:Fine                           | some people or some organization in somewhere was ordered by some adjudicator to pay a fine. |
| Justice:Execute                       | somebody was executed by somebody or some organization at somewhere.        |
| Justice:Extradite                      | somebody was extradicted to somewhere from some place. somebody or some organization was responsible for the extradition. |
| Justice:Acquit                        | somebody was acquitted of the charges by some adjudicator.                  |
| Justice:Pardon                        | somebody received a pardon from some adjudicator.                          |
| Justice:Appeal                         | somebody in somewhere appealed the adjudication from some adjudicator.      |

Table 10: All EAE templates for ERE-EN.
E  Few-Shot and Zero-Shot Event Extraction

In order to further test our models’ generaliability, in this section, we conduct zero-shot and few-shot experiments on the ACE05-E dataset using DEGREE(ED) and DEGREE(EAE). As we will show later, through the experiments, we demonstrate our models not only perform well on data-scarce EE and high-resource EE (in our main context) but also can generalize to few-shot and zero-shot setting due to the exploitation of label semantics.

Settings. For the experiments in this section, we first select the top $n$ common event types as “seen” types and use the rest as “unseen/rare” types, where the top common types are listed in Table 11. To simulate a zero-shot scenario, we remove all events with “unseen/rare” types from the training data. To simulate a few-shot scenario, we keep only $k$ event examples for each “unseen/rare” type (denoted as $k$-shot). During the evaluation, although inference on the full test set, we calculate micro F1-scores just for these “unseen/rare” types.

| n | Seen Event Types for Training/Development |
|---|------------------------------------------|
| 5 | Conflict:Attack, Movement:Transport, Life:Die, Contact:Meet, Personnel:Elect |
| 10 | Conflict:Attack, Movement:Transport, Life:Die, Contact:Meet, Personnel:Elect, Life:Injure, Personnel:End-Position, Justice:Trial-Hearing, Contact:Phone-Write, Transaction:Transfer-Money |

Table 11: Common event types in ACE05-E.

Compared baselines. We consider the following baselines: (1) BERT_QA [Du and Cardie 2020], which encodes label semantics into questions and is able to perform zero-shot event argument extraction. However, its trigger model is a sequential tagging model and the number of event types that it can predict is fixed after training, which prevents it from addressing zero-shot event detection. Thus, when evaluating event detection, BERT_QA can only be used in few-shot experiments. (2) OneIE [Lin et al. 2020], which is a classification model and it always makes a fixed-size categorical estimation. Hence, it cannot generalize to the zero-shot setting. As a result, we only test OneIE in the few-shot scenario. (3) Matching baseline, a proposed baseline that makes trigger predictions by performing string matching between the input passage and the event keywords. (4) Lemmatization baseline, another proposed baseline that performs string matching on lemmatized input passage and the event keywords. (Note: (3) and (4) are baselines only for event detection tasks.)

Experimental results. Figure 4 and Table 12 show the results of $n = 5$ and $n = 10$. From the two subfigures in the left column, we see that DEGREE(ED) achieves promising results in the zero-shot setting. In fact, it performs better than BERT_QA trained in the 10-shot setting and OneIE trained in the 5-shot setting. This demonstrates the great potential of DEGREE(ED) to discover new event types. Interestingly, we observe that our two proposed baselines perform surprisingly well, suggesting that the trigger annotations in ACE05-
Figure 4: The zero/few-shot experimental results. **Left:** The result for the models on event detection task with the scores reported in trigger classification F1. **Middle:** The models are tested under the scenario of given gold trigger and evaluated with argument classification criterion. **Right:** The results for the models to perform event extraction task, which aims to predict triggers and their corresponding arguments. We report the argument classification F1.
| Trigger               | Argument                     | Common 5          | Common 10         |
|-----------------------|------------------------------|-------------------|-------------------|
|                       |                              | Tri-I  Tri-C Arg-I Arg-C | Tri-I  Tri-C Arg-I Arg-C |
| Event Argument Extraction                          | 100.0 100.0 55.8 37.9 | 100.0 100.0 57.2 46.7 |
| Gold Triggers         | BERT_QA 0-shot               | 100.0 100.0 55.8 44.3 | 100.0 100.0 57.8 47.2 |
|                      | BERT_QA 1-shot               | 100.0 100.0 56.6 49.6 | 100.0 100.0 59.1 50.6 |
|                      | BERT_QA 5-shot               | 100.0 100.0 58.8 52.9 | 100.0 100.0 60.5 52.8 |
|                      | BERT_QA 10-shot              | 100.0 100.0 40.9 36.5 | 100.0 100.0 48.3 44.2 |
|                      | OneIE 1-shot                 | 100.0 100.0 55.6 51.4 | 100.0 100.0 58.6 55.0 |
|                      | OneIE 5-shot                 | 100.0 100.0 59.4 56.7 | 100.0 100.0 62.0 59.5 |
|                      | OneIE 10-shot                | 100.0 100.0 56.1 48.0 | 100.0 100.0 66.5 53.3 |
|                      | OneIE 10-shot                | 100.0 100.0 65.2 55.2 | 100.0 100.0 65.4 54.7 |
|                      | OneIE 5-shot                 | 100.0 100.0 70.9 62.2 | 100.0 100.0 68.0 61.7 |
|                      | OneIE 10-shot                | 100.0 100.0 71.1 64.2 | 100.0 100.0 71.6 64.3 |
|                      | OneIE (Full)                 | 100.0 100.0 63.1 57.9 | 100.0 100.0 62.1 56.5 |
|                      | OneIE (Full)                 | 100.0 100.0 70.8 66.4 | 100.0 100.0 67.9 64.1 |
|                      | OneIE (Full)                 | 100.0 100.0 74.5 70.6 | 100.0 100.0 73.6 68.9 |
| Event Extraction     |                              | 42.7 42.1 51.5 50.2 | 46.3 46.3 56.6 56.0 |
| Matching Baseline    |                              | 10.0 1.4 1.3 1.3 | 8.2 1.6 1.1 1.1 |
| Lemmatization Baseline |                           | 14.0 12.6 11.1 10.8 | 20.8 15.4 14.6 13.9 |
|                      | BERT_QA 1-shot               | 37.8 33.5 22.9 22.1 | 32.0 27.8 19.5 18.6 |
|                      | BERT_QA 10-shot              | 4.2 4.2 1.5 1.5 | 4.1 2.7 2.0 2.0 |
|                      | OneIE 1-shot                 | 39.3 38.5 24.8 22.8 | 41.9 41.9 29.7 27.2 |
|                      | OneIE 5-shot                 | 54.8 53.3 36.0 34.9 | 61.5 57.8 41.4 39.2 |
|                      | OneIE 10-shot                | 53.3 46.8 29.6 25.1 | 60.9 54.5 42.0 31.4 |
|                      | OneIE (Full)                 | 60.1 53.3 38.8 31.6 | 61.2 60.9 41.1 34.7 |
|                      | OneIE (Full)                 | 57.8 55.5 40.6 36.1 | 65.8 64.8 45.3 42.7 |
|                      | OneIE (Full)                 | 63.8 61.2 46.0 42.0 | 72.1 68.8 52.5 48.4 |
|                      | OneIE (Full)                 | 72.7 70.5 52.3 49.9 | 74.5 73.0 51.2 48.9 |
|                      | OneIE (Full)                 | 68.4 66.0 51.9 48.7 | 72.0 69.8 52.5 49.2 |

Table 12: Full results of zero/few-shot setting on ACE05-E. BERT_QA refers to the model from (Du and Cardie 2020), and OneIE is from (Lin et al. 2020).