Building an Event Extractor with Only a Few Examples

Pengfei Yu[1]*, **Zixuan Zhang**[1]*, **Clare Voss**[3], **Jonathan May**[2], **Heng Ji**[1]
[1]University of Illinois Urbana-Champaign [2]University of South California
[3]U.S. Army Combat Capabilities Development Command Army Research Laboratory
{pengfei4,zixuan11,hengji}@illinois.edu
jonmay@isi.edu, clare.r.voss.civ@army.mil

Abstract

Supervised event extraction models require a substantial amount of training data to perform well. However, event annotation requires a lot of human effort and costs much time, which limits the application of existing supervised approaches to new event types. In order to reduce manual labor and shorten the time to build an event extraction system for an arbitrary event ontology, we present a new framework to train such systems much more efficiently without large annotations. Our event trigger labeling model uses a weak supervision approach, which only requires a set of keywords, a small number of examples and an unlabeled corpus, on which our approach automatically collects weakly supervised annotations. Our argument role labeling component performs zero-shot learning, which only requires the names of the argument roles of new event types. The source codes of our event trigger detection[1] and event argument extraction[2] models are publicly available for research purposes. We also release a dockerized system connecting the two models into an unified event extraction pipeline[3].

1 Introduction

Supervised event extraction models require sufficient training data to achieve a good performance. However, event annotation is a challenging task costing a lot of time and manual effort due to the sparsity of event mentions in natural language and the potentially large number of emergent event types that human annotators need to keep in mind during annotation. Therefore, annotation becomes a bottleneck that slows down the development of supervised event extraction systems whenever a new scenario of interest emerges with new event types in need of new data.

In order to meet the needs of fast development of event extraction systems for emergent new event types, we present a novel framework that can train event extraction systems with very few resources. Our proposed framework includes a weakly supervised approach to train a event trigger labeling model and a zero-shot model for argument role labeling. Our proposed weakly supervised event trigger labeling model only requires a few keywords and a small number of example event mentions. In our experiments on the ACE 2005 English dataset,[4] we use 4.9 keywords and 7.3 example mentions per event type on average, which are all extracted from the ACE annotation guidelines. We also propose a zero-shot argument role labeling model that only requires the argument role names of new event types to perform the task. Since such information is typically included in the target ontology and annotation guidelines, we believe this required input costs much less than human annotations. Our framework can be applied to any new event types. Our trigger labeling component outperforms existing few-shot and zero-shot methods (Huang et al., 2018; Li et al., 2021; Feng et al., 2020) on ACE 2005 English dataset.

2 Approach

Our framework includes two components: a trigger labeling model trained from a few keywords and example mentions per each new event type and an unlabeled corpus; and a zero-shot argument role labeling model which only needs the corresponding argument role names for extraction.

2.1 Event Trigger Labeling

As shown in Figure 1, our framework requires a list of keywords $\{k_1, \ldots, k_M\}$ for each target

*These authors contributed equally to this work.

[1]https://github.com/Perfec-Yu/efficient-event-extraction

[2]https://github.com/zhangzx-uiuc/zero-shot-event-arguments

[3]https://hub.docker.com/repository/docker/zixuan11/event-extractor

[4]https://catalog.ldc.upenn.edu/LDC2006T06

event type and a set of event mentions as input. Our goal is to annotate an unlabeled corpus $\mathcal{C} = \{s_1, s_2, \ldots, s_N\}$, which is a collection of sentences s_i, and train a model on the weakly supervised annotations. The corpus for weak supervision is disjoint from the evaluation corpus.

Keyword Representation For each keyword k_i, we first find all its occurrences (including morphological inflection) in the corpus and summarize the semantics of each keyword into distributed representations by aggregating the hidden representation of each keyword occurrence using a large-scale language model $\mathcal{M}$ inspired by Meng et al. (2020). $\mathcal{M}$ functions as a sentence encoder to transform tokens in a sentence into hidden representations. A keyword occurrence consists of a sentence $s_j \in \mathcal{C}$ and a token offset (b_{ij}, e_{ij}) indicating the starting and ending offsets of k_i. We average the token hidden representations from the language model $\mathcal{M}$ within the token span as the representation for the j-th occurrence, and use the mean vector of all occurrences as the keyword representation $\boldsymbol{k}_i$. This process is shown in the top right corner of Figure 1.

Keyword Clustering and Annotation Since some keywords have similar meanings, we propose an additional clustering step to group similar keywords together to find mentions of novel trigger words not in the keyword list. We show an example in Figure 1 for the `Attack` event. We apply spherical KMeans (Lloyd, 1982) to acquire a set of cluster centers for an event type$\{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_m\}$. Letting $\boldsymbol{t}$ denote the representation of a token in an unlabeled sentence according to $\mathcal{M}$, we compute the score $S(t)$ of the token being an event trigger as the cosine similarity with the closest cluster representation for all the event type's clusters: $S(t) = \max_i \cos\_\text{sim}(\boldsymbol{c}_i, \boldsymbol{t})$. We accept a token as an event trigger of this type if the score $S(t)$ exceeds a threshold value. We select the threshold for which this annotation procedure achieves the best trigger labeling F1 score on example sentences.

Training with Example-based Denoising At each minibatch training step, let B_w be a sampled batch from the weakly supervised data. We further sample a batch B_e from the example mentions (from the human annotation guidelines). We compute the information consistency between B_w and B_e as $d = \mathbb{I}(\nabla_\theta \mathcal{L}_{B_e}^\top \nabla_\theta \mathcal{L}_{B_w} > 0)$ where $\mathbb{I}$ is the indicator function, $\mathcal{L}$ is the loss with respect to either the example batch or the weakly supervised batch, and θ is the set of model parameters. If $d = 0$, the training gradient has deviated far from the example gradient, in which case we discard the training data for loss computation. The overall loss is $\mathcal{L}_B = d\lambda\mathcal{L}_{B_w} + (1 - d\lambda)\mathcal{L}_{B_e}$, where λ is a hyperparameter that interpolates joint training on example data and weakly supervised data.

2.2 Event Argument Role Labeling

Our zero-shot event argument extraction model only requires the event argument role names (usually single words or phrases) for each event type (e.g., the event argument role names *Giver*, *Beneficiary*, *Recipient* and *Place* for event type *Transaction: Transfer-Money*). Note that our model does not require any detailed information such as natural language descriptions, example annotations or external resources (Zhang et al., 2021). Our model is trained on existing event argument roles with annotations, and is using zero-shot learning to generalize well to any new argument roles.

Zero-shot Training and Classification Inspired from many typical zero-shot learning tasks such as zero-shot image classification (Xian et al., 2018; Liu et al., 2018b), we take a similar approach to build a shared embedding space for both role label semantics and the contextual text features between triggers and arguments. Given an input sentence, we first perform named entity recognition (NER) with Spacy[5] to extract all entity mentions in a sentence. After that, given the event role names $\{r_1, r_2, \cdots, r_R\}$ for a certain event type, we first obtain the semantic embeddings $\{\boldsymbol{r}_1, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_R\}$ using the pretrained language model BERT (Devlin et al., 2019). We also use BERT to get the representation vectors for all extracted event triggers $\boldsymbol{t}_i$ and entity mentions $\boldsymbol{e}_i$ within the sentence, and concatenate the vectors as $[\boldsymbol{t}_i, \boldsymbol{e}_i]$ to represent a trigger-entity pair. The intuition here is to learn two separate neural network projection functions to map each role label and trigger-entity pair into a single shared embedding space, where each trigger-entity pair stays near its correct roles and far from all other event argument roles. During training, we minimize the cosine distance between each $[\boldsymbol{t}_i, \boldsymbol{e}_i]$ and its role label $\boldsymbol{r}_i$, while maximizing the distance between $[\boldsymbol{t}_i, \boldsymbol{e}_i]$ and all other role labels. Specifically, if we use $\mathcal{R}$ to represent the set of all argument role embeddings and use $\boldsymbol{x}_i = [\boldsymbol{t}_i, \boldsymbol{e}_i]$

[5] https://spacy.io/api/entityrecognizer

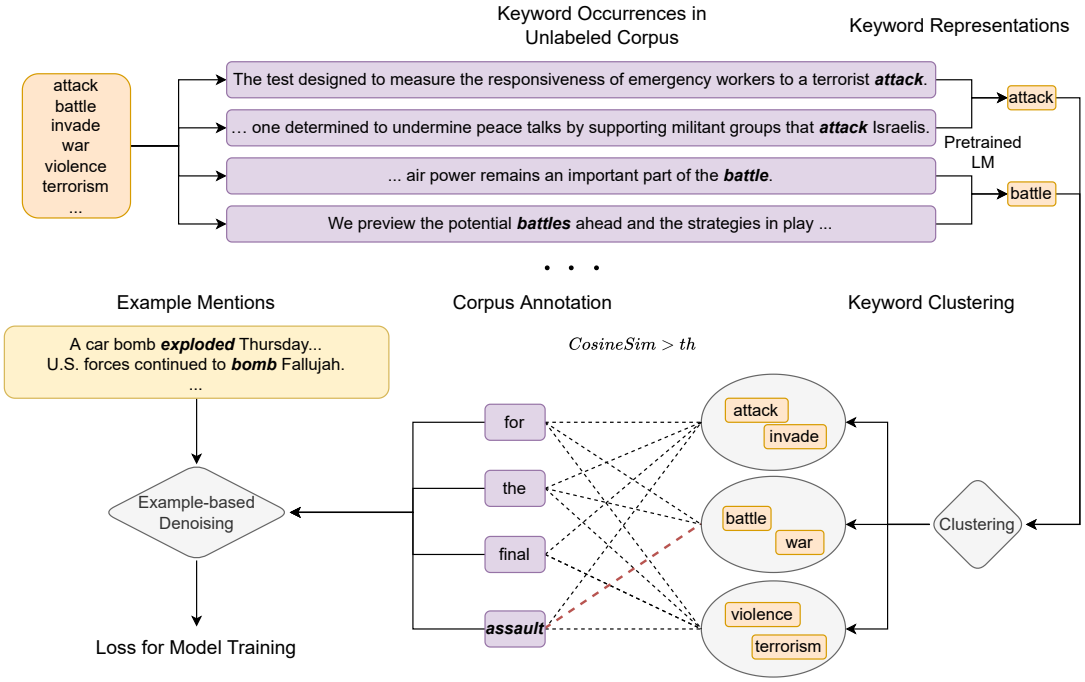

Figure 1: The weakly supervised event trigger labeling framework

to represent trigger-entity pairs, the training objective is to minimize the hinge loss $\mathcal{L}_i = \sum_{j \neq i, \boldsymbol{r}_j \in \mathcal{R}} \max\left(m - C(\boldsymbol{x}_i, \boldsymbol{r}_i) + C(\boldsymbol{x}_i, \boldsymbol{r}_j)\right)$, where $C(\boldsymbol{x}, \boldsymbol{r})$ denotes the cosine similarity. In this way, the trigger-entity pair representations tend to be centered around their argument role labels. During testing, we directly classify each trigger-entity pair as its nearest role label.

3 Evaluation

3.1 Dataset

We evaluate our models with the English portion of the ACE 2005 dataset. It contains 33 event types with 22 event argument role types. We use the training split as the weak supervision corpus, while in zero-shot event argument role labeling, we follow previous work (Huang et al., 2018; Zhang et al., 2021) and use the 10 most frequent event types as training types and other event types along with their role types for testing.

Dataset	Split	#Sents	#Ents	#Events
ACE05-E	Train	17,172	29,006	4,202
	Dev	923	2,451	450
	Test	832	3,017	403

Table 1: Dataset statistics.

3.2 Results

Event Detection. We evaluate event detection performance on two tasks. The first is the traditional trigger labeling. The model detects trigger spans from sentences and predicts an event type for each span. The second task is sentence level event detection (Feng et al., 2020), where the model predicts whether a sentence contains a mention of each event type. We evaluate both of the tasks with the F1 score. To further evaluate the impact of weak supervision, we compare with the **Example** baseline, which uses the same architecture but is trained only with example mentions in the human annotation guidelines. We also show ablation results for the keyword clustering step and example-based denoising step. As an efficient approach for event detection, we also compare with other zero-shot and few-shot methods for each task, as specified next below. We provide more implementation details in the Appendix.

We show the performance of our framework on trigger labeling in Table 2. We compare with the reported performance using two zero-shot methods: **ZSL** (Huang et al., 2018) and **TapKey** (Li et al., 2021). Our framework has the best performance among all the methods. We also show some inconsistent weakly supervised annotations ($d = 0$ in the denoising step in Section 2.1) from the denoising component in Table 3 to demonstrate the effectiveness of the denoising component. To further understand the effect of weak supervision, we compare the weakly supervised results with supervised models trained on varying percentages of training data

Full ACE Ontology (33 Types)	P	R	F
TapKey (Li et al., 2021)	-	-	52.1
Example	57.2	63.0	59.8
Ours	65.6	60.8	**63.1**
w/o denoising	62.2	61.1	61.6
w/o clustering	61.3	59.7	60.4
ACE Subset (23 Types)	P	R	F
ZSL (Huang et al., 2018)	75.5	36.3	49.1
Ours	66.3	60.5	**63.3**

Table 2: Trigger labeling performance (in %). Huang et al. (2018) evaluated on a 23-event-type subset of the complete ACE event ontology. We compute our model's performance on these types for comparison. The slots with "-" are unreported results.

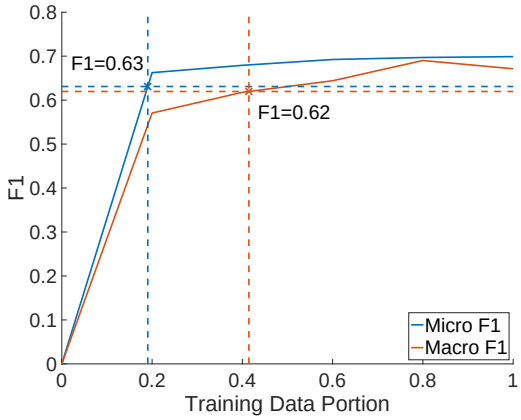

Figure 2: Supervised performance with respect to training data portion. Dotted lines indicate the performance of the weakly supervised methods.

in Figure 2. For sentence-level detection, we compare with the best few-shot (9-shot) results (Feng et al., 2020) in Table 4. Our weakly supervised approach has improved the performance.

Error	Inconsistent Weak Supervision
False Positive	... a minute fraction of the sum of **money**[Transfer-Money] ...
False Negative	... concerns our ability to *travel*[Transport].

Table 3: Inconsistent weakly supervised annotations from the denoising step.

Method	P	R	F
9-shot (Feng et al., 2020)	54.5	57.0	61.8
Example	66.4	68.0	66.9
Ours	66.2	74.2	**69.9**

Table 4: Sentence level event detection result (%).

Event Argument Extraction. In our experiments of event argument extraction, we use the top 10 frequent event types in ACE dataset for training and the other 23 types for testing. We report the precision, recall, and F1 scores on the test split of ACE dataset as shown in Table 5.

Models / Role Types	Prec	Rec	F1
Our Model	39.6	49.7	**41.5**
(Huang et al., 2018)	-	-	14.7
Start-Position:Entity	48.5	76.2	59.3
Justice:Defendant	55.0	44.0	48.9
Justice:Agent	45.5	45.5	45.5

Table 5: Event argument role labeling performance on ACE dataset. We report both overall scores and also top-3 scores on specific event argument roles.

4 Related Work

Supervised Event Detection Event detection under supervised settings has been widely studied (Ji and Grishman, 2008; Chen et al., 2015; Feng et al., 2016; Liu et al., 2017, 2018a, 2019a; Lu et al., 2019; Ding et al., 2019; Yan et al., 2019; Tong et al., 2020; Du and Cardie, 2020; Li et al., 2021). Other methods on joint information extraction (Li et al., 2013; Wadden et al., 2019; Lin et al., 2020) also include event detection as a subtask. However, supervised methods heavily rely on human annotations to perform well.

Weakly Supervised Event Extraction Some previous weakly supervised event extraction methods aim at augmenting data for existing event types. Ferguson et al. (2018) propose a semi-supervised method which requires a strong supervised event extractor for data collection. Chen et al. (2017) propose a distant supervision based framework using Freebase Compound Value Types (CVTs). Wang et al. (2019) follow Chen et al. (2015) and introduce a novel adversarial training method to denoise the noisy training data for event extraction.

Zero-shot Event Argument Extraction In zero-shot learning (Zhang and Saligrama, 2015; Romera-Paredes and Torr, 2015; Zhang et al., 2017), the model is required to make predictions on types that are not observed during training. Such a problem setting has also been widely explored in Computer Vision, especially for zero-shot image classification (Gu et al., 2021; Hanouti and Borgne, 2022). In terms of zero-shot event extraction, Huang et al. (2018) propose a semantic similarity based learning method, and more recently, Zhang et al. (2021) fur-

ther use resources from external corpus as weakly-supervised example annotations.

5 Conclusions

In this work we present an efficient event extraction framework that can be trained with only a few keywords and example event mentions per new event type. We use weak supervision for trigger labeling and apply a zero-shot framework for argument role labeling. Our framework can collect training data and build models for emergent new event types in a significantly shortened time without needing to acquire large-scale human annotations.

Acknowledgement

This research is based upon work supported in part by U.S. DARPA LORELEI Program No. HR0011- 15-C-0115, U.S. DARPA AIDA Program No. FA8750-18-2-0014 and KAIROS Program No. FA8750-19-2-1004. 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, 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

Yubo Chen, Shulin Liu, Xiang Zhang, Kang Liu, and Jun Zhao. 2017. Automatically labeled data generation for large scale event extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 409–419, Vancouver, Canada. Association for Computational Linguistics.

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 (Volume 1: Long Papers)*, pages 167–176, Beijing, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ning Ding, Ziran Li, Zhiyuan Liu, Haitao Zheng, and Zibo Lin. 2019. Event detection with trigger-aware lattice neural network. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 347–356, Hong Kong, China. Association for Computational Linguistics.

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683, Online. Association for Computational Linguistics.

Rui Feng, Jie Yuan, and Chao Zhang. 2020. Probing and fine-tuning reading comprehension models for few-shot event extraction. *arXiv preprint arXiv:2010.11325*.

Xiaocheng Feng, Lifu Huang, Duyu Tang, Heng Ji, Bing Qin, and Ting Liu. 2016. A language-independent neural network for event detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 66–71, Berlin, Germany. Association for Computational Linguistics.

James Ferguson, Colin Lockard, Daniel Weld, and Hannaneh Hajishirzi. 2018. Semi-supervised event extraction with paraphrase clusters. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 359–364, New Orleans, Louisiana. Association for Computational Linguistics.

Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. 2021. Zero-shot detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*.

Celina Hanouti and Hervé Le Borgne. 2022. Learning semantic ambiguities for zero-shot learning. *arXiv preprint arXiv:2201.01823*.

Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare Voss. 2018. Zero-shot transfer learning for event extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2160–2170, Melbourne, Australia. Association for Computational Linguistics.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of ACL-08: HLT*, pages 254–262, Columbus, Ohio. Association for Computational Linguistics.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 73–82, Sofia, Bulgaria. Association for Computational Linguistics.

Sha Li, Heng Ji, and Jiawei Han. 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, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. Association for Computational Linguistics.

Jian Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2019a. Neural cross-lingual event detection with minimal parallel resources. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 738–748, Hong Kong, China. Association for Computational Linguistics.

Shaobo Liu, Rui Cheng, Xiaoming Yu, and Xueqi Cheng. 2018a. Exploiting contextual information via dynamic memory network for event detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1030–1035, Brussels, Belgium. Association for Computational Linguistics.

Shichen Liu, Mingsheng Long, Jianmin Wang, and Michael I Jordan. 2018b. Generalized zero-shot learning with deep calibration network. In *Advances in Neural Information Processing Systems*, pages 2005–2015.

Shulin Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2017. Exploiting argument information to improve event detection via supervised attention mechanisms. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1789–1798, Vancouver, Canada. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Stuart P. Lloyd. 1982. Least squares quantization in PCM. *IEEE Trans. Inf. Theory*, 28(2):129–136.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Yaojie Lu, Hongyu Lin, Xianpei Han, and Le Sun. 2019. Distilling discrimination and generalization knowledge for event detection via delta-representation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4366–4376, Florence, Italy. Association for Computational Linguistics.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. 2020. Text classification using label names only: A language model self-training approach. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9006–9017, Online. Association for Computational Linguistics.

Bernardino Romera-Paredes and Philip Torr. 2015. An embarrassingly simple approach to zero-shot learning. In *International conference on machine learning*, pages 2152–2161. PMLR.

Meihan Tong, Bin Xu, Shuai Wang, Yixin Cao, Lei Hou, Juanzi Li, and Jun Xie. 2020. Improving event detection via open-domain trigger knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5887–5897, Online. Association for Computational Linguistics.

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-IJCNLP)*, pages 5784–5789, Hong Kong, China. Association for Computational Linguistics.

Xiaozhi Wang, Xu Han, Zhiyuan Liu, Maosong Sun, and Peng Li. 2019. Adversarial training for weakly supervised event detection. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 998–1008, Minneapolis, Minnesota. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. 2018. Feature generating networks for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5542–5551.

Haoran Yan, Xiaolong Jin, Xiangbin Meng, Jiafeng Guo, and Xueqi Cheng. 2019. Event detection with multi-order graph convolution and aggregated attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the*

9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 5766–5770, Hong Kong, China. Association for Computational Linguistics.

Hongming Zhang, Haoyu Wang, and Dan Roth. 2021. Zero-shot Label-aware Event Trigger and Argument Classification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1331–1340, Online. Association for Computational Linguistics.

Li Zhang, Tao Xiang, and Shaogang Gong. 2017. Learning a deep embedding model for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2021–2030.

Ziming Zhang and Venkatesh Saligrama. 2015. Zero-shot learning via semantic similarity embedding. In *Proceedings of the IEEE international conference on computer vision*, pages 4166–4174.

A Implementation Details

A.1 Spherical KMeans for Keyword Clustering

Compared with traditional KMeans (Lloyd, 1982), there are two modifications in spherical KMeans. Firstly, the cluster assignment at each iteration step is decided according to the cosine similarities to the cluster centers instead of the Euclidean distance. Besides, after computing the cluster centers as the mean vectors of those keyword representations that are assigned to the corresponding clusters, we add an additional normalizing step to make all cluster centers have unit norm. We use the implementation in `https://github.com/jasonlaska/spherecluster` for experiments.

A.2 Implementation Details for Trigger Labeling

We adopt a sequence labeling model for trigger labeling. Since we observe very few consecutive trigger spans, we use a simplified 'IO' tagging method instead of 'BIO' tagging. Specifically, we assign each token in a sentence a label 'I-<Event_type>' if it is in a trigger span of the corresponding event type. For the model architecture, we use Roberta-Large (Liu et al., 2019b) to encode each token in the sentences into a hidden representation. Then we adopt an additional linear layer to classify each token into one of the tags. We use training batch size of 8 sentences. We truncate sentences to contain at most 96 tokens. For optimization, we use AdamW (Loshchilov and Hutter, 2019) optimizer with initial learning rate 10^{-5}. We also use a linear warmup with 1200 warmup steps. We run experiments with 4 random seeds and report the average score.

A.3 Implementation Details for Sentence-level Event Detection

We use a Roberta-Large model finetuned on MultiNLI (Williams et al., 2018) dataset for textual entailment. The input to the model consists of a candidate sentence and an event-type-specific entailment sentence, such as *Agent attacked Target* for `Attack` event. The complete list of used entailment sentences can be found in the supplementary materials. The model outputs scores for the three labels: s_e for *entailment*, s_n *neutral* and s_c *contradiction*. We compute the probability of mentioning an event as $P(Mention) = \frac{e^{s_e}}{e^{s_e}+e^{s_n+s_c}}$. We use cross entropy loss to train the model. For evaluation, consider the candidate sentence mentioning an event if the probability of entailment is greater than 0.5. We use the same training hyper-parameters as trigger labeling. We run experiments with 4 random seeds and report the average score.

A.4 Implementation Details for Weak Supervision

For the weak annotation, The threshold is chosen from 0.4 to 1.0 with 0.05 incremental steps. We choose the threshold as 0.65 to have the best F1 score on the example mentions. Since we use the ACE 2005 English training corpus for weak supervision, we also compute the F1 score of the weakly supervised annotation directly. The F1 score is 0.46.

For the example-based denoising, we choose the weight parameter $\lambda = 0.7$ for trigger labeling and $\lambda = 0.5$ for sentence-level event detection.

B Keywords and Example Mentions

We show keywords for each event type in Table 6. We include example mentions in the supplementary materials. We have a total of 173 sentences and 241 event mentions in the example data.

Event Type	Keywords
Business:Declare-Bankruptcy	bankruptcy, broke, broken, bankrupt
Business:End-Org	failure, shut, collapse, fold
Business:Merge-Org	merger, merge
Business:Start-Org	initiate, establish, established, launch
Conflict:Attack	conflict, shoot, war, fighting, violence, attack, surge, battle, terrorism, invasion, coalition, warfare, explode, invade, pound, combat, fought
Conflict:Demonstrate	rally, protest, demonstration, demonstrate, riot
Contact:Meet	talk, meet, meeting, seminar, summit, dialogue
Contact:Phone-Write	call, phone, letter, email, video, cable, telephone, correspondence, mail, dial
Justice:Acquit	acquittal
Justice:Appeal	appeal
Justice:Arrest-Jail	jail, arrest, imprison
Justice:Charge-Indict	charge, accuse, indictment, accusation
Justice:Convict	convict
Justice:Execute	execute, execution
Justice:Extradite	deport, expel, extradite
Justice:Fine	penalty, fine, fee, penalize
Justice:Pardon	mercy, forgive, pardon
Justice:Release-Parole	parole, release, free
Justice:Sentence	sentence
Justice:Sue	sue, lawsuit
Justice:Trial-Hearing	trial, hearing, testify
Life:Be-Born	birth, born
Life:Die	die, death, suicide, murder, kill, slaughter, survive, killing, stabbed, fatal
Life:Divorce	divorce, split
Life:Injure	hurt, harm, hit, wound, injure, injured, wounded
Life:Marry	wedding, marry, wed
Movement:Transport	head, move, retreat, leave, visit, trip, travel, shift, tour, remove, return, arrive, carry, moving, ship, journey, transport, cruise, transition, deploy
Personnel:Elect	elect, election, vote, voting, poll, electoral, voter
Personnel:End-Position	resign, former, previous, fire, late, retire, dismiss, formerly, defunct
Personnel:Nominate	name, nominate
Personnel:Start-Position	appoint, employ, hire
Transaction:Transfer-Money	pay, spend, compensate, borrow, transfer, donate, lend
Transaction:Transfer-Ownership	buy, buying, acquire, purchase, acquisition, takeover, obtain

Table 6: Keywords used for each event type. Although we performed lemmatization for matching, there are some situations that lemmatization cannot handle perfectly. Therefore we also include various tenses for some verbs.