Zero-shot Event Detection using a Textual Entailment Model as an Enhanced Annotator

Ziqian Zeng, Runyu Wu, Yuxiang Xiao, Xiaoda Zhong, Hanlin Wang, Zhengdong Lu, Huiping Zhuang
South China University of Technology
{zqzeng, hpzhuang}@scut.edu.cn,
{rywu0911, yuxiangxiao02, xiaodaz0929, hlwang1024, 2910641515g}@gmail.com

Abstract

Zero-shot event detection is a challenging task. Recent research work proposed to use a pre-trained textual entailment (TE) model to solve this task. However, those methods treated the TE model as a frozen annotator. We treat the TE model as an annotator that can be enhanced. We propose to use a TE model to annotate large-scale unlabeled text and use annotated data to finetune the TE model, yielding an improved TE model. Finally, the improved TE model is used for inference on the test set. To improve the efficiency, we propose to use keywords to filter out sentences with a low probability of expressing event(s). To improve the coverage of keywords, we expand limited number of seed keywords using WordNet, so that we can use the TE model to annotate unlabeled text efficiently. The experimental results show that our method can outperform other baselines by 15% on the ACE05 dataset.

Keywords: Event Detection, Zero-shot Methods, Textual Entailment

1. Introduction

Event detection (ED) (Nguyen and Grishman, 2015; Chen et al., 2015; Nguyen et al., 2016a,b; Sha et al., 2018; Zhang et al., 2019; Yang et al., 2019; Nguyen and Nguyen, 2019; Zhang et al., 2020) is an important task in information extraction. ED methods are mostly accomplished in a supervised manner which requires a large number of annotated data. To mitigate the burden of extensive annotations, a more challenging paradigm named zero-shot event extraction (Huang et al., 2016) is proposed. In this zero-shot setting, annotated training data is not available.

Recently, Lyu et al. (2021) and Sainz et al. (2022) used a pre-trained Textual Entailment (TE) model to detect event types on the test set without using any annotated training data. They generated a description for each event, considered it a hypothesis, and treated the input text as a premise. A TE model (Bowman et al., 2015; Williams et al., 2018) is used to infer whether the premise entails the hypothesis. A high entailment score indicates that the text contains an event the hypothesis describes.

However, the aforementioned methods (Lyu et al., 2021; Sainz et al., 2022) treat the TE model as a frozen annotator which is used solely for inference on the test set. In this paper, we view the TE model as an annotator that can be enhanced. We propose to use TE models to annotate large-scale unlabeled text and then use the annotated data to finetune the TE model to enhance its performance. Finally, the enhanced TE model is used to infer event types in the test data. Specifically, we generated a hypothesis for each event type and treated the input sentence as the premise. We set a filter threshold to determine whether an input text expresses an event. In this way, the TE model can naturally identify sentences that express single, multiple, or no event mentions. After obtaining annotated data, we used it to finetune the TE model, resulting in an enhanced TE model that is subsequently used for inference on the test set. The difference between a frozen annotator and an enhanced annotator is shown in Figure 1.

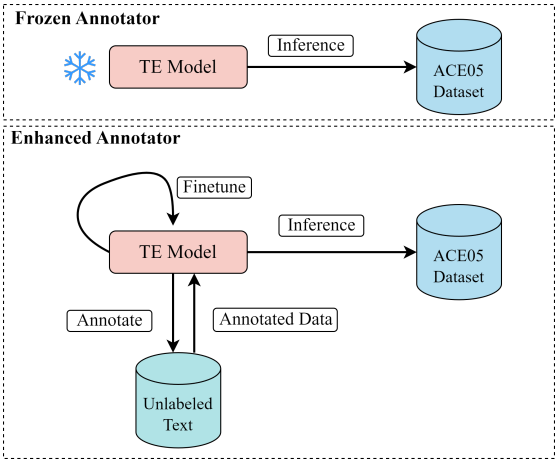

Figure 1: The illustration of the difference between a textual entailment model as a frozen annotator and an enhanced annotator.

Ideally, TE models can be used to annotate massive amounts of unlabeled text to detect events. However, this process is time-consuming. To mitigate this issue, we propose to use keywords to filter out the text that is unlikely to express an event. Human-provided keywords usually have high accuracy but low coverage. Inspired by Araki and Mitamura (2018); Tong et al. (2020), we use WordNet

LREC-COLING 2024, pages 17851–17857
20-25 May, 2024.

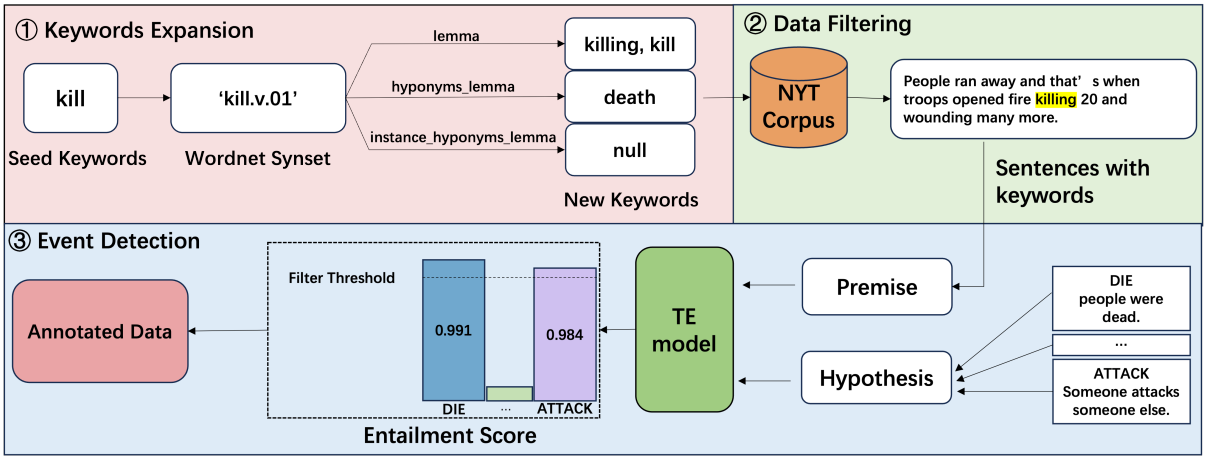

Figure 2: The general workflow of using a pre-trained TE model and keyword expansion to annotate unlabeled data.

(Miller, 1995) to expand the keywords. Specifically, we include words in the synsets that have one of the three relations (lemma, hyponyms, instance hyponyms) with synsets to which seed keywords belong.

The contributions are summarized as follows:

• We turn the TE model into an enhanced annotator by utilizing it to annotate massive amounts of unlabeled data and subsequently finetune it.

• To improve the efficiency, we propose to use keywords to filter out sentences with a low probability of expressing events. To improve the coverage of keywords, we expand the limited number of seed keywords using WordNet.

• The experimental results show that our method can outperform other baselines by 15% on the ACE05 dataset.

The code is available at:
https://github.com/ZeroNLP/ZS_TE

2. Related Work

Most existing event detection methods (Nguyen et al., 2016a; Lin et al., 2020) are supervised methods, relying on high-quality labeled data. Zero-shot event detection methods (Huang et al., 2016; Zhang et al., 2021; Lyu et al., 2021; Wang et al., 2021; Sainz et al., 2022) accomplish the task without using any annotated training data. Lyu et al. (2021); Sainz et al. (2022) used a pre-trained TE model to detect events. However, they used the TE model to infer the test data directly.

In this paper, the zero-shot setting means no annotated training data is available and inference is conducted on the entire test set. We follow Lyu et al. (2021) and Sainz et al. (2022) and call this setting as the zero-shot setting. Traditionally, the zero-shot setting (Larochelle et al., 2008) means training on some annotated seen data and testing on unseen data. In this paper, we do not refer to this setting.

Some existing event detection methods (Chen et al., 2017; Zeng et al., 2018; Tong et al., 2020) proposed to annotate unlabeled data using knowledge bases or augment data using a labeled dataset. Zeng et al. (2018) directly used the ACE05 event extraction dataset as a knowledge base to automatically generate annotation data consistent with the ACE05 event ontology. Tong et al. (2020) annotated more triggers using WordNet (Miller, 1995) and proposed a knowledge distillation model to leverage annotated triggers, where the teacher model is trained on an annotated dataset. Compared with the above methods, our method only uses a few keywords instead of a labeled dataset to annotate massive labeled data.

3. Methodology

3.1. Data Annotation

The general overview of generating annotated data is shown in Figure 2. There are two steps in data annotation. First, we expand keywords using WordNet (Miller, 1995). Secondly, we extract sentences that contain keywords from the New York Times (NYT) corpus (Sandhaus, 2008) and then use a pre-trained TE model to annotate them.

The seed keywords we used are human-provided. Zhang et al. (2021) create seed keywords for the event types in the ACE05-E+ (Lin et al., 2020). There are averaged 3 keywords for each event type. Inspired by Araki and Mitamura (2018); Tong et al. (2020), we use WordNet (Miller, 1995) to expand the keywords. Following the first step in Tong et al. (2020), we first disambiguate each seed keyword into WordNet sense using a tool named IMS (Zhong and Ng, 2010). We then re-

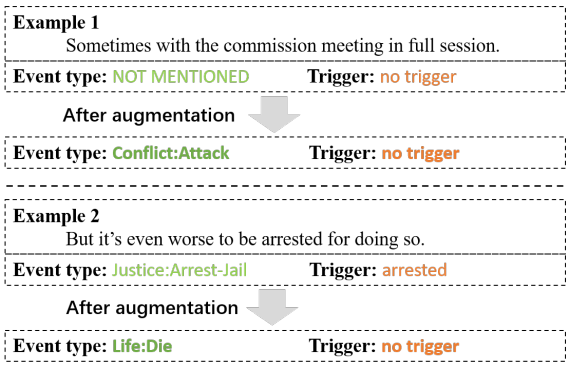

Figure 3: Augmented Examples.

trieve the words in the same synset, the hyponyms synset, and the instance hyponyms synset of the synsets that seed keywords belong to. All retrieved keywords are included in the final keyword set.

After that, we extract all the sentences that contain these keywords from the NYT corpus. We use a pre-trained textual entailment (TE) model to detect the event type for each sentence. The hypothesis of each event type is manually written according to the ACE event annotation guidelines. The premise is the input sentence. We set a filter threshold τ for the TE model to annotate data. If any hypothesis yields an entailment score larger than the filter threshold τ, we label the sentence as the corresponding event type. Furthermore, if the entailment scores of all hypotheses are smaller than the filter threshold τ, we annotate it as "not mentioned".

3.2. TE Model Finetuning

For the event detection task, we use the annotated NYT data to finetune the TE model. The input of the TE model is a pair of text, i.e., premise and hypothesis. The label of an input can be "entailment", "neutral", or "contradiction". We construct the input as follows. Since all sentences in annotated NYT data can naturally form "entailment" pairs, we have to create some "neutral" and "contradiction" pairs. We create a "neutral" pair by setting the hypothesis to a hypothesis corresponding to another event type for the same sentence. We create a "contradiction" pair by setting the hypothesis as "This sentence does not express any event." In the inference phase, we follow Lyu et al. (2021) which uses a confidence threshold γ (a hyperparameter) to determine whether a hypothesis is confident enough to be considered as expressing an event.

Most existing event detection methods (Nguyen and Nguyen, 2019; Lin et al., 2020) solve the ED task by detecting and classifying triggers. Remind that the goal of event detection is to recognize and categorize events, thus triggers could be viewed as intermediate results of this task. Considering the fact it is challenging and time-consuming for annotators to select the word(s) that most clearly express an event, some trigger-free methods (Liu et al., 2019; Zhao and Yang, 2022) have been proposed. They achieve competitive performance compared with mainstream methods, even without requiring trigger-level annotations. Our method falls into the trigger-free category.

Splits	Train	Dev	Test
Sentences	19,240	902	676
Events	4,419	468	424

Table 1: Statistics of ACE05-E+ Dataset.

In case triggers are needed in downstream tasks, we also propose a method to identify triggers given detected event types as inputs. We finetune the BERT (Devlin et al., 2018) model using the annotated NYT data via prompt tuning. Since each sentence contains at least one keyword, we consider the keyword(s) in the sentence in NYT data as the trigger(s). The prompt we used is "`[EVENT]` **<event>** `[EVENT]` The trigger is `[MASK]` `[SEP]` **<sentence>**", where **<event>** is a placeholder for the event type which is predicted by the finetuned TE model, and **<sentence>** is a placeholder for input, and `[EVENT]` is a special token.

If a sentence does not express any event, we let the trigger classification model to predict "no trigger." We propose two data augmentation methods to generate "no trigger" data. First, we randomly assign an event type and annotate it as "no trigger" for the "not mentioned" sentence. Second, we randomly assign a wrong event type for a sentence that actually expresses an event, and annotate it as "no trigger". Figure 3 illustrates the above two data augmentation methods.

4. Experiments

4.1. Experimental Settings

ACE05-E+ (Lin et al., 2020) dataset is a widely used dataset for the event extraction task, which pre-defines 8 event types and 33 subtypes. Details of dataset splits are shown in Table 1. Our method does not use the ACE05 training set.

Annotated NYT Data We used 107 keywords manually crafted by Zhang et al. (2021) as the seed keywords. After expansion, we obtain **1,347** keywords. Out of these expanded keywords, only 131 overlapped with trigger words found in the ACE05 dataset, which contains a total of 1,223 unique trigger words. Therefore, the overlapped words accounted for less than 10% of the trigger words in the dataset. Despite the relatively low cover-

age rate, the expansion process still resulted in improvements. We extract sentences that contain keywords in the New York Times (NYT) corpus (Sandhaus, 2008) from Sep.1987 to Dec.1988. We use a TE [1] model as an annotator. The TE model was trained on the MNLI dataset (Williams et al., 2018). Finally, we collected 322,570 data, including 268,406 single-event data and 54,164 multi-event data. The single-event (multi-event) data express one (more than one) event within a sentence. The annotated NYT data is available at the Github repository.

The compared methods include various zero-shot event detection baseline methods such as **Liberal_EE** (Huang et al., 2016), **ZS4IE** (Sainz et al., 2022), **ZS_Transfer** (Lyu et al., 2021), **ZS_CLEVE** (Wang et al., 2021), **Label_Aware** (Zhang et al., 2021) and **Chat4ED** (Li et al., 2023), three supervised methods including supervised **CLEVE**, **OneIE** (Lin et al., 2020) and **TBNNAM** (Liu et al., 2019) as our upper-bound methods.

Liberal_EE (Huang et al., 2016) applies Word Sense Disambiguation to extract semantic triggers and arguments, then constructs the trigger's event representation using semantically related functions. Finally, (Huang et al., 2016) names the clusters of triggers using the joint clustering network.

ZS4IE (Sainz et al., 2022) identifies potential targets for extraction through candidate generation, then employs user-defined templates to describe these candidates. They finally adopt a pre-trained TE model for inference.

ZS_Transfer (Lyu et al., 2021) formulates the zero-shot event detection as a Textual Entailment (TE) task. They treat a text piece as the premise.

ZS_CLEVE/CLEVE (Wang et al., 2021) ZS_CLEVE utilizes a contrastive learning framework to train a model on unlabeled data. They train a text encoder to learn event semantics and a graph encoder to learn event structures. By contrast, the supervised CLEVE is fine-tuned on annotated datasets instead of the AMR (Banarescu et al., 2013) structures of unsupervised corpora.

Label_Aware (Zhang et al., 2021) acquires the cluster of contextualized embedding for labels, then maps the contextualized representation of triggers and arguments to their corresponding types based on their similarities to clusters in the embedding space.

Chat4ED (Li et al., 2023) utilizes ChatGPT for Event Detection(ED) task in specific settings involving instructions. They ask ChatGPT to generate responses and select the most suitable answer from a predefined set of candidate labels.

OneIE (Lin et al., 2020) constructs an information graph based on entity mentions and event triggers, calculates label scores for nodes and links, and finally searches for globally optimal extraction results.

TBNNAM (Liu et al., 2019) detects the event types without detecting triggers. They encode the representation of a sentence based on target event types and propose a type-aware bias neural network with attention mechanisms. TBNNAM is a trigger-free event detection method that detect events without labeled triggers.

Like other zero-shot methods (Lyu et al., 2021; Zhang et al., 2021; Sainz et al., 2022), we tune hyperparameters on the development set. The filter threshold τ and the confidence threshold γ are both set to 0.9. We run each experiment three times and report the mean and std.

[1] huggingface.co/microsoft/deberta-v2-xlarge-mnli

Methods	P	R	F1
CLEVE (Wang et al., 2021)	78.1	81.5	79.8
OneIE (Lin et al., 2020)	74.3	70.3	72.2
TBNNAM (Liu et al., 2019)	76.2	64.5	69.9
Liberal_ EE (Huang et al., 2016)	55.7	45.1	49.8
ZS4IE (Sainz et al., 2022)	32.0	52.9	39.9
ZS_Transfer (Lyu et al., 2021)	31.7	60.6	41.7
ZS_CLEVE (Wang et al., 2021)	62.0	47.3	53.7
Label_Aware (Zhang et al., 2021)	54.1	53.1	53.6
Chat4ED (Li et al., 2023)	9.4	44.3	15.5
ZS_TE (our method)	**65.6**	72.3	**68.8**$\pm$0.003
w/o keyword expansion	54.0	**83.6**	65.6$\pm$0.006

Table 2: Precision, recall, and F1 scores (%) in the event detection task.

Data Combinations	P	R	F1
Single	58.0	74.9	65.3$\pm$0.018
Multi	37.3	**94.5**	53.5$\pm$0.012
Single + Multi	**65.6**	72.3	**68.8**$\pm$0.003

Table 3: Precision, recall, and F1 scores (%) of our methods in the event detection task using different data combinations.

4.2. Results Analysis

Event Detection. As shown in Table 2, our method outperforms the baseline ZS_CLEVE by 15%. Our method can achieve 86% performance of the upper-bound supervised CLEVE. Without using expanded keywords, our method drops 3%, which shows the effectiveness of the keyword expansion strategy. We also evaluate the effects of single-event data and multi-event data. As shown in Table 3, the combination of single-event and multi-event data yields the best F1 score.

Trigger Classification. In case triggers are needed in downstream tasks, we also propose a method to identify triggers given detected event types as inputs. As shown in Table 4, the trigger classification result drops 9%. The possible reason is that BERT (Devlin et al., 2018) model may not

be proficient in identifying and classifying words. We leave the problem of identifying triggers given detected events as inputs to future work.

ZS_TE (our method)	P	R	F1
Event Detection	65.6	72.3	68.8±0.003
Trigger Classification	66.9	54.1	59.8±0.002

Table 4: Precision, recall, and F1 scores (%) in the event detection and trigger classification task.

4.3. Low-resource Settings

We evaluate our method and two supervised methods on a low-resource setting in which we use 10% ~50% ACE data for training. In Figure 4, our method consistently outperforms TBNNAM (Liu et al., 2019) by a large margin in different proportions. Note that OneIE used trigger-level annotations while our method and TBNNAM do not use them. Direct comparison between OneIE and trigger-free methods is not fair. OneIE here serves as a reference rather than a baseline.

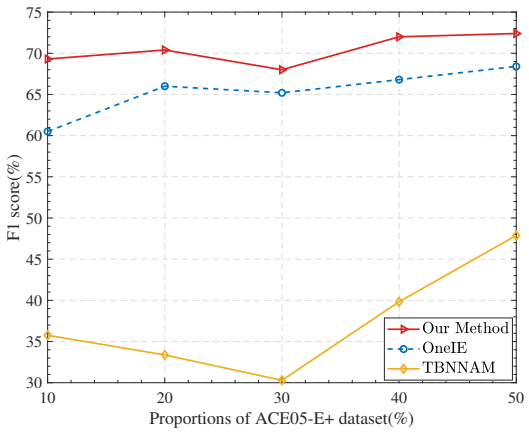

Figure 4: F1 scores (%) of our method and OneIE in the event detection task in different low-resource settings.

4.4. Hyperparameter Analysis

The search range of filter threshold τ is $\{0.5, \cdots, 0.9\}$. As shown in Figure 5, when the filter threshold τ is larger, the performance is better since a high filter threshold τ can filter out more samples with wrong event types.

The search range of confidence threshold γ is $\{0.5, \cdots, 0.9\}$. As shown in Figure 5, 0.9 yields the best performance and stability among all threshold values. When the confidence threshold γ is larger, the performance is better because a high confidence threshold γ can rule out more wrong event types.

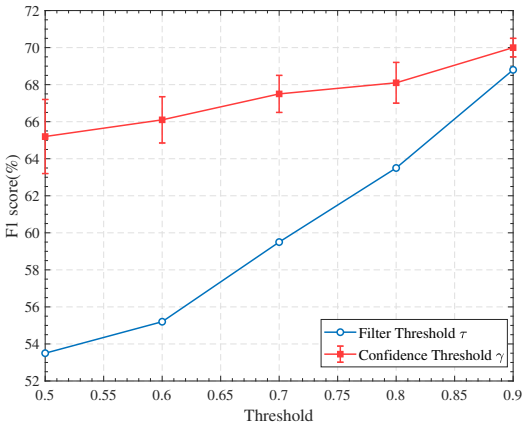

Figure 5: F1 scores (%) in the event detection task under different filter threshold τ and confidence threshold γ.

5. Conclusion

We explore a new way to use pre-trained TE models to detect event types. We turn the TE model into an enhanced annotator by utilizing it to annotate unlabeled data and subsequently finetune it. To improve the efficiency, we propose to use keywords to filter out sentences with a low probability of expressing events. To improve the coverage of keywords, we expand the limited number of seed keywords using WordNet.

Limitations

It is time-consuming to annotate unlabeled data using an off-the-shelf TE model. It takes an average of about 14 seconds to annotate one sentence in a single NVIDIA RTX 3090 GPU.

Acknowledgements

This research was supported by the Guangzhou Basic and Applied Basic Research Foundation (Grant No. 2023A04J1687), National Natural Science Foundation of China (Grant No. 6230070401), South China University of Technology-TCL Technology Innovation Fund.

6. References

Jun Araki and Teruko Mitamura. 2018. Open-domain event detection using distant supervision. In *Proceedings of COLING*, pages 878–891.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of EMNLP*, pages 632–642.

Yubo Chen, Shulin Liu, Xiang Zhang, Kang Liu, and Jun Zhao. 2017. Automatically labeled data generation for large scale event extraction. In *Proceedings of ACL*, pages 409–419.

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 ACL-IJCNLP*, pages 167–176.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Lifu Huang, Taylor Cassidy, Xiaocheng Feng, Heng Ji, Clare Voss, Jiawei Han, and Avirup Sil. 2016. Liberal event extraction and event schema induction. In *Proceedings of ACL*, pages 258–268.

Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio. 2008. Zero-data learning of new tasks. In *AAAI*, pages 646–651.

Bo Li, Gexiang Fang, Yang Yang, Quansen Wang, Wei Ye, Wen Zhao, and Shikun Zhang. 2023. Evaluating chatgpt's information extraction capabilities: An assessment of performance, explainability, calibration, and faithfulness. *arXiv preprint arXiv:2304.11633*.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of ACL*, pages 7999–8009.

Shulin Liu, Yang Li, Feng Zhang, Tao Yang, and Xinpeng Zhou. 2019. Event detection without triggers. In *Proceedings of NAACL-HLT*, pages 735–744.

Qing Lyu, Hongming Zhang, Elior Sulem, and Dan Roth. 2021. Zero-shot event extraction via transfer learning: Challenges and insights. In *Proceedings of ACL*, pages 322–332.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016a. Joint event extraction via recurrent neural networks. In *Proceedings of NAACL-HLT*, pages 300–309.

Thien Huu Nguyen, Lisheng Fu, Kyunghyun Cho, and Ralph Grishman. 2016b. A two-stage approach for extending event detection to new types via neural networks. In *Proceedings of the Workshop on Representation Learning for NLP*, pages 158–165.

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *Proceedings of the ACL-IJCNLP*, pages 365–371.

Trung Minh Nguyen and Thien Huu Nguyen. 2019. One for all: Neural joint modeling of entities and events. In *Proceedings of AAAI*, pages 6851–6858.

Oscar Sainz, Haoling Qiu, Oier Lopez de Lacalle, Eneko Agirre, and Bonan Min. 2022. ZS4IE: A toolkit for zero-shot information extraction with simple verbalizations. In *Proceedings of NAACL*, pages 27–38.

Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752.

Lei Sha, Feng Qian, Baobao Chang, and Zhifang Sui. 2018. Jointly extracting event triggers and arguments by dependency-bridge rnn and tensor-based argument interaction. In *Proceedings of AAAI*.

Meihan Tong, Bin Xu, Shuai Wang, Yixin Cao, Lei Hou, Juanzi Li, and Jun Xie. 2020. Improving event detection via open-domain event trigger knowledge. ACL.

Ziqi Wang, Xiaozhi Wang, Xu Han, Yankai Lin, Lei Hou, Zhiyuan Liu, Peng Li, Juanzi Li, and Jie Zhou. 2021. Cleve: contrastive pre-training for event extraction. In *Proceedings of ACL*, pages 6283–6297.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of NAACL-HLT*, pages 1112–1122.

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. 2019. Exploring pre-trained language models for event extraction and generation. In *Proceedings of ACL*, pages 5284–5294.

Ying Zeng, Yansong Feng, Rong Ma, Zheng Wang, Rui Yan, Chongde Shi, and Dongyan Zhao. 2018. Scale up event extraction learning via automatic training data generation. In *Proceedings of AAAI*, pages 6045–6052.

Hongming Zhang, Haoyu Wang, and Dan Roth. 2021. Zero-shot label-aware event trigger and argument classification. In *Findings of ACL*, pages 1331–1340.

Junchi Zhang, Yanxia Qin, Yue Zhang, Mengchi Liu, and Donghong Ji. 2019. Extracting entities and events as a single task using a transition-based neural model. In *IJCAI*, pages 5422–5428.

Yunyan Zhang, Guangluan Xu, Yang Wang, Daoyu Lin, Feng Li, Chenglong Wu, Jingyuan Zhang, and Tinglei Huang. 2020. A question answering-based framework for one-step event argument extraction. *IEEE Access*, 8:65420–65431.

Jiachen Zhao and Haiqin Yang. 2022. Trigger-free event detection via derangement reading comprehension. *arXiv preprint arXiv:2208.09659*.

Zhi Zhong and Hwee Tou Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of ACL*, pages 78–83.

7. Language Resource References

Lin, Ying and Ji, Heng and Huang, Fei and Wu, Lingfei. 2020. *A joint neural model for information extraction with global features*.

Sandhaus, Evan. 2008. *The new york times annotated corpus*.