Learning from a Friend: Improving Event Extraction via Self-Training with Feedback from Abstract Meaning Representation

Zhiyang Xu[♠] **Jay-Yoon Lee**[♣*] **Lifu Huang**[♠*]
[♠]Department of Computer Science, Virginia Tech
[♣]Graduate School of Data Science, Seoul National University
{zhiyangx,lifuh}@vt.edu
lee.jayyoon@snu.ac.kr

Abstract

Data scarcity has been the main factor that hinders the progress of event extraction. To overcome this issue, we propose a **S**elf-**T**raining with **F**eedback (STF) framework that leverages the large-scale unlabeled data and acquires feedback for each new event prediction from the unlabeled data by comparing it to the Abstract Meaning Representation (AMR) graph of the same sentence. Specifically, STF consists of (1) a base event extraction model trained on existing event annotations and then applied to large-scale unlabeled corpora to predict new event mentions as pseudo training samples, and (2) a novel scoring model that takes in each new predicted event trigger, an argument, its argument role, as well as their paths in the AMR graph to estimate a compatibility score indicating the correctness of the pseudo label. The compatibility scores further act as feedback to encourage or discourage the model learning on the pseudo labels during self-training. Experimental results on three benchmark datasets, including ACE05-E, ACE05-E$^+$, and ERE, demonstrate the effectiveness of the STF framework on event extraction, especially event argument extraction, with significant performance gain over the base event extraction models and strong baselines. Our experimental analysis further shows that STF is a generic framework as it can be applied to improve most, if not all, event extraction models by leveraging large-scale unlabeled data, even when high-quality AMR graph annotations are not available.[1]

1 Introduction

Event extraction (EE), which aims to identify and classify event triggers and arguments, has been a long-stand challenging problem in natural language processing. Despite the large performance leap brought by advances in deep learning, recent studies (Deng et al., 2021; Wang et al., 2021b) have shown that the data scarcity of existing event annotations has been the major issue that hinders the progress of EE. For example, in ACE-05[2], one of the most popular event extraction benchmark datasets, 10 of the 33 event types have less than 80 annotations. However, creating event annotations is extremely expensive and time-consuming, e.g., it takes several linguists over one year to annotate 500 documents with about 5000 event mentions for ACE-05.

To overcome the data scarcity issue of EE, previous studies (Chen and Ji, 2009; Liao and Grishman, 2011a; Ferguson et al., 2018a) develop self-training methods that allow the trained EE model to learn further by regarding its own predictions on large-scale unlabeled corpora as pseudo labels. However, simply adding the high-confidence event predictions to the training set inevitably introduces noises (Liu et al., 2021; Arazo et al., 2020; Jiang et al., 2018), especially given that the current state-of-the-art performance of event argument extraction is still less than 60% F-score. To tackle this challenge, we introduce a **S**elf-**T**raining with **F**eedback framework, named STF, which consists of an *event extraction model* that is firstly trained on the existing event annotations and then continually updated on the unlabeled corpus with self-training, and a *scoring model* that is to evaluate the correctness of the new event predictions (pseudo labels) from the unlabeled corpus, and the scores further act as feedback to encourage or discourage the learning of the event extraction model on the pseudo labels during self-training, inspired by the REINFORCE algorithms (Williams, 1992).

Specifically, the event extraction model of our STF framework can be based on any state-of-the-art architecture. In this paper, we choose OneIE (Lin et al., 2020) and AMR-IE (Zhang and Ji, 2021), due

[*] corresponding authors

[1]The source code and model checkpoints are publicly available at https://github.com/VT-NLP/Event_Extraction_with_Self_Training.

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

Findings of the Association for Computational Linguistics: ACL 2023, pages 10421–10437

to their superior performance and publicly available source code. The scoring model leverages the Abstract Meaning Representation (AMR) (Banarescu et al., 2013) which has been proven to be able to provide rich semantic and structural signals to map AMR structures to event predictions (Huang et al., 2016, 2018; Wang et al., 2021b) and thus their compatibility can indicate the correctness of each event prediction. The scoring model is a self-attention network that takes in a predicted event trigger, a candidate argument and its argument role, as well as their path in the AMR graph of the whole sentence, and computes a score ranging in [-1, 1] based on the compatibility between the AMR and the predicted event structure: -1 means *incompatible*, 1 means *compatible*, and 0 means *uncertain*. Inspired by the REINFORCE algorithm (Williams, 1992), we multiply the compatibility scores and the gradient of the EE model computed on the pseudo event labels during self-training, so as to (1) encourage the event extraction model to follow the gradient and hence maximize the likelihood of the pseudo label when it is compatible with the AMR structure; (2) negate the gradient and minimize the likelihood of the pseudo label when it is incompatible with the AMR structure; and (3) reduce the magnitude of the gradient when the scoring model is uncertain about the correctness of the pseudo label.

We take AMR 3.0[3] and part of the New York Times (NYT) 2004 corpus[4] as additional unlabeled corpora to enhance the event extraction model with STF, and evaluate the event extraction performance on three public benchmark datasets: ACE05-E[5], ACE05-E$^+$[6], and ERE-EN[7]. The experimental results demonstrate that: (1) the vanilla *self-training* barely improves event extraction due to the noise introduced by the pseudo examples, while the proposed STF framework leverages the compatibility scores from the scoring model as feedback and thus makes more robust and efficient use of the pseudo labels; (2) STF is a generic framework and can be applied to improve most, if not all, of the event extraction models optimized by gradient descent algorithm and achieves significant improvement over the base event extraction models and strong baselines on event argument extraction on the three public benchmark datasets; (3) By exploiting different unlabeled corpora with gold or system-based AMR parsing, STF always improves the base event extraction models, demonstrating that it can work with various qualities of AMR parsing. Notably, different from previous studies (Huang et al., 2018; Zhang and Ji, 2021; Wang et al., 2021b) that require high-quality AMR graphs as input to the model during both training and inference, STF does not require any AMR graphs during inference, making it more computationally efficient and free from the potential errors propagated from AMR parsing.

[3]https://catalog.ldc.upenn.edu/LDC2020T02.
[4]https://catalog.ldc.upenn.edu/LDC2008T19
[5]https://catalog.ldc.upenn.edu/LDC2006T06
[6]https://catalog.ldc.upenn.edu/LDC2006T06
[7]Deep Exploration and Filtering of Test (DEFT) program.

2 STF for Event Extraction

The event extraction task consists of three subtasks: event detection, argument identification and argument role classification. Given an input sentence $W = [w_1, w_2, ..., w_N]$, *event detection* aims to identify the span of an event trigger τ_i in W and assign a label $l_{\tau_i} \in \mathcal{T}$ where $\mathcal{T}$ denotes the set of target event types. *Argument identification* aims to find the span of an argument ε_j in W, and *argument role classification* further predicts a role $\alpha_{ij} \in \mathcal{A}$ that the argument ε_j plays in an event τ_i given the set of target argument roles $\mathcal{A}$.

Figure 1 shows the overview of our STF framework which consists of two training stages. At the first stage, a **base event extraction model** (Section 2.1) is trained on a labeled dataset. At the second stage, we apply the trained event extraction model to an unlabeled corpus to predict new event mentions. Instead of directly taking the new event predictions as pseudo training examples like the vanilla *self-training*, we propose a novel **scoring model** (Section 2.2) to estimate the correctness of each event prediction by measuring its compatibility to the corresponding AMR graph, and then take both event predictions and their compatibility scores to continue to train the base event extraction model while the scores update the gradient computed on pseudo labels (Section 2.3). After the training of the second stage, we get a new event extraction model and evaluate it on the test set.

2.1 Base Event Extraction Model

Our proposed framework can be applied to most, if not all, event extraction models. We select OneIE (Lin et al., 2020) and AMR-IE (Zhang and Ji, 2021) as base models given their state-of-the-art performance on the event extraction task and publicly available source code. Next, we briefly describe the common architectures in the two models

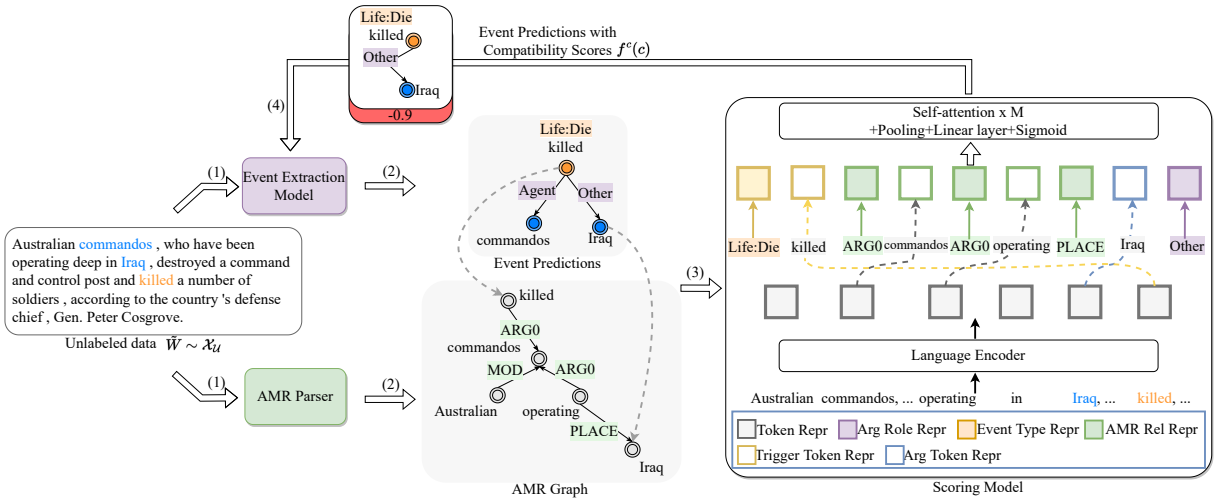


Figure 1: The overall framework of STF (We omit the first stage of STF.). Given an unlabeled sentence $\tilde{W}$: (1) run an event extraction model to compute event predictions and an AMR parser to parse it into an AMR graph; (2) map the predicted trigger and argument to corresponding nodes in the AMR graph, find their AMR path and combine it with the predicted event type and argument role into a new sequence (Life:Die, killed, ARG0, commandos, ARG0, PLACE, Iraq, Other); (3) feed the sequence into the scoring model to compute a compatibility score; (4) leverage the pseudo label and compatibility score to further update the event extraction model.

and refer readers to the original papers for more details. OneIE and AMR-IE perform event extraction in four[8] steps. **First**, a language model encoder (Devlin et al., 2019; Liu et al., 2019) computes the contextual representations $\mathbf{W}$ for an input sentence W. **Second**, two identification layers take in the contextual representations $\mathbf{W}$. One identifies the spans of event triggers and the other identifies the spans of arguments (i.e., entities). Both of them are based on a linear classification layer followed by a CRF layer (Lafferty et al., 2001) to capture the dependencies between predicted tags. They are optimized by minimizing the negative log-likelihood of the gold-standard tag path, which is denoted as $\mathbf{L}^{\text{Tri\_I}}$ and $\mathbf{L}^{\text{Arg\_I}}$ for trigger and argument identification, respectively. **Third**, for each trigger or argument candidate, we compute its representation by averaging the token representations within the whole identified span. Each trigger representation is fed into a classification layer to predict its type by minimizing the cross-entropy classification loss $\mathbf{L}^{\text{Tri\_C}}$. Each pair of trigger and argument representations are concatenated and fed into another classification layer to predict the argument role, which is also optimized by the cross-entropy loss $\mathbf{L}^{\text{Arg\_C}}$. **Finally**, both OneIE and AMR-IE learn an additional global feature vector to capture the interactions across sub-tasks (e.g., a *LOC* entity is impossible to be the *Attacker* of an *Attack* event) and instances (e.g., the *Defendant* of a *Sentence* event can also be an *Agent* of a *Die* event). During training, a global feature score is computed for the predicted information graph and the gold annotation, respectively, from their global feature vectors. The training objective is to minimize the gap between these two global feature scores, denoted as $\mathbf{L}^{G}$. Thus, the overall loss for the base event extraction model is:

$$\mathbf{L}^{\text{E}} = \mathbf{L}^{\text{Tri\_I}} + \mathbf{L}^{\text{Arg\_I}} + \mathbf{L}^{\text{Tri\_C}} + \mathbf{L}^{\text{Arg\_C}} + \mathbf{L}^{G},$$

As the first stage of our STF framework, we optimize the base event extraction model on labeled event mentions $\mathcal{X}_L$ based on $\mathbf{L}^{\text{E}}$ and the trained model will later be used to predict new event mentions for self-training.

2.2 Scoring Model

At the second stage of STF, we aim to further improve the event extraction model by taking the event mentions predicted from an external unlabeled corpus $\mathcal{X}_u$ as pseudo samples for self-training. To avoid the noise contained in the pseudo samples, we propose a scoring model that can evaluate the correctness of each event prediction. Our scoring model takes AMR graph as a reference motivated by the observation that *an event structure usually shares similar semantics and network topology as the AMR graph of the same sentence, thus their compatibility can be used to measure the correctness of each event structure*. This observation

[8] We only focus on event extraction task and thus omit the description of relation extraction.

has also been discussed and shown effective in previous studies (Rao et al., 2017; Huang et al., 2018; Zhang and Ji, 2021). However, previous studies directly take AMR graphs as input to the extraction model and thus require AMR graphs during both training and inference, making their performance highly dependent on the quality of AMR parsing. Different from them, our proposed STF only takes AMR graphs during reference to measure the correctness of event predictions during self-training, making it free from the potential errors propagation from AMR parsing during inference.

Given a sentence $\tilde{W} \in \mathcal{X}_u$ from the unlabeled corpus and a predicted trigger $\tilde{\tau}_i$ and its argument $\tilde{\varepsilon}_j$ from $\tilde{W}$, we aim to estimate a correctness score for each pair of the trigger and argument prediction based on its compatibility with their path in the AMR graph[9]. Thus, we first apply the state-of-the-art AMR parsing tool (Astudillo et al., 2020) to generate an AMR graph for $\tilde{W}$: $G = (V, E)$, $E = \{(v_i, e_{ij}, v_j)|e_{ij} \in \mathcal{R})\}$. We follow (Huang et al., 2016; Zhang and Ji, 2021) and group the original set of AMR relations into 19 categories[10], thus e_{ij} denotes a particular relation category and $\mathcal{R}$ denotes the set of AMR relation categories. Then, we identify the v_i, v_j from AMR graph G as the corresponding node of $\tilde{\tau}_i$, $\tilde{\varepsilon}_j$, by node alignment following Zhang and Ji (2021). Then, we utilize the Breadth First Search to find the shortest path $p_{i,j}$ that connects and includes, v_i and v_j in G. If there is no path between v_i and v_j, we add a new edge to connect them and assign *other* as the relation.

Given a predicted trigger $\tilde{\tau}_i$ and its type $\tilde{l}_{\tau_i}$, a predicted argument $\tilde{\varepsilon}_j$ and its argument role $\tilde{\alpha}_{ij}$, the scoring model estimates their correctness by taking $[\tilde{l}_{\tau_i}, p_{ij}, \tilde{\alpha}_{ij}]$ as input and outputs a compatibility score. As Figure 1 shows, it consists of a language model encoder (Devlin et al., 2019; Liu et al., 2019) to encode the sentence $\tilde{W}$ and obtain the contextual representations for the tokens[11], which are then used to initialize the representation of each node in p_{ij} based on the alignment between the input tokens and the nodes in AMR graph following Zhang and Ji (2021). We draw edge representations from the AMR relation embedding matrix $\mathbf{E}^{rel}$ and combine them with node representations to form $\mathbf{H}_{p_{ij}}$, a representation for path p_{ij}. We also get an event type representation $\mathbf{h}_{\tau_i}$ for $\tilde{l}_{\tau_i}$ from the event-type embedding matrix $\mathbf{E}^{tri}$ and an argument role representation $\mathbf{h}_{\alpha_{ij}}$ for $\tilde{\alpha}_{ij}$ from the argument role embedding matrix $\mathbf{E}^{arg}$. Here, $\mathbf{E}^{rel}$, $\mathbf{E}^{tri}$, and $\mathbf{E}^{arg}$ are all randomly initialized and will be optimized during training. Finally, we obtain the initial representations $\mathbf{H}_{ij}^{init} = [\mathbf{h}_{\tau_i}, \mathbf{H}_{p_{ij}}, \mathbf{h}_{\alpha_{ij}}]$ for the sequence $[\tilde{l}_{\tau_i}, p_{ij}, \tilde{\alpha}_{ij}]$.

To estimate the compatibility between the event trigger and argument prediction and their path in the AMR graph, we apply multi-layer *self-attention* (Vaswani et al., 2017) over the joint representation of the AMR path and the event prediction $\mathbf{H}_{ij}^{init}$ to learn better contextual representations for the sequence $[\tilde{l}_{\tau_i}, p_{ij}, \tilde{\alpha}_{ij}]$ and we add the position embedding $\mathbf{E}^{pos}$ to $\mathbf{H}_{ij}^{init}$ before feed it into the self-attention layers:

$$\mathbf{H}_{ij}^{final} = \textit{self-attention}(\mathbf{H}_{ij}^{init}) \times M,$$

where M denotes the number of attention layers.

Finally, we compute an overall vector representation $\hat{\mathbf{H}}_{ij}^{final}$ from $\mathbf{H}_{ij}^{final}$ via average-pooling and feed it into a *linear-layer* and a *Sigmoid* function to compute a probability c_{ij}, indicating the correctness of the predicted event trigger and argument. We optimize the scoring model based on the binary cross-entropy objective:

$$\mathbf{L}^{\text{Score}} = \text{BCE}\left(y_{ij}, c_{ij}; \psi\right),$$

where $y_{ij} \in (0, 1)$ is a binary label that indicates the argument role is correct ($y_{ij} = 1$) or not ($y_{ij} = 0$)[12], and ψ is the parameters of the scoring model. During training, we have gold triggers and arguments as positive training instances and we swap the argument roles in positive training instances with randomly sampled incorrect labels to create negative training instances. After training the scoring model, we will fix its parameters and apply it to self-training.

2.3 Self-Training with Feedback

To improve the base event extraction model with *self-training*, we take the new event predictions $(\tilde{\tau}_i, \tilde{l}_{\tau_i}, \tilde{\varepsilon}_j, \tilde{\alpha}_{ij})$ from the unlabeled corpus $\mathcal{X}_u$ as pseudo samples to further train the event extraction

[9]Comparing with the whole AMR graph, the path of the trigger and argument in the AMR graph shows more improvement for the scoring model.

[10]The details of AMR relation categories are shown in Appendix A.

[11]If a token is split into multiple subtokens, we average the representations of all subtokens to obtain an overall token representation.

[12]We don't not consider the cases where the trigger labels are incorrect, since by observation the semantics and structure of AMR graphs are more related to the argument role types between event triggers and their arguments.

model. The gradients of the event extraction model on each pseudo sample is computed as:

$$g_{ij}^{st} = \nabla_\theta \mathbf{L}^{\text{E}} \left(\tilde{W}, (\tilde{\tau}_i, \tilde{l}_{\tau_i}, \tilde{a}_{ij}, \tilde{\varepsilon}_j); \theta \right)$$

where θ denotes the parameters of the event extraction model. Note that there can be multiple event predictions in one sentence.

Due to the prediction errors of the pseudo labels, simply following the gradients g_{ij}^{st} computed on the pseudo labels can hurt model's performance. Thus, we utilize the correctness score c_{ij} predicted by the scoring model to update the gradients, based on the motivation that: (1) if an event prediction is compatible with the AMR structure, it's likely to be correct and we should encourage the model learning on the pseudo label; (2) on the other side, if an event prediction is incompatible with its AMR structure, it's likely incorrect and we should discourage the model learning on the pseudo label; (3) if the scoring model is uncertain about the correctness of the event prediction, we should reduce the magnitude of the gradients learned from the pseudo label. Motivated by this, we first design a transformation function f^c to project the correctness score $c_{ij} \in [0, 1]$ into a range $[-1, 1]$ where -1 (or $c_{ij} = 0$) indicates incompatible, 1 (or $c_{ij} = 1$) means compatible, and 0 (or $c_{ij} = 0.5$) means uncertain. Here, f^c is based on a linear mapping:

$$f^c(c_{ij}) = 2 \times c_{ij} - 1$$

We then apply the compatibility scores as feedback to update the gradients of the event extraction model on each pseudo sample during self-training:

$$\mathbf{L}^{\text{STF}} = \sum_{i,j} f^c(c_{ij}) \cdot \mathbf{L}^{\text{E}} \left(\tilde{W}, (\tilde{\tau}_i, \tilde{l}_{\tau_i}, \tilde{a}_{ij}, \tilde{\varepsilon}_j); \theta \right)$$

To improve the efficiency of *self-training*, we update the event extraction model on every minibatch, and to avoid the model diverging, we combine the supervised training and self-training, so the overall loss for STF is:

$$\mathbf{L} = \mathbf{L}^{\text{E}} + \beta \mathbf{L}^{\text{STF}}$$

where β is the combining ratio, $\mathbf{L}^{\text{E}}$ is computed on the labeled dataset $\mathcal{X}_L$ and $\mathbf{L}^{\text{STF}}$ is computed on the pseudo-labeled instances from $\mathcal{X}_u$.

3 Experimental Setups

For evaluation, we consider two base event extraction models: OneIE (Lin et al., 2020) and AMR-IE (Zhang and Ji, 2021) due to their superior performance on event extraction and publicly available source code, and demonstrate the effectiveness of STF on three benchmark datasets: ACE05-E, ACE05-E$^+$ and ERE-EN, with the same evaluation metrics following previous studies (Wadden et al., 2019; Lin et al., 2020; Zhang and Ji, 2021; Wang et al., 2022)[13]. To show the generalizability of STF, we explore two unlabeled corpora for self-training: (1) **AMR 3.0** (Knight et al., 2021) which originally contains 55,635 sentences in the training set while each sentence is associated with a manually annotated AMR graph. (2) **New York Times Annotated Corpus** (NYT) contains over 1.8 million articles that were published between 1987 to 2007. We randomly sample 55,635 sentences[14] from articles published in 2004. Because NYT dataset does not have AMR annotations, we run a pre-trained AMR parser (Astudillo et al., 2020) to generate system AMR parsing.

Besides taking the recent state-of-the-art event extraction studies (Wadden et al., 2019; Du and Cardie, 2020; Lin et al., 2020; Nguyen et al., 2021; Zhang and Ji, 2021; Lu et al., 2021; Wang et al., 2022; Hsu et al., 2022; Lu et al., 2022)[15] as baselines, we also compare our proposed STF with two other training strategies: (1) vanilla *Self-Training* (Rosenberg et al., 2005) which consists of two stages similar as STF but in the second stage takes each new event prediction from the unlabeled data with a probability higher than 0.9 based on the base event extraction model as a pseudo label and combines them with the labeled data to re-train the event extraction model; and (2) *Gradient Imitation Reinforcement Learning* (GradLRE) (Hu et al., 2021b). GradLRE encourages the gradients computed on the pseudo-labeled data to imitate the gradients computed on the labeled data by using the cosine distance between the two sources of gradients as a reward to perform policy gradi-

[13]The detailed statistics of ACE05-E, ACE05-E$^+$, and ERE-EN are shown in Appendix B.

[14]To show the effect of unlabeled dataset vs labeled dataset, we sample the same number of the unlabeled sentences as AMR 3.0

[15]The scores reported in (Nguyen et al., 2022) are not comparable in the table 3, as their results are not averaged across random seeds. We tried to report their averaged performance by running their model ourselves by contacting the authors, however, their code is publicly unavailable.

ent reinforcement learning (Sutton et al., 1999). GradLRE showed improvements over other self-training methods on low-resource relation extraction which is a similar task to argument role classification. Appendix C describes the training details for both baselines and our approach.

4 Results and Discussion

4.1 Evaluation of Scoring Model

We first evaluate the performance of the scoring model by measuring how well it distinguishes the correct and incorrect argument role predictions from an event extraction model. Specifically, we compute event predictions by running a fully trained event extraction model (i.e., OneIE or AMR-IE) on the validation and test sets of the three benchmark datasets. Based on the gold event annotations, we create a gold binary label (*correct* or *incorrect*) for each argument role prediction to indicate its correctness. For each event prediction, we pass it along with the corresponding AMR graph of the source sentence into the scoring model. If the correctness[16] predicted by the scoring model agrees with the gold binary label, we treat it as a true prediction for scoring model, otherwise, a false prediction.

To examine the impact of leveraging AMR in scoring model performance, we develop a baseline scoring model that shares the same structure with our proposed scoring model except that it does not take an AMR graph as an input. Specifically, the baseline scoring model just takes the event mention (triggers, arguments and argument labels) in order to measure the compatibility score. The baseline scoring model is essentially an ablation of our scoring model where the AMR path is absent. As shown in Table 2, the performance of our scoring model outperforms the baseline scoring model by 1.4-1.7 F-score on the test sets, demonstrating the effectiveness of AMR graph in characterizing the correctness of each event prediction.

In Table 1, we can observe that the semantics and structure of AMR paths can be easily mapped to argument role types. Sometimes, the even triggers are far from their arguments in plain text, but the AMR paths between them is short and informative. Another observation is that the scoring model tends to assign positive scores to argument roles that are more compatible with the AMR paths, although sometimes the scores for the gold argument roles are not the highest.

4.2 Evaluation of STF on Event Extraction

Table 3 shows the event extraction results of both our approach and strong baselines[17]. For clarity, in the rest of the section, we refer to our proposed framework as STF_{AMR} and our proposed framework with the baseline scoring model as $\text{STF}_{\text{W/O\_AMR}}$. We can see that, both STF_{AMR} and $\text{STF}_{\text{W/O\_AMR}}$ improve the performance of the event extraction models on argument role classification while the vanilla *self-training* and GradLRE barely work, demonstrating the effectiveness of leveraging the feedback to the pseudo labels during self-training.

We further analyze the reasons in terms of why the vanilla *self-training* and GradLRE do not work and notice that: due to the data scarcity, the base event extraction model (i.e., OneIE) performs poorly on many argument roles (lower than 40% F-score). Thus, the event predictions on unlabeled corpora can be very noisy and inaccurate. The model suffers from confirmation bias (Tarvainen and Valpola, 2017; Arazo et al., 2020; Pham et al., 2020): it accumulates errors and diverges when it's iteratively trained on such noisy pseudo labeled examples during *self-training*. In addition, we also notice that with *self-training*, the event extraction model becomes overconfident about its predictions. We check the averaged probability of all the argument role predictions on the unlabeled dataset which is 0.93. In such case, it is clear that the predicted probability can not faithfully reflect the correctness of the predictions, which is referred as the calibration error (Guo et al., 2017; Niculescu-Mizil and Caruana, 2005). Thus, the *self-training* process which relies on overconfident prediction can become highly biased and diverge from the initial baseline model. In GradLRE, the quality of the reward is highly depend on the averaged gradient direction computed during the supervised training process. However, due to the scarcity of the training data, the stored gradient direction can be unreliable. In addition, the gradient computed on the pseudo-labeled dataset with high reward is used to update the average gradient direction, which can introduce noises into the reward function. As seen

[16] When the correctness score $c^{ij} > 0.5$ computed by the scoring model, the predicted label is *correct*, otherwise, *incorrect*.

[17] We show the variance of Base OneIE and $+\text{STF}_{\text{AMR}}$ on three datasets in Appendix D.

Tell that to the family of Margaret Hassan, the school teacher who was brutally tortured and then slaughtered by these same guys, they aren't so bad are they Chris Matthews?	AMR Path: [slaughtered, ARG1, teacher, MODIFIER, Margaret Hassan] Pred Arg: O; Compatibility Score: -0.99 Gold Arg: Victim; Compatibility Score: 0.99
It is irritating enough to get sued by Sam Sloan; imagine how irritating it would be to get BEATEN by him because you have done something so egregious that a court is forced to agree with him.	AMR Path: [sued, ARG0, Sam Sloan] Pred Arg: Adjudicator; Compatibility Score: -0.99 Gold Arg: Plaintiff; Compatibility Score: 0.67
Protests against the action aimed at toppling Iraqi President Saddam Hussein were held in cities across Libya, Egypt and Lebanon, as well as in Amman, Damascus and the Gaza Strip.	AMR Path: [Protests, ARG0, held, ARG1, were, PLACE, Amman] Pred Arg: O; Compatibility Score: -0.52 Gold Arg: Place; Compatibility Score: 0.74
Meanwhile Blair arrived in Washington late Wednesday for two days of talks with Bush at the Camp David presidential retreat.	AMR Path: [arrived, OTHER, talks, PLACE, retreat] Pred Arg: Destination; Compatibility Score: -0.94 Gold Arg: O; Compatibility Score: 0.31

Table 1: Qualitative Results of the compatibility scores.

	ACE05-E		ACE05-E+		ERE-EN	
	Dev	Test	Dev	Test	Dev	Test
Scoring w/o AMR	87.4	85.9	87.9	86.9	82.8	83.1
Scoring w/ AMR	**88.2**	**87.4**	**88.8**	**88.6**	**84.4**	**84.5**

Table 2: The F-score (%) of the scoring models on various datasets. Scoring w/o AMR is the baseline scoring model without using AMR path. Scoring w/ AMR is the scoring model we proposed.

in Table 3, the best models of *self-training* and GradLRE are on par or worse than the baseline approach, and these approaches show the detrimental effects as they show a continuous decline of the performance as training proceeds.

By considering AMR structure, STF$_{\text{AMR}}$ encourages the event extraction models to predict event structures that are more compatible with AMR graphs. This claim is supported by Table 4, which compares the compatibility scores between the model without STF (OneIE baseline) and one with STF (OneIE +STF) framework on the three benchmark datasets. The compatibility scores are measured by the AMR based scoring models. We can clearly see that the compatibility scores measured on OneIE+STF$_{\text{AMR}}$ are much higher than the scores measured on base OneIE.

Lastly, we observe that OneIE+STF$_{\text{AMR}}$ outperforms AMR-IE+STF$_{\text{AMR}}$, even when AMR-IE performs better than OneIE baseline without STF. We argue the reason is that even though STF$_{\text{AMR}}$ does not need AMR parsing at inference time, AMR-IE does require AMR graphs at inference time which causes it to suffer from potential errors in the AMR parsing. On the other hand, OneIE trained by STF$_{\text{AMR}}$ does not require AMR graphs at inference time, making it free from potential error propagation. Figure 2 shows more examples to illustrate how the feedback from AMR structures in STF helps to improve event predictions.

4.3 Effect of Confidence Threshold

Intuitively, STF can leverage both certain (including compatible and incompatible) and uncertain pseudo labeled examples, as when the example is uncertain, the probability c predicted by the scoring model is close to 0.5 and thus $f^c(c)$ is close to 0, making the gradients computed on this pseudo-labeled example close to 0. To verify this claim, we conduct experiments with STF$_{\text{AMR}}$ by using the probability c predicted by the scoring model to determine certain and uncertain pseudo labels and analyzing their effect to STF$_{\text{AMR}}$. Note that we don't use the probability from the base event extraction model due to its calibration error (Guo et al., 2017) [18]. Specifically, we first select a threshold $s^{st} \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$. For each pseudo example, if the probability c predicted by the scoring model is higher than s^{st} (indicating a confident positive prediction) or lower than $1 - s^{st}$ (indicating a confident negative prediction), we will add it for STF$_{\text{AMR}}$. The higher the threshold s^{st}, the most certain pseudo labels we can select for STF$_{\text{AMR}}$. As Figure 3 shows, STF$_{\text{AMR}}$ can even benefit from the less-confident pseudo labeled examples with threshold s^{st} around 0.6, demonstrating that it can make better use of most of the predicted events from the unlabeled corpus for *self-training*.

4.4 Impact of AMR Parsing

AMR annotations are very expensive and hard to obtain. To show the potential of STF$_{\text{AMR}}$ in the scenarios where gold AMR parsing is not available, we conduct experiments by leveraging the NYT 2004 corpus as the external unlabeled corpus with system generated AMR parsing for self-training. As shown in Table 5, with system-based AMR, STF can also improve the performance of base event extraction models on all three benchmark datasets,

[18] See detailed explanations in Appendix **??**

	ACE05-E		ACE05-E+		ERE-EN	
	Tri-C	Arg-C	Tri-C	Arg-C	Tri-C	Arg-C
DyGIE (Wadden et al., 2019)	69.7	48.8	67.3	42.7	-	-
BERT_QA_Arg (Du and Cardie, 2020)	72.4	53.3	70.6	48.3	57.0	39.2
FourIE (Nguyen et al., 2021)	**75.4**	**58.0**	73.3	**57.5**	57.9	48.6
Text2Event (Lu et al., 2021)	71.9	53.8	71.8	54.4	59.4	48.3
DEGREE (Hsu et al., 2022)	73.3	55.8	70.9	56.3	57.1	49.6
Query_Extract (Wang et al., 2022)	-	-	**73.6**	55.1	**60.4**	**50.4**
UIE (Lu et al., 2022)	73.4	54.8	-	-	-	-
Base OneIE (Lin et al., 2020)	74.0	57.4	73.4	57.2	60.2	49.8
+*self-training** (Rosenberg et al., 2005)	74.0	57.2	**73.8**	57.3	60.1	49.4
+GradLRE* (Hu et al., 2021b)	74.6	57.4	73.5	57.4	60.5	50.3
+STF$_{w/o_AMR}$	74.4	57.9	**73.8**	57.6	60.4	51.0
+STF$_{AMR}$ (ours)	**75.0**	**58.9**	73.4	**59.0**	**60.6**	**52.0**
Base AMR-IE (Zhang and Ji, 2021)	74.4	57.7	73.4	57.2	60.4	50.5
+*self-training** (Rosenberg et al., 2005)	74.2	57.4	73.4	57.1	60.1	50.2
+GradLRE* (Hu et al., 2021b)	74.4	57.8	73.3	57.4	60.3	50.5
+STF$_{w/o_AMR}$	74.3	58.0	73.5	57.6	**60.5**	51.1
+STF$_{AMR}$ (ours)	**74.5**	**58.5**	**73.6**	**58.2**	60.4	**51.7**

Table 3: Test F1 scores of event trigger classification (Tri-C), and argument role classification (Arg-C) on three benchmark datasets. * denotes methods we re-implement to fit them into the event extraction task. Bold denotes the best performance in each local section and underline denotes the best global performance.

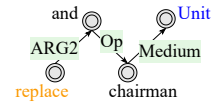
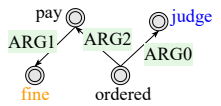
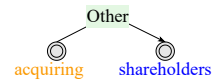
Sentence & Gold Event Mentions	**AMR Path**	**Base OneIE**	**STF**
With marathon talks at the top world body failing late Thursday to reconcile French and Russian opposition to US-British war plans, the *United States* upped its military presence, *deploying* more missile-firing warships to the Red Sea. *(Movement:Transport)* *deploying* -> *(Agent)* -> *United States*		Arg Role:*Other* Compatibility Score: *-1.0*	Arg Role:*Agent* Compatibility Score: *1.0*
In Paris, the French media group said parent company chairman Jean - Rene Fourtou will *replace* Diller as chairman and chief executive of US *unit*. *(Personnel:Start-Position)* *replace* -> *(Entity)* -> *unit*		Arg Role:*Other* Compatibility Score: *0.22*	Arg Role:*Entity* Compatibility Score: *0.93*
In a verdict handed down on Saturday, the *judge* also ordered Ranjha to pay a *fine* of 50,000 rupees (about 870 US dollars), they said. *(Justice:Fine)* *fine* -> *(Adjudicator)* -> *judge*		Arg Role:*Entity* Compatibility Score: *-0.98*	Arg Role:*Adjudicator* Compatibility Score: *1.0*
The Daily Planet raised 3.5 million dollars (2.2 million US) in its initial public offering with one of the new 600 *shareholders* *acquiring* 1.0 million dollars worth of shares. *(Transaction:Transfer-Money)* *acquiring* -> *(Other)* -> *shareholders*		Arg Role:*Buyer* Compatibility Score: *-1.0*	Arg Role:*Other* Compatibility Score: *0.72*

Figure 2: Qualitative results of STF. Examples are taken from the development and test splits of ACE05-E. The orange tokens denote event triggers and blue tokens denote arguments. The AMR paths are between event triggers and arguments. The Base OneIE and STF fields show the predicted argument roles from two methods respectively. All the predictions from STF are correct. The compatibility scores are computed by the same scoring model. Note that OneIE and STF do not use AMR graph at inference time and AMR graph is shown just to provide intuitions.

	ACE05-E		ACE05-E+		ERE-EN	
	Dev	Test	Dev	Test	Dev	Test
Base OneIE	70.1	68.4	76.9	61.9	76.4	69.2
+ STF$_{AMR}$	**72.2**	**70.8**	**80.2**	**64.0**	**78.0**	**75.1**

Table 4: The compatibility scores computed by scoring models on the development and test sets of the three benchmark datasets.

	ACE-E	ACE-E+	ERE-EN
Base OneIE	57.4	57.2	49.8
+ STF w/o AMR	57.9	57.6	51.0
+ STF w/ sys_AMR	58.2	58.1	51.4
+ STF w/ gold_AMR	58.9	59.0	52.0

Table 5: Performance comparison between using gold AMR, system-labeled AMR, and not using AMR.

and improve over the baseline scoring model without using AMR. The gap between STF with gold AMR and STF with system AMR is small, demonstrating that STF is more robust to the potential errors from AMR parsing.

5 Related Work

Most prior studies have been focusing on learning supervised models (Ji and Grishman, 2008; McClosky et al., 2011; Li et al., 2013; Chen et al., 2015; Feng et al., 2016; Nguyen et al., 2016; Wadden et al., 2019; Du and Cardie, 2020; Lin et al.,

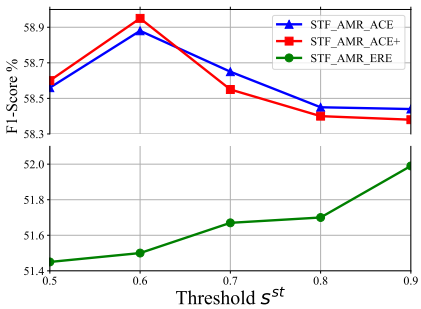


Figure 3: Performance change with different thresholds to select certain pseudo labeled examples for self-training.

2020; Zhang and Ji, 2021; Wang et al., 2022; wan; Nguyen et al., 2021) based on manually annotated event mentions. However, the performance of event extraction has been barely improved in recent years, and one of the main reasons lies in the data scarcity and imbalance of the existing event annotations. Several self-training and semi-supervised studies have been proposed to automatically enrich the event annotations. Huang and Riloff (2012) uses extraction patterns based on nouns that, by definition, play a specific role in an event, to automatically label more data. Li et al. (2014) proposes various event inference mechanisms to reveal additional missing event mentions. (Huang, 2020; Huang and Ji, 2020) propose semi-supervised learning to automatically induce new event types and their corresponding event mentions while the performance of old types is also improved. (Liao and Grishman, 2010, 2011b; Ferguson et al., 2018b) propose techniques to select a more relevant and informative corpus for self-training. All these studies cannot handle the noise introduced by the automatically labeled data properly. Compared with them, our STF framework leverages a scoring model to estimate the correctness of each pseudo-labeled example, which further guides the gradient learning of the event extraction model, thus it can efficiently mitigate the impact of the noisy pseudo-labeled examples.

Self-training has been studied for many years (Yarowsky, 1995; Riloff and Wiebe, 2003; Rosenberg et al., 2005) and widely adopted in many tasks including speech recognition (Kahn et al., 2020; Park et al., 2020), biomedical imaging (You et al., 2022a,b), parsing (McClosky et al., 2006; McClosky and Charniak, 2008), and pre-training (Du et al., 2021). Self-Training suffers from inaccurate pseudo labels (Arazo et al., 2020, 2019; Hu et al., 2021a) especially when the teacher model is trained on insufficient and unbalanced datasets. To address this problem, (Pham et al., 2020; Wang et al., 2021a; Hu et al., 2021a) propose to utilize the performance of the student model on the held-out labeled data as a Meta-Learning objective to update the teacher model or improve the pseudo-label generation process. Hu et al. (2021b) leverage the cosine distance between gradients computed on labeled data and pseudo-labeled data as feedback to guide the self-training process. (Mehta et al., 2018; Xu et al., 2021) leverage the span of named entities as constraints to improve semi-supervised semantic role labeling and syntactic parsing, respectively.

6 Conclusion

We propose a self-training with feedback (STF) framework to overcome the data scarcity issue of the event extract task. The STF framework estimates the correctness of each pseudo event prediction based on its compatibility with the corresponding AMR structure, and takes the compatibility score as feedback to guide the learning of the event extraction model on each pseudo label during self-training. We conduct experiments on three public benchmark datasets, including ACE05-E, ACE05-E$^+$, and ERE, and prove that STF is effective and general as it can improve any base event extraction models with significant gains. We further demonstrate that STF can improve event extraction models on large-scale unlabeled corpora even without high-quality AMR annotations.

Limitations

Our method utilizes the AMR annotations as additional training signals to alleviate the data scarcity problem in the event extraction task. In this problem setup, generally speaking, AMR annotations are more expensive than event extraction annotations. Nonetheless, in reality, the AMR dataset is much bigger than any existing event extraction dataset, and AMR parsers usually have higher performance than event extraction models. Leveraging existing resources to improve event extraction without requiring additional cost is a feasible and practical direction. Our work has demonstrated the effectiveness of leveraging the feedback from AMR to improve event argument extraction. However, it's still under-explored what additional information and tasks can be leveraged as feedback to improve trigger detection.

We did not have quantitative results for the alignment between AMR and event graphs. The authors randomly sampled 50 event graphs from ACE05-E and found 41 are aligned with their AMR graphs based on human judgment. In future work, more systematic studies should be conducted to evaluate the alignment.

There is a large gap between the validation and testing datasets in terms of label distribution on ACE05-E and ACE05-E+. We observe that performance improvement on the validation set sometimes leads to performance decreasing on the test set. Both the validation and test dataset miss certain labels for event trigger types and argument role types. The annotations in the training set, validation set, and test set are scarce and highly unbalanced, which causes the low performance on trained models. We argue that a large-scale more balanced benchmark dataset in the event extraction domain can lead to more solid conclusions and facilitate research.

Acknowledgments

We thank the anonymous reviewers and area chair for their valuable time and constructive comments. This research is based upon work supported by the Amazon Research Award. Jay-Yoon Lee was supported in part by the New Faculty Startup Fund from Seoul National University.

References

Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. 2019. Pseudo-labeling and confirmation bias in deep semi-supervised learning. *CoRR*, abs/1908.02983.

Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. 2020. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks, IJCNN 2020, Glasgow, United Kingdom, July 19-24, 2020*, pages 1–8. IEEE.

Ramón Fernandez Astudillo, Miguel Ballesteros, Tahira Naseem, Austin Blodgett, and Radu Florian. 2020. Transition-based parsing with stack-transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 1001–1007. Association for Computational Linguistics.

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 the 7th Linguistic Annotation Workshop and Interoperability with Discourse, LAW-ID@ACL 2013, August 8-9, 2013, Sofia, Bulgaria*, pages 178–186. The Association for Computer Linguistics.

Y. Chen, L. Xu, K. Liu, D. Zeng, and J. Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proc. ACL2015*.

Zheng Chen and Heng Ji. 2009. Can one language bootstrap the other: a case study on event extraction. In *Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing*, pages 66–74.

Shumin Deng, Ningyu Zhang, Luoqiu Li, Chen Hui, Tou Huaixiao, Mosha Chen, Fei Huang, and Huajun Chen. 2021. Ontoed: Low-resource event detection with ontology embedding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2828–2839.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Jingfei Du, Edouard Grave, Beliz Gunel, Vishrav Chaudhary, Onur Celebi, Michael Auli, Veselin Stoyanov, and Alexis Conneau. 2021. Self-training improves pre-training for natural language understanding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5408–5418. 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.

Xiaocheng Feng, Lifu Huang, Duyu Tang, Bing Qin, Heng Ji, and Ting Liu. 2016. A language-independent neural network for event detection. In *The 54th Annual Meeting of the Association for Computational Linguistics*, page 66.

James Ferguson, Colin Lockard, Daniel S Weld, and Hannaneh Hajishirzi. 2018a. 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.

James Ferguson, Colin Lockard, Daniel S. Weld, and Hannaneh Hajishirzi. 2018b. 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, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 359–364. Association for Computational Linguistics.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR.

I-Hung Hsu, Kuan-Hao Huang, Elizabeth Boschee, Scott Miller, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022. DEGREE: A data-efficient generation-based event extraction model. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 1890–1908. Association for Computational Linguistics.

Xuming Hu, Chenwei Zhang, Fukun Ma, Chenyao Liu, Lijie Wen, and Philip S. Yu. 2021a. Semi-supervised relation extraction via incremental meta self-training. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 487–496. Association for Computational Linguistics.

Xuming Hu, Chenwei Zhang, Yawen Yang, Xiaohe Li, Li Lin, Lijie Wen, and Philip S. Yu. 2021b. Gradient imitation reinforcement learning for low resource relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 2737–2746. Association for Computational Linguistics.

Lifu Huang. 2020. *Cold-start universal information extraction*. Ph.D. thesis.

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 the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 258–268.

Lifu Huang and Heng Ji. 2020. Semi-supervised new event type induction and event detection. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 718–724.

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

Ruihong Huang and Ellen Riloff. 2012. Bootstrapped training of event extraction classifiers. In *EACL 2012, 13th Conference of the European Chapter of the Association for Computational Linguistics, Avignon, France, April 23-27, 2012*, pages 286–295. The Association for Computer Linguistics.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of ACL-08: Hlt*, pages 254–262.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2018. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2309–2318. PMLR.

Jacob Kahn, Ann Lee, and Awni Hannun. 2020. Self-training for end-to-end speech recognition. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7084–7088.

Kevin Knight, Bianca Badarau, Laura Baranescu, Claire Bonial, Madalina Bardocz, Kira Griffitt, Ulf Hermjakob, Daniel Marcu, Martha Palmer, Tim O'Gorman, et al. 2021. Abstract meaning representation (amr) annotation release 3.0.

John D. Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.

Peifeng Li, Qiaoming Zhu, and Guodong Zhou. 2014. Employing event inference to improve semi-supervised chinese event extraction. In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pages 2161–2171. ACL.

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, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 73–82. The Association for Computer Linguistics.

Shasha Liao and Ralph Grishman. 2010. Filtered ranking for bootstrapping in event extraction. In *COLING 2010, 23rd International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2010, Beijing, China*, pages 680–688. Tsinghua University Press.

Shasha Liao and Ralph Grishman. 2011a. Can document selection help semi-supervised learning? a case study on event extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 260–265.

Shasha Liao and Ralph Grishman. 2011b. Can document selection help semi-supervised learning? A case study on event extraction. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA - Short Papers*, pages 260–265. The Association for Computer 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, ACL 2020, Online, July 5-10, 2020*, pages 7999–8009. Association for Computational Linguistics.

Kun Liu, Yao Fu, Chuanqi Tan, Mosha Chen, Ningyu Zhang, Songfang Huang, and Sheng Gao. 2021. Noisy-labeled NER with confidence estimation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 3437–3445. Association for Computational Linguistics.

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

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2795–2806, Online. Association for Computational Linguistics.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 5755–5772. Association for Computational Linguistics.

D. McClosky, M. Surdeanu, and C. D. Manning. 2011. Event extraction as dependency parsing. In *ACL*, pages 1626–1635.

David McClosky and Eugene Charniak. 2008. Self-training for biomedical parsing. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA, Short Papers*, pages 101–104. The Association for Computer Linguistics.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 4-9, 2006, New York, New York, USA*. The Association for Computational Linguistics.

Sanket Vaibhav Mehta, Jay Yoon Lee, and Jaime G. Carbonell. 2018. Towards semi-supervised learning for deep semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4958–4963. Association for Computational Linguistics.

Minh Van Nguyen, Viet Lai, and Thien Huu Nguyen. 2021. Cross-task instance representation interactions and label dependencies for joint information extraction with graph convolutional networks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 27–38. Association for Computational Linguistics.

Minh Van Nguyen, Bonan Min, Franck Dernoncourt, and Thien Nguyen. 2022. Joint extraction of entities, relations, and events via modeling inter-instance and inter-label dependencies. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4363–4374, Seattle, United States. Association for Computational Linguistics.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 300–309.

Alexandru Niculescu-Mizil and Rich Caruana. 2005. Predicting good probabilities with supervised learning. pages 625–632.

Daniel S. Park, Yu Zhang, Ye Jia, Wei Han, Chung-Cheng Chiu, Bo Li, Yonghui Wu, and Quoc V. Le. 2020. Improved noisy student training for automatic speech recognition. *CoRR*, abs/2005.09629.

Hieu Pham, Qizhe Xie, Zihang Dai, and Quoc V. Le. 2020. Meta pseudo labels. *CoRR*, abs/2003.10580.

Sudha Rao, Daniel Marcu, Kevin Knight, and Hal Daumé III. 2017. Biomedical event extraction using Abstract Meaning Representation. In *BioNLP 2017*,

pages 126–135, Vancouver, Canada,. Association for Computational Linguistics.

Ellen Riloff and Janyce Wiebe. 2003. Learning extraction patterns for subjective expressions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2003, Sapporo, Japan, July 11-12, 2003*.

Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. 2005. Semi-supervised self-training of object detection models. In *7th IEEE Workshop on Applications of Computer Vision / IEEE Workshop on Motion and Video Computing (WACV/MOTION 2005), 5-7 January 2005, Breckenridge, CO, USA*, pages 29–36. IEEE Computer Society.

Richard S. Sutton, David A. McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12, [NIPS Conference, Denver, Colorado, USA, November 29 - December 4, 1999]*, pages 1057–1063. The MIT Press.

Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1195–1204.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

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 2019, Hong Kong, China, November 3-7, 2019*, pages 5783–5788. Association for Computational Linguistics.

Sijia Wang, Mo Yu, Shiyu Chang, Lichao Sun, and Lifu Huang. 2022. Query and extract: Refining event extraction as type-oriented binary decoding. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 169–182. Association for Computational Linguistics.

Yaqing Wang, Subhabrata Mukherjee, Haoda Chu, Yuancheng Tu, Ming Wu, Jing Gao, and Ahmed Hassan Awadallah. 2021a. Meta self-training for few-shot neural sequence labeling. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, pages 1737–1747. ACM.

Ziqi Wang, Xiaozhi Wang, Xu Han, Yankai Lin, Lei Hou, Zhiyuan Liu, Peng Li, Juanzi Li, and Jie Zhou. 2021b. CLEVE: contrastive pre-training for event extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 6283–6297. Association for Computational Linguistics.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8:229–256.

Zhiyang Xu, Andrew Drozdov, Jay-Yoon Lee, Tim O'Gorman, Subendhu Rongali, Dylan Finkbeiner, Shilpa Suresh, Mohit Iyyer, and Andrew McCallum. 2021. Improved latent tree induction with distant supervision via span constraints. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 4818–4831. Association for Computational Linguistics.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd Annual Meeting of the Association for Computational Linguistics, 26-30 June 1995, MIT, Cambridge, Massachusetts, USA, Proceedings*, pages 189–196. Morgan Kaufmann Publishers / ACL.

Chenyu You, Weicheng Dai, Fenglin Liu, Haoran Su, Xiaoran Zhang, Lawrence H. Staib, and James S. Duncan. 2022a. Mine your own anatomy: Revisiting medical image segmentation with extremely limited labels. *CoRR*, abs/2209.13476.

Chenyu You, Weicheng Dai, Lawrence H. Staib, and James S. Duncan. 2022b. Bootstrapping semi-supervised medical image segmentation with anatomical-aware contrastive distillation. *CoRR*, abs/2206.02307.

Zixuan Zhang and Heng Ji. 2021. Abstract meaning representation guided graph encoding and decoding for joint information extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 39–49. Association for Computational Linguistics.

A Groups of AMR Relations

Table 7 shows the new categories and labels of AMR relations.

B The Statistics of Datasets

Table 7 shows the statistics of the three public benchmark datasets, including ACE05-E, ACE05-E+ and ERE-EN.

Group Label	AMR Relations
ARG0	ARG0
ARG1	ARG1
ARG2	ARG2
ARG3	ARG3
ARG4	ARG4
Destination	destination
Source	source
Instrument	instrument
Beneficiary	beneficiary
Prep roles	role starts with prep
Op roles	role start with op
Entity role	wiki, name
Arg-X role	ARG5, ARG6, ARG7 ARG8, ARG9
Place role	location, path, direction
Medium role	manner, poss, medium, topic
Modifier role	domain, mod, example
Part-whole role	part, consist, subevent, subset
Time role	calendar, century, day, dayperiod, decade, era, month, quarter, season, timezone, weekday, year, year2, time
Others	purpose, li, quant, polarity, condition, extent, degree, snt1, snt2, ARG5, snt3, concession, ord, unit, mode, value, frequency, polite, age, accompanier, snt4, snt10, snt5, snt6, snt7, snt8, snt9, snt11, scale, conj-as-if, rel

Table 6: The 19 groups of the AMR relations used in our paper.

C Training Details

For all experiments, we use Roberta-large as the language model which has 355M parameters. We train all of our models on a single A100 GPU.

Base OneIE We follow the same training process as (Lin et al., 2020) to train the OneIE model. We use BertAdam as the optimizer and train the model for 80 epochs with 1e-5 as learning rate and weight decay for the language encoder and 1e-3 as learning rate and weight decay for other parameters. The batch size is set to 16. We keep all other hyperparameters the same as (Lin et al., 2020). For each dataset we train 3 OneIE models and report the averaged performance.

Base AMR-IE We follow the same training process as (Zhang and Ji, 2021) to train the AMR-IE model. We use BertAdam as the optimizer and train the model for 80 epochs with 1e-5 as learning rate and weight decay for the language encoder and 1e-3 as learning rate and weight decay for other parameters. The batch size is set to 16. We keep all other hyperparameters exactly the same as (Zhang and Ji, 2021). For each dataset we train 3 AMR-IE models and report the averaged performance.

Scoring Model We use BertAdam as the optimizer and train the score model for 60 epochs with 1e-5 as learning rate and weight decay for the language encoder and 1e-4 as learning rate and weight decay for other parameters. The batch size is set to 10. The scoring model contains two self-attention layers. We train 3 scoring models and reported the averaged performance.

Self-Training For *self-training* we use SGD as optimizer and continue to train the converged base OneIE model for 30 epochs with batch size 12, learning rate 1e-4, weight decay for the language encoder as 1e-5, and learning rate 1e-3 and weight decay 5e-5 for all other parameters except the CRF layers and global features which are frozen. For *self-training*, we use 0.9 as the threshold to select the confident predictions as pseudo-labeled instances. For all the experiments, we train 3 models and report the averaged performance.

Gradient Imitation Reinforcement Learning For GradLRE, we use the BertAdam as the optimizer with batch size 16, learning rate 1e-5 and weight decay 1e-5 for the language encoder and learning rate 1e-3 and weight decay 1e-3 for other parameters to first train OneIE model for 60 epochs. The standard gradient direction vector is computed by averaging the gradient vector on each optimization step. Then following the same training process in the original paper, we perform 10 more epochs of *Gradient Imitation Reinforcement Learning*, and set the threshold for high reward as 0.5. For all the experiments, we train 3 models and report the averaged performance.

Self-Training with Feedback from Abstract Meaning Representation For STF, we first train the OneIE model on the labeled dataset for 10 epochs and continue to train it on the mixture of unlabeled data and labeled dataset for 70 more epochs with batch size 10, learning rate 1e-4, weight decay for the language encoder as 1e-5, and learning rate 1e-3 and weight decay 5e-5 for all other parameters. We leverage a linear scheduler to compute the value for the loss combining ratio β. The value of β is computed as $\frac{\text{epoch}}{70}$. For all the experiments, we train 3 models and report the averaged performance. For model selection, we propose a new method called *Compatibility-Score Based Model*

	ACE05-E			ACE05-E+			ERE-EN		
	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test
# Sent	17,172	923	832	19,240	902	676	14,736	1,209	1,163
# Entities	29,006	2,451	3,017	47,525	3,422	3,673	38,864	3,320	3,291
# Events	4,202	450	403	4,419	468	424	6,208	525	551

Table 7: The statistics of the three benchmarks used in our paper.

Selection which is discussed in the following paragraph.

Compatibility-Score Based Model Selection The data scarcity problem not only appears in the training data of ACE-05, ACE-05+ and ERE-EN but appears in the development set. For example, in ACE-05, the development set only contains only 603 labeled argument roles for 22 argument role classes and 7 argument role classes have lees than 10 instances. To alleviate this problem, we propose to leverage part of the large-scale unlabeled dataset as a held-out development set. At the end of each epoch, instead of evaluating the event extraction model on the development set, we run the event extraction model on the unlabeled held-out development set to make event predictions and run the scoring model on the event predictions to compute compatibility scores. We utilize the averaged compatibility scores computed on all instances in the unlabeled held-out development datasets as the model selection criteria. We argue this is another application of the scoring model since its goal is to evaluate the correctness of event predictions. The size of the unlabeled held-out development set is 2,000.

D Results of Base OneIE and +STF$_{\text{AMR}}$

We show the F1 scores of Base OneIE and +STF$_{\text{AMR}}$ on three benchmark datasets with variances denoted. As one can see that Base OneIE and +STF$_{\text{AMR}}$ have similar variances on all three datasets except ACE05-E+. We leave how to reduce the variance of argument role classification to future work.

	ACE05-E	ACE05-E+	ERE-EN
	Arg-C	Arg-C	Arg-C
Base OneIE	57.4 $\pm$1.23	57.2 $\pm$0.32	49.8 $\pm$0.45
+STF$_{\text{AMR}}$ (ours)	58.9 $\pm$1.28	59.0 $\pm$1.03	52.0 $\pm$0.40

Table 8: Test F1 scores of argument role classification (Arg-C) on three benchmark datasets.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
7

☒ A2. Did you discuss any potential risks of your work?
We are not aware of potential risks in our work.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

3

☑ B1. Did you cite the creators of artifacts you used?
3

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
We don't introduce new artifacts in the paper.

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
not applicable

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
we didn't collect data

☒ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
not applicable

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Appendix B

C ☑ Did you run computational experiments?

4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
appendix C

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
appendix C

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
appendix C

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
3

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

not applicable

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.