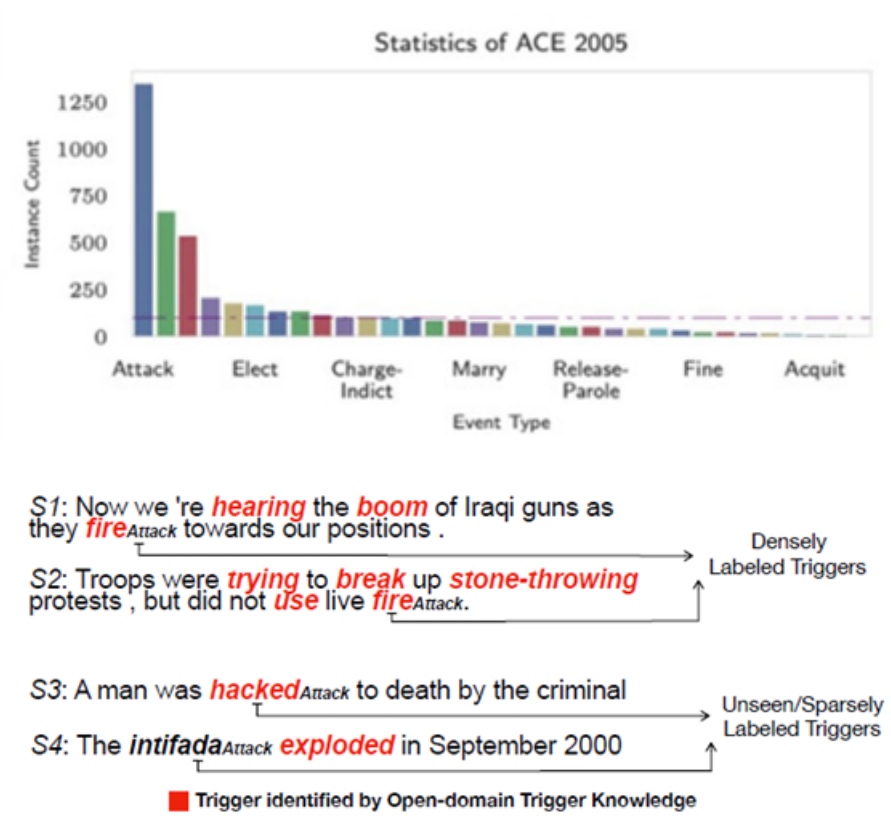
Improving Event Detection via Open-domain Trigger Knowledge

Motivation

The long tail issue in Event Detection: 78.2% trigger words with frequency less than 5.



Trigger words in sentences can be divided into Densely, Sparsely and Unseen labels. Previous semi-supervised and distantly-supervised methods can't solve this problem well.

Method	Unseen	Sparse	Dense
$\text{DMBERT}_{sup-only}$	54.4	72.5	84.1
$\text{BOOTSTRAP}_{semi-sup}$	56.6	73.6	86.9
$\text{DGBERT}_{distant-sup}$	54.7	72.8	84.3

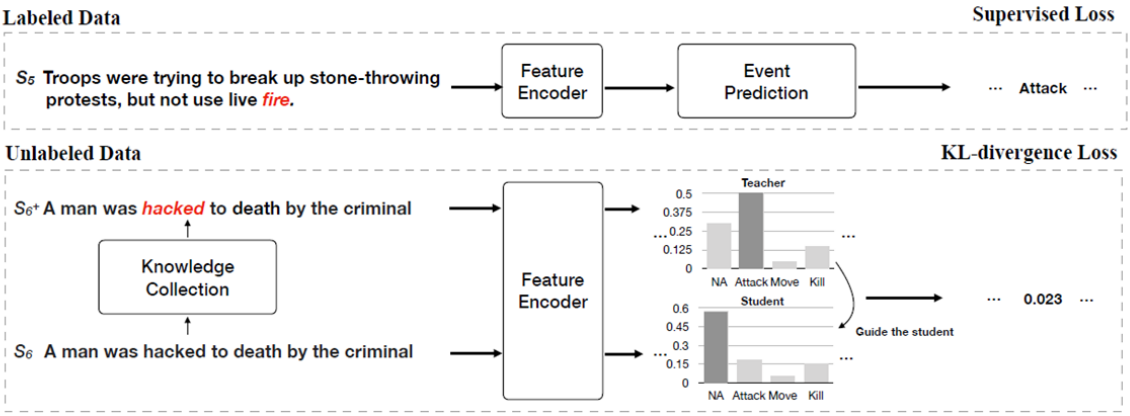
Contribution

- To the best of our knowledge, we are the first to leverage the wealth of the open-domain trigger knowledge to improve ED.
- We propose a novel teacher-student model (EKD) that can learn from both labeled and unlabeled data, so as to improve ED performance by reducing the in-built biases in annotations.
- Experiments on benchmark ACE2005 show that our method surpasses nine strong baselines which are also enhanced with knowledge. Detailed studies show that our method can be

conveniently adapted to distill other knowledge, such as entities.

Methodology

Architecture



Knowledge enrichment (Wordnet)

Hypothesis: The word sense in wordnet contains information about whether it is a trigger。

- Step 1: Disambiguate all words in the target sentence and convert them to the wordnet standard word sense. (*Using nltk package*)
- Step 2: Determine whether word sense is trigger. (Using a dictionary lookup in *Open-Domain Event Detection using Distant Supervision*)

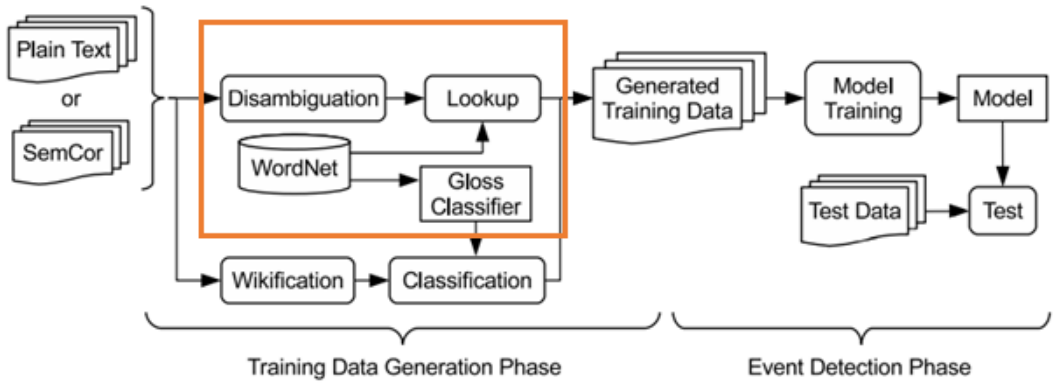


Figure 1: An overview of our distantly-supervised open-domain event detection.

Using New York Times corpus as unlabeled data, get 733,848 annotated sentences :

We 're *hearing* the *boom* of Iraqi guns as they ***fire*** towards our positions

We 're *hearing* the *boom* of Iraqi guns as they *fire* towards our positions

Statistic of Event-oriented words	
Total Number	2.65 million
Average Number per Sentence	3.6

After knowledge enrichment, one sentence is derived into three sentences:

- Raw sentence: $S = \{w_1, w_2, \ldots, w_i, \ldots, w_n\}$;
- Knowledge-attending Sentences: $S^+ = \{w_1, w_2, \ldots, B-TRI, w_i, E-TRI, \ldots, w_n\}$;
- Knowledge-absent Sentences: $S^- = \{w_1, w_2, \ldots, [MASK], \ldots, w_n\}$;

The `B-TRI` & `E-TRI` tokens are paper-defined and using BERT MLM task to get their embeddings.

Event prediction

Using **bert-encoder** as feature representation, using a **full-connected layer** as classification layer.

- Output: `O`, O_{ijc}: The logistical value of the `j-th` word in the `i-th` sentence being `c-th` type of trigger;
- Loss:
  - $p\left(Y_{(i)} \mid S_{(i)}, \theta\right)=\sum_{j=1}^{n} \frac{\exp \left(O_{ijc}\right)}{\sum_{c=1}^{C} \exp \left(O_{ijc}\right)} / n$;
  - $J_{L}(\theta)=-\sum_{i=1}^{N_{L}} \log p\left(Y_{(i)} \mid S_{(i)}, \theta\right)$.

Knowledge distillation

- Using a **teacher model** and a **student model**, which share the **same parameters**;
- Feed teacher model with S^{+} while feed student model with S^{-};
- Using `KL-divergence` on teacher and student models :

$$
\begin{aligned}
J_{T}(\theta) &=\mathbf{K L}\left(p\left(Y \mid S^{+}, \theta\right) \| p\left(Y \mid S^{-}, \theta\right)\right) \\
&=\sum_{k=1}^{N_{L}+N_{U}} p\left(Y_{(k)} \mid S_{(k)}^{+}, \theta\right) \frac{p\left(Y_{(k)} \mid S_{(k)}^{+}, \theta\right)}{p\left(Y_{(k)} \mid S_{(k)}^{-}, \theta\right)}
\end{aligned}
$$

- Train model **jointly** by: $J(\theta)=J_{L}(\theta)+\lambda * J_{T}(\theta)$.

Experiments

Experiment setting

- Datasets:
  - ACE2005 contains **13,672 labeled sentences** distributed in **599 articles**;
  - unlabeled data: **40,236**;
- Evaluation: **Precision, Recall** and **micro-averaged F1 scores** in the form of percentage over all **33 events**;
- Hyperparameters:
  - BERT : **24 16-head attention layers** and **1024 hidden embedding dimension**;
  - batch size: **32**;
  - maximum sequence length: **128**;
  - λ: 1;
- learning rate: **3e-5**;
- best result: around **12,500 epochs**

Overall performance

Method	Precision	Recall	F1
DMCNN	75.6	63.6	69.1
DLRNN	77.2	64.9	70.5
ANN-S2	78.0	66.3	71.7
GMLATT	78.9	66.9	72.4
GCN-ED	77.9	68.8	73.1
Lu's DISTILL	76.3	71.9	74.0
TS-DISTILL	76.8	72.9	74.8
AD-DMBERT	77.9	72.5	75.1
DRMM	77.9	74.8	76.3
EKD (Ours)	**79.1**	**78.0**	**78.6**

Test without trigger knowledge

Table 3: Performance of test set with or without open-domain trigger knowledge

Test Set	P	R	F
without knowledge	78.8	78.1	78.4
with knowledge	79.1	78.0	78.6

The model has already learned the open-domain trigger knowledge and don't need to tag trigger in the test data.

Domain Adaption & Various Labeling Frequencies

Table 4: Performance on domain adaption. We train our model on two source domains bn and nw, and test our model on three target domains bc, cts and wl.

Methods	In-Domain (bn+nw)			bc			cts			wl		
	P	R	F	P	R	F	P	R	F	P	R	F
MaxEnt	74.5	59.4	66.0	70.1	54.5	61.3	66.4	49.9	56.9	59.4	34.9	43.9
Joint	73.5	62.7	67.7	70.3	57.2	63.1	64.9	50.8	57.0	59.5	38.4	46.7
Nguyen's CNN	69.2	67.0	68.0	70.2	65.2	67.6	68.3	58.2	62.8	54.8	42.0	47.5
PLMEE	77.1	65.7	70.1	72.9	67.1	69.9	70.8	64.0	67.2	62.6	51.9	56.7
EKD (ours)	**77.8**	**76.1**	**76.9**	**80.8**	65.1	**72.1**	**71.7**	61.3	66.1	**69.0**	49.9	**57.9**

Table 5: Performance of our method on various labeling frequencies trigger words.

Methods	Unseen			Sparsely Labeled			Densely Labeled		
	P	R	F	P	R	F	P	R	F
DMBERT$_{supervised-only}$	66.7	45.9	54.4	74.4	70.7	72.5	84.8	83.5	84.1
DGBERT$_{distant-supervised}$	76.5	42.6	54.7	75.7	70.1	72.8	85.9	83.8	84.3
BOOTSTRAP$_{semi-supervised}$	73.7	45.9	56.6	76.0	71.3	73.6	90.6	83.5	86.9
EKD (ours)	**79.0**	**52.0**	**62.7**	**80.8**	**72.4**	**76.4**	**92.5**	82.2	**87.1**

Knowledge-Agnostic

Table 7: Knowledge-Agnostic.

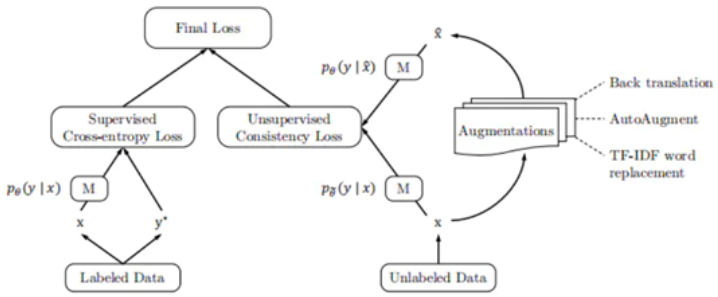
Knowledge Type	Methods	Metrics P	R	F
Entity	TS-DISTILL	76.8	72.9	74.8
	EKD-Ent	74.5	78.6	**76.5**
	improvement	-2.3	+4.7	+1.7
Syntactic	GCN-ED	77.9	68.8	73.1
	EKD-Syn	76.5	76.3	**76.4**
	improvement	-1.4	+7.5	+3.3
Argument	ANN-S2	78.0	66.3	71.7
	EKD-Arg	75.8	78.4	**77.1**
	improvement	-2.2	+23.1	+5.4

The architecture can also learn open-domain knowledge of entity, syntactic and argument.

Codes

https://github.com/shuaiwa16/ekd

Modified from Google/uda



Discussion

Advantage

- Combines distant-supervised (wordnet) and semi-supervised (EKD)
- Data enhancement method

Disadvantage

- Did not do ablation analysis
- The experiment of module versatility has no details
- The code is incomplete and the authenticity is questionable