Graph Learning Regularization and Transfer Learning for Few-Shot Event Detection

Viet Dac Lai[1], Minh Van Nguyen[1], Thien Huu Nguyen[1], Franck Dernoncourt[2]
{vietl,minhnv,thien}@cs.uoregon.edu,franck.dernoncourt@adobe.com
[1]Dept. of Computer and Information Science, University of Oregon, Eugene, Oregon, USA
[2]Adobe Research, San Jose, California, USA

ABSTRACT

We address the poor generalization of few-shot learning models for event detection (ED) using transfer learning and representation regularization. In particular, we propose to transfer knowledge from open-domain word sense disambiguation into few-shot learning models for ED to improve their generalization to new event types. We also propose a novel training signal derived from dependency graphs to regularize the representation learning for ED. Moreover, we evaluate few-shot learning models for ED with a large-scale human-annotated ED dataset to obtain more reliable insights for this problem. Our comprehensive experiments demonstrate that the proposed model outperforms state-of-the-art baseline models in the few-shot learning and supervised learning settings for ED. Code and data splits are available at https://github.com/laiviet/ed-fsl

CCS CONCEPTS

• **Computing methodologies → Regularization**.

KEYWORDS

event detection, few-shot learning, transfer learning

ACM Reference Format:
Viet Dac Lai[1], Minh Van Nguyen[1], Thien Huu Nguyen[1], Franck Dernoncourt[2]. 2021. Graph Learning Regularization and Transfer Learning for Few-Shot Event Detection. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3404835.3463054

1 INTRODUCTION

Event Detection (ED) is a natural language processing (NLP) task that detects event triggers/mentions (i.e., the most important words to clearly express an event) and categorizing them into a set of predefined event types. For instance, given the following sentence, an ED model should detect the word *skirmish* as an event trigger and classify it as *conflict-attack*:

Fans ***skirmish*** *ahead of the match in Marseille on Saturday.*

Existing works have mostly solved ED in the supervised learning setting [4, 8, 30, 41]. In real-world applications, a major problem of these supervised ED models is the poor transferability to new event types [15]. As such, the predictions of trained models are limited to predefined event types, thereby failing to extract event triggers of new types. Recent studies address this issue by formulating ED as a low-shot learning problem in low-resource conditions including zero-shot learning [15] and few-shot learning (FSL) [20]. These methods enable models to effectively extend the operation to new event types, for which no or a few training samples are annotated. In this paper, we focus on the few-shot learning setting, aiming to address three issues in the existing FSL methods for ED.

First, current models in few-shot learning for ED are only evaluated on datasets with small numbers of event types. For instance, recent few-shot learning studies [20] mainly use the popular ACE 2005 dataset that only contains 33 event types [11]. This makes the reported performance in those prior work less reliable as the utilized datasets cannot cover a wide range of possible event types to better estimate the generalization. Besides, due to the small numbers of event types, prior FSL work for ED has to use the same event types for the development and test datasets [20], thereby violating the requirement of disjoint event types for the training, testing, and development data in FSL and leading to an unrealistic setting for this problem. To address this issue, this work conducts the first FSL research for ED where the evaluation is performed on a human-annotated ED dataset with a large number of event types to enable more realistic and reliable performance. In particular, we employ a recently released event extraction dataset RAMS, *Roles Across Multiple Sentences* [7] (with 139 event types), to extensively evaluate various FSL models for ED in this work.

The second issue involves the failure to exploit knowledge from ED-related datasets/tasks to advance the generalization for the models [20]. As such, our intuition is that FSL models can generalize better to new event types if they are augmented with knowledge (knowledge transferring) from datasets with a large number of event types (ideally all the possible event types).

Motivated by the prior work on supervised ED [26], we resort to Semcor, a human-annotated dataset for word sense disambiguation (WSD), to obtain the knowledge about open-domain event types and transfer it to FSL models for ED. Besides the high quality of the data (due to the human annotation), Semcor provides the annotations for a large number of word senses in WordNet that can cover a variety of event types and potentially improve the type generalization of the augmented FSL models [26]. To our knowledge, this is the first work to explore transfer learning for FSL in ED.

Finally, to further improve the performance of FSL models for ED, we propose a novel regularization mechanism to produce better representation vectors. Our mechanism differentiates two types of words in a sentence for an event trigger, i.e., relevant words and

SIGIR '21, July 11–15, 2021, Virtual Event, Canada

ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3463054

irrelevant words. On the one hand, we argue that the representation vector for the event trigger should be computed mainly based on the relevant words. On the other hand, we expect that the irrelevant words can also provide useful training signals for ED models by introducing constraints to force these words to not contribute significantly to the learned hidden vectors. As such, in addition to inducing hidden vectors based on the relevant words, we propose to obtain representation vectors from every word in the sentence (i.e., including both relevant and irrelevant words). To minimize the contribution of the irrelevant words, we then introduce a regularization term to enforce the similarity between the hidden vectors from the relevant words and the whole sentence. Our extensive experiments demonstrate the effectiveness of the proposed techniques for ED, leading to a state-of-the-art performance in both FSL and supervised learning settings.

2 BACKGROUND

In few-shot learning, we are given a set of labeled data $\mathcal{D}^{train}$ corresponding to a set of classes $\mathcal{Y}^{train}$. A learning model has to exploit knowledge from this data so later it can predict on a completely new set of classes $\mathcal{Y}^{test}$ (with the labeled data set $\mathcal{D}^{test}$), in which only a few annotated samples (e.g., 5 or 10) is provided for each new class. As such, the model is trained over a set of classes $\mathcal{Y}^{train}$, then it is tested on $\mathcal{Y}^{test}$ which is disjoint from $\mathcal{Y}^{train}$.

Few-Shot Learning To emulate the above setting, we follow the conventional *episodic training* [38] to sample training tasks. In each training episode (i.e., training iteration), we sample a subset of N classes $\mathcal{Y}$ from $\mathcal{Y}^{train}$. For each class $t_i \in \mathcal{Y}$, we sample $K + Q$ examples of which K examples serve as training data, and Q examples are used for testing data. Gathering training data and testing data for all classes, we have a *meta-training set* and a *meta-testing set*. In the literature, they are also called *support set* and *query set* respectively. In each training episode, the parameters of a learner are updated based on the loss over the query set.

Once we have a meta-trained model, the same episodic sampling process is employed multiple times over the $\mathcal{D}^{test}$ to evaluate how quickly the model adapts to a brand-new set of classes. In particular, we first sample N classes from $\mathcal{Y}^{test}$, then, we sample K examples per class as the support set and Q examples per class as the query set. To clarify, the *N-way K-shot* few-shot learning setting refers to the task of making prediction over the query set, given a support set of $N \times K$ examples **during meta-testing**.

Framework Following prior works in ED [33], we add an additional *NULL* class in every task to indicate a *not-an-event* class. Thus, the FSL ED problem can be formulated as *N+1-way K-shot* few-shot classification problem. We employ the following general metric-based framework for FSL with two following components:

Instance Encoder: Given a sentence of N words $s = \{w_1, .., w_N\}$ and the position a of the trigger word $w_a \in s$ for some example/instance. We employ a deep neural network, denoted by a function f, to encode the instance into a fixed-dimension representation vector $f(s, a) \in R^d$.

Few-shot Classifier: A prototype is a representative vector c for each class appearing in the support set (called the prototype vector for the class). It can be an average [35] or a weighted sum with query-based attention weights [10] of vectors from the support set. Then, by computing the distance between the representation vector of a query instance $q = (s^q, a^q, t^q)$ and the prototype vectors, we can obtain a distance-based distribution over the possible classes in the current episode for q: $P(y = t^j|q, \mathcal{S}) = \frac{e^{-D(f(s^q,a^q),c^j)}}{\sum_{k=1}^{N+1} e^{-D(f(s^q,a^q),c^k)}}$, where D is a distance function (e.g. Euclidean distance [35], cosine similarity [38]), c^k is the prototype vector for the k-th class [35]. Given this distribution, the loss function L_{FSL} to train the FSL models is the negative log-likelihood computed for each query instance q: $L_{FSL} = -\log P(y = t^q|q, S)$

3 PROPOSED MODEL

Instance Encoder To differentiate between relevant words and irrelevant words, the instance encoder component in our model first focuses on relevant words in sentences to achieve this goal. As such, to identify the relevant words for an event trigger candidate in a sentence, we rely on the structure of the arguments of the trigger candidate where arguments have been shown to provide useful information to identify the event trigger [25]. In particular, we use the dependency parsing tree and their argument-related dependency paths to compute the representation vector for the trigger candidate. Given the sentence $s = w_1, w_2, \ldots, w_N$ and the trigger position a, we first embed s using the BERT model [6] to produce a representation vector h_i^0 for each word $w_i \in s$. Next, to induce hidden representation using the relevant words for the trigger, we build a pruned dependency graph following two steps:

Given a sentence, we first obtain its dependency tree. Then we convert it into an undirected graph by eliminating all directions and inserting self loops. This process results in a full dependency graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

Having a list of all entity mentions in the sentence, we find all the paths from the trigger candidate to the entity mention words. Then we eliminate all the edges of $\mathcal{G}$ that do not belong to any of the above paths, leading to a pruned dependency graph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$. Note that $\mathcal{G}$ and $\mathcal{G}'$ involve the same set of nodes for the words in the input sentence. For convenience, let A and A' be the adjacent matrices of the graphs $\mathcal{G}$ and $\mathcal{G}'$, respectively. In the next step, given the graphs $\mathcal{G}$ and $\mathcal{G}'$, we seek to induce abstract representation vectors for the nodes using GCNs [18]. As such, the GCN model in our work involve several hidden layers in which the representation vector of the i-th node/word at the l-th layer is computed as follow:

$$h_i^l(\mathcal{G}^{(\cdot)}) = \text{ReLU}(d_i^{-1}\Sigma_{j=1}^N A_{ij}^{(\cdot)} W^l h_j^{l-1} + b^l) \qquad (1)$$

where $(\cdot)$ indicate which graph (i.e., $\mathcal{G}$ or $\mathcal{G}'$) to be used, $d_i = \sum_{j=1}^N A_{ij}^{(\cdot)}$ is the degree of the node w_i, W^l, b^l are learnable parameters [18], and *ReLU* is the Rectified Linear Unit.

Finally, to embed the trigger candidate w_a into a representation vector, we concatenate the hidden vectors of the trigger candidate from BERT h_a^0 and all GCN layers $h_a^k(\mathcal{G}')(k > 0)$ (based on $\mathcal{G}'$), then feed it to an one-layer feed-forward neural network:

$$f(s, a) = v(\mathcal{G}') = W\tanh([h_a^0, h_a^1(\mathcal{G}'), \cdots, h_a^L(\mathcal{G}')]) + b \qquad (2)$$

where W, b are trainable parameters; L is the number of GCN layers. For convenience, the encoder with BERT and GCN as in Equation 2 is called the **BERTGCN** model to contrast with the **BERTMLP** model where $f(s, a)$ is only set to $Wh_a^0 + b$ (i.e., not using GCN model). Note that **BERTMLP** is also one of the current state-of-the-art models for ED [21].

Graph-based Regularization Our target is to regulate the representation learning based on dependency graphs, aiming to eliminate the contribution of irrelevant words. By introducing the pruned graphs, we have partially achieved this goal. However, irrelevant words might still contribute to the representation vectors in the model due to the BERT encoder that is run over the entire input sentence. To further constrain the contribution of irrelevant words for representation learning, we seek to impose a similarity requirement over representation vectors obtained via the pruned tree $\mathcal{G}'$ and the full tree $\mathcal{G}$. In other words, we ensure that adding irrelevant words in the pruned tree does not change representation vectors significantly.

To implement this idea, given the full dependency graph $\mathcal{G}$ and the pruned graph $\mathcal{G}'$, we first obtain two representation vectors V and V' for the input sentence s based on $\mathcal{G}$ and $\mathcal{G}'$ respectively via:

$$\begin{aligned} m^l(\mathcal{G}^{(\cdot)}) &= \max_i(h_1^l(\mathcal{G}^{(\cdot)}), \cdots, h_N^l(\mathcal{G}^{(\cdot)})) \\ V^{(\cdot)} &= \text{concat}(m^1(\mathcal{G}^{(\cdot)}), \cdots, m^L(\mathcal{G}^{(\cdot)})) \end{aligned} \tag{3}$$

In the next step, to limit the contribution of irrelevant words, we enforce the similarity between V and V' by adding the KL divergence, i.e., $L_{GRAPH} = KL(\sigma(V), \sigma(V'))$, between them into the overall loss function for minimization (σ is the softmax function to obtain distributions for the KL divergence).

Transfer Learning Our goal is to improve the generalization of the FSL ED model by transferring open-domain knowledge from WSD into the FSL ED model. Prior work on transfer learning for ED employs a matching method [26] which presents two separate neural networks with identical architecture and different parameters for ED and WSD. In each training iteration, a task is sampled and the model for that task is trained [26] using the cross-entropy loss (called **ALTERNATE** training). In addition, transfer learning is achieved by introducing an auxiliary loss to enforce the similarity between hidden vectors generated by the two models on the same sentences. However, directly applying this method for FSL might result in a drastic reduction of performance. First, the vectors generated by the two models might be mismatched due to the semantic difference of the tasks. Second, a significant difference between the learning speed of the two models requires manual calibration of learning rates during the training, leading to suboptimal solutions [12, 26]. This learning speed gap might be even more pronounced in FSL as FSL tends to converge faster than supervised learning. Finally, sharing an identical architecture might limit the robustness of WSD and ED models because the best model for a particular task cannot be employed. Therefore, we propose to separately pre-train the WSD model from the ED model that allows the WSD model to inherit the best WSD architecture to produce effective representations for sentences upfront. The ED model is trained afterward, acquiring the transferred knowledge from the WSD model. In this way, the learning rate gap issue is also automatically avoided to enhance the ED performance.

Formally, we employ two separate deep neural networks whose encoders are denoted as f_{ed} and f_{wsd} for ED and WSD, respectively. We have two datasets: $D_{ed} = \{(s_i^{ed}, a_i^{ed}, t_i^{ed})\}$ for ED and $D_{wsd} = \{(s_j^{wsd}, a_j^{wsd}, t_j^{wsd})\}$ where the notation of (s, a, t) are similar for two tasks [26]. They stand for a sentence s, the position a of a candidate anchor word in s, and the golden label t (i.e., an event type in ED and a word sense in WSD).

First, we train a WSD model using WSD data. The parameters of the trained WSD model will be fixed and its knowledge will be later transferred to the ED model:

$$f_{wsd}^* \leftarrow \underset{f_{wsd}}{\text{argmin}} \textstyle\sum_{(s,a,t)\in D_{wsd}} L(f_{wsd}(s,a), t)$$

Second, we train the ED model. In each ED training iteration, we sample an instance (s, a, t) from either D_{ed} or D_{wsd}, then feed it to the two model encoders to get two corresponding representations v^{ed} and v^{wsd} (using Equation 2). Finally transfer learning regularization from WSD to ED is performed by minimizing the KL divergence between v^{ed} and v^{wsd} (i.e., to promote the representation similarity over the same example (s, a)): $L_{WSD} = KL(\sigma(f_{ed}(s,a)), \sigma(f_{wsd}^*(s,a)))$. Finally, to train the proposed model, we minimize the combination of the proposed losses with α, β as two trade-off coefficients: $L = L_{FSL} + \alpha L_{WSD} + \beta L_{GRAPH}$.

4 EVALUATION

Datasets: We evaluate our methods on two ED datasets. First, as presented in the introduction, to enable a more realistic evaluation for FSL ED models, we employ the RAMS dataset (recently released by [7]) that provides human annotation for a large number of event types, involving 9124 examples/triggers for 139 event types. As RAMS is originally divided (for train/dev/test data portions) for traditional supervised learning, we first combine the data portions and re-split RAMS based on event types to facilitate FSL evaluation.

Second, to further evaluate the ED models in the traditional supervised learning setting, we utilize the widely used ACE-2005 dataset [39] that annotates 33 event subtypes. As discussed in [21], using the same data preprocessing is crucial for a fair comparison between methods on ACE-2005. To this end, we use the exact data split (i.e., train/dev/test) and data preprocessing provided by [21], the current state-of-the-art ED model for model evaluation on ACE-2005 in this work. Finally, we employ the **Semcor** dataset for WSD [28] (annotated with word senses in WordNet 3.0 [27]) to pre-train the WSD model for our transfer learning component.

Hyperparameters: We select the hyper-parameters for the proposed model based on the performance on the development set of RAMS. We employ the BERT-base-cased version of BERT and use the hidden vectors of the top $M = 4$ layers for the representation vectors h_i^0. For the GCN model, we stack $L = 2$ GCN layers; each has 512 hidden units. The dimensionality d of the representation vectors $f(s, a)$ for instances is set to 128. We use the state-of-the-art BERT-based WSD model in [13] to pre-train the WSD model for transfer learning in this work. Our FSL models are trained in 6000 episodes and tested with 500 episodes. The learning rate for FSL models is set to $2e10^{-4}$ with the Adam optimizer.

FSL setting: We evaluate all the models using the 5+1-way 5-shot FSL setting. As the previous study has observed that training FSL setting with a larger N^{train} results in better performance during testing [35], we sample $N^{train} = 20$ event subsubtypes in each training batch while still keeping $N^{test} = 5$ during test time.

Baseline: We consider two classes of baseline methods for FSL ED. The first class involves FSL methods that have been designed

Model	BERTMLP			BERTGCN		
	P	R	F	P	R	F
Prototypical	66.5	70.1	68.2	69.9	72.4	71.0
InterIntra	67.6	70.9	69.2	71.1	73.7	72.4
GraphTransfer (ours)	**68.9**	**70.6**	**69.7**	**71.9**	**74.7**	**73.2**

Table 1: Performance of FSL models with the 5+1-way 5-shot FSL on the RAMS test set.

Model	P	R	F
GraphTransfer (full)	**71.9**	**74.7**	**73.2**
-WSD	71.4	74.2	72.7
-GRAPH	70.8	73.5	72.1
-GRAPH-WSD	69.9	72.4	71.0
-GRAPH-WSD-Prune	69.1	72.6	70.7
-FIX (using ALTERNATE)	71.8	73.3	72.5

Table 2: Ablation study on RAMS dataset

Model	RAMS			ACE-2005		
	P	R	F	P	R	F
DMBERT	62.6	44.0	51.7	79.1	71.3	74.9
BERTMLP	62.4	49.3	55.0	77.8	74.6	76.2
BERTGCN	**66.5**	59.0	62.5	80.2	74.8	77.4
Gated-GCN	64.8	64.5	64.7	78.8	76.3	77.6
GraphTransfer (ours)	66.3	**65.8**	**66.1**	**80.3**	**78.0**	**79.1**

Table 3: Supervised learning performance.

for other NLP tasks, including matching networks [38], prototypical networks [35], hybrid-attention prototypical networks [10], and relation networks [36]. Among these methods, the prototypical network (called **Prototypical**) produces the best performance in our experiments and we will use it to represent the first class of baselines in this work. Note that the selection of prototypical networks will also determine the distance function D in Equation 2. Second, we also utilize **InterIntra**, the current state-of-the-art technique for FSL ED in [20] as the baseline. Finally, we examine both **BERTMLP** and **BERTGCN** as the instance encoders for FSL models in this work.

4.1 Few-Shot Learning Evaluation

Result: Table 1 compares the baseline FSL models without proposed method (called GraphTransfer) on the RAMS test set. The first observation is that the GCN-based encoder BERTGCN is significantly better than the non-graph encoder BERTMLP across different FSL methods, thus highlighting the benefits of GCN for FSL ED. More importantly, the proposed model significantly outperforms all the baseline models with $p < 0.05$. The consistent improvement for both instance encoder architectures demonstrates the effectiveness of the proposed FSL models for ED in this work.

Ablation study: Our proposed method GraphTransfer involves two main components: (i) transferring learned knowledge from pre-trained WSD task (**WSD**) and (ii) graph-based regularization (**GRAPH**). We also propose the fix training strategy, called **FIX**, to pre-train the WSD model for transfer learning (i.e., in contrast to the ALTERNATE method in [26]), and the use of relevant words derived from the pruned graph for prediction (**Prune**). To analyze the contribution of these components, we incrementally remove these components from the full model and reevaluate the remaining models. Note that by eliminating the **WSD** component, we also exclude the **FIX** strategy due to their dependency.

Table 2 presents the performance of 5+1-way 5-shot few-shot learning on RAMS. As shown in the table, eliminating either **WSD** or **GRAPH** significantly hurts the performance of the model. In addition, the performance is further reduced when the full dependency graph is used to compute the instance representations (i.e., instead of using the pruned graph equation 1). Finally, excluding the **FIX** training strategy in transfer learning (i.e., using ALTERNATE in [26] instead) also leads to significantly reduced performance.

4.2 Supervised Learning Evaluation

Baseline: We compare our proposed model against current state-of-the-art models for ED in the supervised learning setting on the ACE-2005 dataset, including **DMBERT** [40] (a BERT-based model with dynamic pooling), **BERTGCN** (as presented above), and **BERTMLP** and **Gated-GCN** [21]. Note that **Gated-GCN** also uses BERT and it is the current state-of-the-art ED model for supervised learning with our dataset setting on ACE-2005. For completeness, we also provide Gate-GCN 's performance on RAMS in the supervised learning setting using its original data split.

Result: Table 3 reports the performance of the models. It is clear from the table that the proposed model significantly outperforms all baseline models with large margins over the current best model, i.e., 3.6% on RAMS, and 1.5% on ACE-2005, thereby further confirming the effectiveness of the proposed model for ED.

5 RELATED WORK

Early studies have addressed ED via the supervised learning setting [1, 4, 8, 14, 17, 23, 29–33]. Extending ED to unseen event types is an emerging direction for which several approaches have been proposed, including bootstrapping [16], self-training [24], zero-shot learning [15], distant supervision [3, 37], and FSL [19, 20]. FSL promotes effective learning from small numbers of examples for new types. The major approaches include metric learning [5, 10, 35, 36, 38] and meta-learning [9, 22]. Finally, several studies have employed transfer learning for few-shot learning [2, 34]; however, none of the has explored transfer learning for FSL ED as we do.

6 CONCLUSION

We present how transferring open-domain knowledge from word sense disambiguation and regulating representation based on pruned dependency graphs can improve few-shot learning for ED on large-scale datasets. Our proposed model achieves state-of-the-art performance on both few-shot learning and supervised learning on two ED datasets. In the future, we plan to explore other types of knowledge for transfer learning to improve the models in this work.

ACKNOWLEDGMENTS

This research has been supported by the Army Research Office (ARO) grant W911NF-21-1-0112 and the NSF grant CNS-1747798 to the IUCRC Center for Big Learning. This research is also based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA Contract No. 2019-19051600006 under the Better Extraction from Text Towards Enhanced Retrieval (BETTER) Program. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ARO, NSF, ODNI, IARPA, the Department of Defense, or the U.S. Government.

REFERENCES

[1] David Ahn. 2006. The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*.

[2] Yujia Bao, Menghua Wu, Shiyu Chang, and Regina Barzilay. 2020. Few-shot Text Classification with Distributional Signatures. In *ICLR*.

[3] Yubo Chen, Shulin Liu, Xiang Zhang, Kang Liu, and Jun Zhao. 2017. Automatically Labeled Data Generation for Large Scale Event Extraction. In *ACL*.

[4] Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *ACL-IJCNLP*.

[5] Shumin Deng, Ningyu Zhang, Jiaojian Kang, Yichi Zhang, Wei Zhang, and Huajun Chen. 2020. Meta-Learning with Dynamic-Memory-Based Prototypical Network for Few-Shot Event Detection. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 151–159.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL-HTL* (2019).

[7] Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-Sentence Argument Linking. In *ACL*.

[8] Xiaocheng Feng, Lifu Huang, Duyu Tang, Heng Ji, Bing Qin, and Ting Liu. 2016. A Language-Independent Neural Network for Event Detection. In *ACL (Volume 2: Short Papers)*, Vol. 2. 66–71.

[9] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. 1126–1135.

[10] Tianyu Gao, Xu Han, Zhiyuan Liu, and Maosong Sun. 2019. Hybrid attention-based prototypical networks for noisy few-shot relation classification. In *AAAI*.

[11] Ralph Grishman, David Westbrook, and Adam Meyers. 2005. NYU's English ACE 2005 System Description. In *ACE 2005 Evaluation Workshop*.

[12] Jiang Guo, Wanxiang Che, Haifeng Wang, Ting Liu, and Jun Xu. 2016. A unified architecture for semantic role labeling and relation classification. In *COLING*. 1264–1274.

[13] Christian Hadiwinoto, Hwee Tou Ng, and Wee Chung Gan. 2019. Improved word sense disambiguation using pre-trained contextualized word representations (to appear). In *EMNLP-IJCNLP*.

[14] Yu Hong, Jianfeng Zhang, Bin Ma, Jianmin Yao, Guodong Zhou, and Qiaoming Zhu. 2011. Using Cross-Entity Inference to Improve Event Extraction. In *ACL*.

[15] Lifu Huang, Heng Ji, Kyunghyun Cho, and Clare R Voss. 2018. Zero-Shot Transfer Learning for Event Extraction. In *ACL*. 2160–2170.

[16] Ruihong Huang and Ellen Riloff. 2012. Modeling Textual Cohesion for Event Extraction. In *AAAI*.

[17] Heng Ji and Ralph Grishman. 2008. Refining Event Extraction Through Cross-document Inference. In *ACL*.

[18] Thomas N. Kipf and Max Welling. 2017. Semi-supervised Classification with Graph Convolutional Networks. In *ICLR*.

[19] Viet Dac Lai, Franck Dernoncourt, and Thien Huu Nguyen. 2020. Exploiting the Matching Information in the Support Set for Few Shot Event Classification. In *Proceedings of the 24th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*.

[20] Viet Dac Lai, Thien Huu Nguyen, and Frank Dernoncourt. 2020. Extensively Matching for Few-shot Learning Event Detection. In *Proceedings of the First Joint Workshop on Narrative Understanding, Storylines, and Events*. 38–45.

[21] Viet Dac Lai, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2020. Event Detection: Gate Diversity and Syntactic Importance Scores for Graph Convolution Neural Networks. In *EMNLP*.

[22] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. 2019. Meta-learning with differentiable convex optimization. In *CVPR*. 10657–10665.

[23] Shasha Liao and Ralph Grishman. 2010. Using Document Level Cross-Event Inference to Improve Event Extraction. In *ACL*.

[24] Shasha Liao and Ralph Grishman. 2011. Acquiring Topic Features to improve Event Extraction: in Pre-selected and Balanced Collections. In *RANLP*.

[25] Shulin Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2017. Exploiting Argument Information to Improve Event Detection via Supervised Attention Mechanisms. In *ACL*.

[26] Weiyi Lu and Thien Huu Nguyen. 2018. Similar but not the Same: Word Sense Disambiguation Improves Event Detection via Neural Representation Matching. In *EMNLP*.

[27] George A. Miller. 1995. WordNet: a Lexical Database for English. In *Communications of the ACM, 38(11):39–41*.

[28] George A Miller, Martin Chodorow, Shari Landes, Claudia Leacock, and Robert G Thomas. 1994. Using a semantic concordance for sense identification. In *Proceedings of the Workshop on Human Language Technology*.

[29] Minh Van Nguyen, Viet Dac Lai, and Thien Huu Nguyen. 2021. Cross-Task Instance Representation Interactions and Label Dependencies for Joint Information Extraction with Graph Convolutional Networks. In *NAACL-HLT*.

[30] Thien Nguyen and Ralph Grishman. 2018. Graph Convolutional Networks With Argument-Aware Pooling for Event Detection. In *AAAI*.

[31] Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint Event Extraction via Recurrent Neural Networks. In *NAACL*.

[32] Thien Huu Nguyen, Lisheng Fu, Kyunghyun Cho, and Ralph Grishman. 2016. A Two-stage Approach for Extending Event Detection to New Types via Neural Networks. In *Proceedings of the 1st ACL Workshop on Representation Learning for NLP (RepL4NLP)*.

[33] Thien Huu Nguyen and Ralph Grishman. 2015. Event Detection and Domain Adaptation with Convolutional Neural Networks. In *ACL-IJCNLP*.

[34] Igor Shalyminov, Sungjin Lee, Arash Eshghi, and Oliver Lemon. 2019. Few-Shot Dialogue Generation Without Annotated Data: A Transfer Learning Approach. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*. 32–39.

[35] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *NIPS*.

[36] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *CVPR*.

[37] Meihan Tong, Bin Xu, Shuai Wang, Yixin Cao, Lei Hou, Juanzi Li, and Jun Xie. 2020. Improving Event Detection via Open-domain Trigger Knowledge. In *ACL*. 5887–5897.

[38] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. In *NIPS*.

[39] Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. ACE 2005 multilingual training corpus. In *Technical report, Linguistic Data Consortium*.

[40] Xiaozhi Wang, Xu Han, Zhiyuan Liu, Maosong Sun, and Peng Li. 2019. Adversarial training for weakly supervised event detection. In *NAACL-HLT*. 998–1008.

[41] Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. 2019. Exploring Pre-trained Language Models for Event Extraction and Generation. In *ACL*. 5284–5294.