A Two-stage Approach for Extending Event Detection to New Types via Neural Networks

Thien Huu Nguyen, Lisheng Fu, Kyunghyun Cho and Ralph Grishman
Computer Science Department, New York University, New York, NY 10003, USA
thien@cs.nyu.edu lisheng@cs.nyu.edu
kyunghyun.cho@nyu.edu grishman@cs.nyu.edu

Abstract

We study the event detection problem in the new type extension setting. In particular, our task involves identifying the event instances of a target type that is only specified by a small set of seed instances in text. We want to exploit the large amount of training data available for the other event types to improve the performance of this task. We compare the convolutional neural network model and the feature-based method in this type extension setting to investigate their effectiveness. In addition, we propose a two-stage training algorithm for neural networks that effectively transfers knowledge from the other event types to the target type. The experimental results show that the proposed algorithm outperforms strong baselines for this task.

1 Introduction

Event detection (ED) is an important task of information extraction that seeks to locate instances of events with some types in text. Each event mention is associated with a phrase, the event trigger[1], which evokes that event. Our task, more precisely stated, involves identifying event triggers of some types of interest. For instance, in the sentence "*A* ***cameramen*** *was* ***shot*** *in* ***Texas today***", an ED system should be able to recognize the word "*shot*" as a trigger for the event "*Attack*". ED is a crucial component in the overall task of event extraction, which also involves event argument discovery.

There have been two major approaches to ED in the literature. The first approach extensively leverages linguistic analysis and knowledge resources to capture the *discrete* structures for ED, focusing on the combination of various properties such as lexicon, syntax, and gazetteers. This is called the feature-based approach that has dominated the ED research in the last decade (Ji and Grishman, 2008; Gupta and Ji, 2009; Liao and Grishman, 2011; McClosky et al., 2011; Riedel and McCallum, 2011; Li et al., 2013; Venugopal et al., 2014). The second approach, on the other hand, is proposed very recently and uses convolutional neural networks (CNN) to exploit the *continuous* representations of words. These continuous representations have been shown to effectively capture the underlying structures of a sentence, thereby significantly improving the performance for ED (Nguyen and Grishman, 2015; Chen et al., 2015).

The previous research has mainly focused on building an ED system in a supervised setting. The performance of such systems strongly depends on a sufficient amount of labeled instances for each event type in the training data. Unfortunately, this setting does not reflect the real world situation very well. In practice, we often have a large amount of training data for some old event types but are interested in extracting instances of a new event type. The new event type is only specified by a small set of seed instances provided by clients (the event type extension setting). How can we effectively leverage the training data of old event types to facilitate the extraction of the new event type?

Inspired by the work on transfer learning and domain adaptation (Blitzer et al., 2006; Jiang and Zhai, 2007; Daume III, 2007; Jiang, 2009), in this paper, we systematically evaluate the representative methods (i.e, the feature based model and the CNN model) for ED to gain an insight into which kind of method performs better in the new extension setting. In addition, we propose a two-stage algorithm to train a CNN model that effectively learns and transfers the knowledge from the old event types for the extraction of the target type.

[1] most often a single verb or nominalization

Proceedings of the 1st Workshop on Representation Learning for NLP, pages 158–165,
Berlin, Germany, August 11th, 2016.

The experimental results show that this two-stage algorithm significantly outperforms the traditional methods in the type extension setting for ED and demonstrates the benefit of CNN in transfer learning. To our knowledge, this is the first work on the type extension setting as well as on transferring knowledge with neural networks for ED of natural language processing.

2 Task Definition

The event type extension setting in this work is as follow. We are given a document set D annotated for a large set D_A of trigger words (positive instances) of some event types (the *auxiliary* types, denoted by A). However, we are interested in extracting trigger words of a new event type T (the *target* type, $T \notin A$) that is only specified by a small annotated set D_T of positive instances (the seeds) in D. Note that while D_A involves all the positive instances of the auxiliary types, D_T might only be partial and not necessarily include all the trigger words of type T in D.

Also, we call D_N the set of the negative instances generated from D under this setting (to be discussed in more details later). In general, D_N might contains unannotated trigger words of T (false negatives), making this task more challenging. Eventually, our goal is to *learn an event detector for T, leveraging the training data D_T, D_A and D_N* for both the target and auxiliary types. Note that our work is related to Jiang (2009) who studies the relation type extension problem.

3 Models for Event Detection

In this section, we first present the representative approaches for ED. The two-stage algorithm will be discussed in the next section.

We treat the event detection problem for the target type T as a binary classification problem. For every token in a given sentence, we want to predict if the current token is an event trigger of type T or not? The current token along with its context in the sentence constitute an event trigger candidate or an example in the binary classification terms.

3.1 The Feature-based Model

In the feature-based model (denoted by FET), the event trigger candidates are first transformed into rich feature vectors to encapsulate linguistically useful properties for ED. These vectors are then fed into a statistical classifier such as maximum entropy (MaxEnt) and classified as the type T or not. In this work, we employ the feature set for ED from (Li et al., 2013), which is the state-of-the-art FET.

3.2 The Convolutional Neural Networks

In a CNN for ED, we limit the context of the trigger candidates to a fixed window size by trimming longer sentences and padding shorter sentences with a special token when necessary. Let $2w + 1$ be the fixed window size, and $x = [x_{-w}, x_{-w+1}, \ldots, x_0, \ldots, x_{w-1}, x_w]$ be some trigger candidate where the current token is positioned in the middle of the window (token x_0). Before entering CNN, each token x_i is transformed into a real-valued vector $\mathbf{x}_i$ by concatenating the continuous look-up vectors from the following tables:

1. Word Embedding Table E (Turian et al., 2010; Mikolov et al., 2013a; Mikolov et al., 2013b).

2. Position Embedding Table: to embed the relative distance i of x_i to the current token x_0.

3. Entity Type Embedding Table: to capture the entity type information for each token. Following Nguyen and Grishman (2015), we assign the entity type labels to each token using the heads of the entity mentions in x with the BIO schema.

As a result, the original event trigger candidate x is transformed into a matrix $\mathbf{x} = [\mathbf{x}_{-w}, \mathbf{x}_{-w+1}, \ldots, \mathbf{x}_0, \ldots, \mathbf{x}_{w-1}, \mathbf{x}_w]$. This matrix will serve as the input for CNN.

For CNN, the matrix $\mathbf{x}$ is first passed through a convolution layer and then a max pooling layer to compute the global representation vector R_C for the trigger candidate x (Nguyen and Grishman, 2015). In addition, we obtain the local representation vector R_L by concatenating the embedding vectors of the words in a window size $2d + 1$ of x_0, motivated by the models in Chen et al. (2015):

$$R_L = [E[x_{-d}], \ldots, E[x_0], \ldots, E[x_d]]$$

Finally, the concatenation of the global and local vectors R_C and R_L is used as the input for a feed-forward neural network with a softmax layer in the end to perform trigger identification for T. Note that our CNN model is similar to (Nguyen and Grishman, 2015) and applies multiple window sizes for the feature maps in the convolution layer.

4 Event Type Extension Systems

4.1 The Baseline Systems

For each of the two models presented above (i.e, FET and CNN), we have two baseline mechanisms to train an event detector for T (Jiang, 2009). In the first baseline (denoted by TARGET), we use the small instance set D_T of the target type T together with the negative instances in D_N to train a binary classifier for T. In the second baseline (denoted by UNION), we combine the positive instances in both D_T and D_A as well as the negative instances in D_N to train a binary classifier for T.

Eventually, we have 4 baseline systems corresponding to the two choices of models (i.e, FET, CNN) and the two choices of the training mechanisms (i.e, TARGET, UNION). We denote these four baselines by: FET-TARGET, FET-UNION, CNN-TARGET, and CNN-UNION.

4.2 Hypothesis About the Baselines

The underlying assumption of transfer learning for type extension is the existence of the general features that are effective for prediction across different types (Jiang, 2009). The performance of a model for a given target type, thus, depends on two factors: (i) how well the model identifies and quantifies general features, and (ii) how effectively the model transfers the knowledge about the general features and adapt it to the target type.

Hypothesis: the UNION training mechanism is more effective than TARGET when the number of seed instances of the target type is small. The reason originates from the inclusion of the training data D_A of the auxiliary types in UNION that would provide more evidences to estimate the importance of the general features better (factor (i)).

4.3 The Two-stage Algorithm

Although UNION can help to learn the general features, its major limitation lies in the lack of the directing mechanisms to make the model specific to the target type (factor (ii)). Essentially, UNION treats the positive instances of the target and auxiliary types similarly, making it more about a general purpose event detector rather than a specific detector for the target type. Therefore, we propose to consider the positive instances of the target D_T and the auxiliary types D_A in two separate stages.

In the first stage, a large amount of the training data D_A of the auxiliary types are used by a CNN to learn the general feature extractors across event types. In the second stage, the seed instances of the target type in D_T are used to adapt the models to the target type. In order to transfer the knowledge from the auxiliary types to the target type between these two stages, we propose to utilize a CNN that facilitates the transferring process via the weight initialization. The two-stage algorithm (CNN-2-STAGE) is presented below.

Algorithm 1: CNN-2-STAGE

```
Input  : $D_T$, $D_A$ and $D_N$
Output: An event detector for $T$
Stage I: Train a CNN model on $D_A \cup D_N$ with
  randomly initialized weight matrices and embedding.
Let $P$ be the set of weight matrices and embedding
  tables after the training process of CNN in stage I.
Stage II: Train a CNN on $D_T \cup D_N$ with the weight
  matrices and embedding tables initialized by the
  corresponding elements in $P$.
Return the CNN model trained in Stage II
```

Note that similar to stage I of the algorithm and previous work on neural networks (Nguyen and Grishman, 2015; Chen et al., 2015), the weight matrices and embedding tables are also initialized randomly in the training mechanisms UNION and TARGET. The only exception is the word embedding table that is pre-trained on a large corpus for UNION, TARGET as well as the stage I.

All the weight matrices and embedding tables are optimized during training (for UNION, TARGET as well as CNN-2-STAGE) to achieve the optimal state. This is especially important in Stage II of CNN-2-STAGE as it helps to adapt the general feature extractors in Stage I to the target type T.

5 Training

Following Nguyen and Grishman (2015), we train the NN models using stochastic gradient descent with shuffled mini-batches, the AdaDelta update rule, back-propagation and dropout. Finally, we rescale the weights whose l_2-norms exceed a predefined threshold.

6 Experiments

6.1 Parameters and Resources

For all the experiments below, we utilize the pre-trained word embeddings `word2vec` (300 dimensions) from Mikolov et al. (2013a) to initialize the word embedding table. The parameters for CNN and training the network are inherited from the previous studies, i.e, the fixed window size $w = 15$, the window size set for feature maps

= $\{2, 3, 4, 5\}$, 150 feature maps for each window size, 50 dimensions for all the embedding tables (except the word embedding table), the dropout rate $= 0.5$, the mini-batch size $= 50$, the hyperparameter for the l_2 norms = 3 and the window for local context $d = 5$ (Nguyen and Grishman, 2015; Chen et al., 2015).

6.2 Dataset and Settings

Following the previous work (Li et al., 2013; Chen et al., 2015; Nguyen and Grishman, 2015), we consider the ED task of the 2005 Automatic Context Extraction (ACE) evaluation that annotates 8 event types and 33 event subtypes [2]. As the numbers of event mentions (triggers) for each subtype in ACE are small, in this work, we focus on the extraction of the event types: "*Life*", "*Movement*", "*Transaction*", "*Business*", "*Conflict*", "*Contact*", "*Personell*", and "*Justice*". We remove the event triggers of types "*Transaction*" and "*Business*" due to their small numbers of occurrences, resulting in the dataset with six remaining event types (denoted from 1 to 6).

In the experiments, we use the same data split in Li et al. (2013) with 40 newswire documents as a test set, 30 other documents as a development set and the 529 remaining documents as a training set. Note that the training documents correspond to our original dataset D above. Let P_i be the positive instance set of the type i in D ($i = 1$ to 6).

We take each event type i as the target type T and treat the other 5 types as the auxiliary types, constituting 6 sets of experiments. In each set of experiments for a target type i (T), we randomly select S positive instances of T for the seed set D_T ($S = |D_T|$) and *treat the remaining target instances $P_i \setminus D_T$ as negative*. Note that this essentially introduces false negatives into the training data and makes the task more challenging.

In order to deal with false negatives, we remove all the sentences that do not contain any events in the original dataset D. In this way, we remove a large number of true negatives along with a fraction of the false negatives, leading to the reduced dataset D'. We do the experiments on D' with:

$$D_A = \bigcup_{j=1(j \neq i)}^{6} P_j$$

$$D_N = D' \setminus D_T \setminus D_A$$

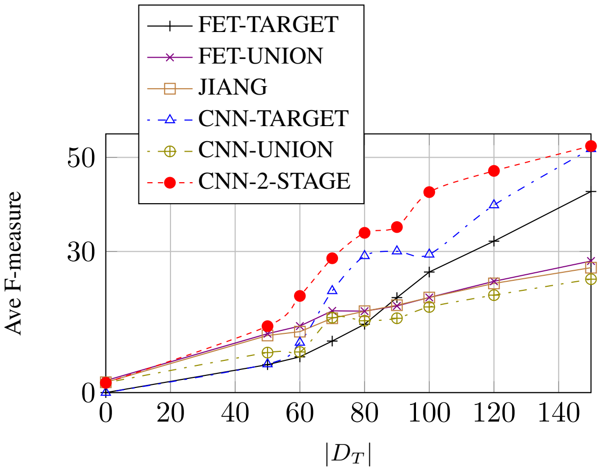

Figure 1: Average F measures vs $|D_T|$.

We note that (Jiang, 2009) uses a different setting in training where she removes all the remaining target instances $P_i \setminus D_T$ directly. In our opinion, this is unrealistic as it assumes the label of the instances in $P_i \setminus D_T$ while we are only provided with the label of the seed set D_T in practice.

Finally, similar to (Jiang, 2009), we remove the positive instances of the auxiliary event types from the test set to concentrate on the classification accuracy for the target type. We also remove all the positive instances of the target type in the development set to make it more realistic.

6.3 Evaluation

This section compares the four baseline models in Section 4.1 with the proposed two-stage model CNN-2-STAGE. For completeness, we also evaluate the transfer learning model in Jiang (2009), adapted to the event type extension task (called JIANG). For JIANG, we apply the automatic feature separation method as the general syntactic patterns and type constraints for relation in Jiang (2009) are not applicable to our ED task.

For each described model, we perform six sets of experiments in Section 6.2, where the number of seed instances $|D_T|$ is varied from 0 to 150. We then report the *average F-scores* of the six experiment sets for each value of S. Figure 1 shows the curves.

Assuming the same kind of model (i.e, either FET or CNN), we see that UNION is better than TARGET when $|D_T|$ is small, confirming our hypothesis in Section 4.2. This demonstrates the benefit of UNION and the training data D_A of the auxiliary types when there are not enough training

[2] https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-events-guidelines-v5.4.3.pdf

Target Type	FET TARGET	FET UNION	JIANG	CNN TARGET	CNN UNION	CNN 2-STAGE
Movement	**21.8**	9.2	9.6	4.1	4.0	19.7
Personnel	19.4	15.8	17.3	27.3	16.4	**40.5**
Conflict	12.8	18.0	17.9	12.8	29.8	**43.0**
Contact	45.4	35.7	34.6	**62.5**	19.2	54.6
Life	29.8	21.8	22.5	22.2	24.7	**50.0**
Justice	24.6	20.9	19.4	47.4	15.3	**48.0**
Average	25.6	20.2	20.2	29.4	18.2	**42.6**

Table 1: System Performance

Event Type	Examples
Personnel	*Georgia* ***fired*** *football coach Jim Donnan Monday after a disappointing 7-4 season ...* *The bad doctors are* ***removed*** *from the practice of medicine.*
Conflict	*U.S. forces continued to* ***bomb*** *Fallujah.* *Israel retaliated with rocket* ***attacks*** *and terrorists* ***blew*** *a hole in a United States warship in Yemen.* *Protesters* ***rallied*** *on the White House lawn.*
Life	*... and two Israeli soldiers were* ***wounded****, one critically.* *Witnesses said the soldiers responded by firing tear gas and rubber bullets, which led to ten demonstrators being* ***injured****.* *John Hinckley attempted to* ***assassinate*** *Ronald Reagan.*
Justice	*Since May, Russia has* ***jailed*** *over 20 suspected terrorists without a trial.* *A judicial source said today, Friday, that five Croatians were* ***arrested*** *last Tuesday during an operation ...*

Table 2: Examples for the trigger words with the latent semantic. The trigger words are underlined.

instances for T. However, when we are provided with more seed instances for the target type (i.e, $|D_T|$ becomes larger), TARGET turns out to be significantly better than UNION.

We also observe that CNN outperforms FET in the TARGET mechanism. This is consistent with the previous studies for ED (Nguyen and Grishman, 2015). However, in the UNION mechanism, CNN is less effective than FET, suggesting that UNION is not a good mechanism to transfer knowledge in CNN.

We do not see much performance improvement of JIANG over FET-UNION. This can be explained by the lack of explicit linguistic guidance (i.e, the syntactic patterns and type constraints) for the general features in the event extension task that are crucial to the success of the model in Jiang (2009).

Finally and most importantly, we see that the two-stage model CNN-2-STAGE outperforms all the compared models regardless of $|D_T|$. This is significant when $|D_T|$ is greater than 50. These results suggest the effectiveness of the two-stage training algorithm on transferring knowledge from the auxiliary types to the target type for CNN.

6.4 Analysis

In order to further understand the systems on the separate event types, Table 1 presents the performance of the compared systems for the six experiment sets in Section 6.2 (corresponding to the 6 different choices of the event target type T in the dataset) when S is set to 100.

One of the most important observations from the table is that CNN-2-STAGE is significantly better than JIANG, CNN-TARGET and CNN-UNION on five target types (i.e, Y = {*Movement*, *Personnel*, *Conflict*, *Life*, *Justice*})[3] and only worse than CNN-TARGET on the *Contact* type. This raises a question on the distinction between *Contact* and the other event types in Y that affects the transferring effectiveness of CNN-2-STAGE. Also, what is the common feature of the event types in Y that helps CNN-2-STAGE successfully transfers knowledge between them?

The key insight of our system output analysis is the shared latent semantic among a large por-

[3] Although it is less pronounced for *Justice*

Even Type	Event Subtypes	Most Frequent Triggers
Contact	*Meet*, *Phone-Write*	*meeting*, *talks*, *meet*, *call*, *summit*, *meetings*, *met*, *letters*, *talked*, *conference*
Movement	*Transport*	*go*, *come*, *arrived*, *get*, *trip*, *leave*, *went*, *moving*, *moved*, *take*
Personnel	*Start-Position*, *End-Position*, *Nominate*, *Elect*	*election*, *elections*, *former*, *elected*, *appointed*, *resigned*, *fired*, *retired*, *won*, *leaving*
Conflict	*Attack*, *Demonstrate*	*war*, *attack*, *fighting*, *attacks*, *fire*, *bombing*, *fight*, *hit*, *combat*, *shot*
Life	*Be-Born*, *Marry*, *Divorce*, *Injure*, *Die*	*killed*, *death*, *died*, *suicide*, *injured*, *dead*, *killing*, *divorce*, *married*, *die*
Justice	*Arrest-Jail*, *Release-Parole*, *Trial-Hearing*, *Charge-Indict*, *Sue*, *Convict*, *Sentence*, *Fine*, *Execute*, *Extradite*, *Acquit*, *Appeal*, *Pardon*	*trial*, *convicted*, *sentence*, *charges*, *arrested*, *appeal*, *sentenced*, *charged*, *sued*, *parole*

Table 3: Event types, subtypes and the most frequent trigger words.

tion of trigger words of the four event types in $Y\setminus \{\textit{Movement}\}$. In particular, all the four event types in $Y\setminus \{\textit{Movement}\}$ includes trigger words that induce some level of conflict between their subjects and objects. These conflicts are often manifested by some physical and irritating actions between the two engaged parties. Some examples of the trigger words with the latent semantic for the event types in $Y\setminus \{\textit{Movement}\}$ are given in Table 2[4]. This latent semantic is first captured by word embeddings and CNN in Stage I of CNN-2-STAGE, and then transferred to the target type in Stage II. The feature-based transfer learning systems like JIANG, on the other hand, cannot encode such latent semantics effectively as they rely on the discrete features with the symbolic representation of words.

In the ACE 2005 corpus, the event type *Movement* only has one subtype of *Transport* which mainly focuses on the transportation of weapons, vehicles or people. The context of the trigger words of the subtype *Transport* often involves the military or struggling objects such as soldiers, Iraq, forces etc. These context words are similar to those of the trigger words of the types *Conflict* and *Life*. As a result, the CNN-2-STAGE algorithm can learn these general features from the trigger words of *Conflict* and *Life*, and then transfer them to improve the extraction of *Movement*. We show some examples of *Movement* below:

1. *After today's air strikes, 13 Iraqi soldiers* **abandoned** *their posts and surrendered to Kurdish fighters.*
2. *The convoy was* **escorted** *by U.S. soldiers.*
3. *Israeli forces* **moved** *into Hebron's Al-Sheikh district where his family lived . . .*

Finally, regarding the event type *Contact*, it occurs when two or more entities engage in discussion either directly or remotely[5]. The purpose of such discussions are often about information or opinion exchange rather than a mean to express discussions or conflicts with irritating actions (as the event types in Y do). This divergence between *Contact* and Y leads to the poor quality of the general features learnt by the transfer learning methods (i.e, JIANG and CNN-2-STAGE), eventually degrading their performances. Some examples of the *Contact* event type are given below:

1. *People can* **communicate** *with international friends without the hefty phone bills.*
2. *I'm chewing gum and* **talking** *on the phone while writing this note.*
3. *Mr. Erekat is due to travel to Washington to* **meet** *with US Secretary of State Madeleine Albright and other US officials . . .*

In order to further demonstrate the difference between *Contact* and the other event types, Table 3 enumerates the event subtypes and the most frequent trigger words for each event. The event subtypes in Table 3 can be considered as the concepts or topics covered by the corresponding event types in the ACE 2005 corpus. As we can see from the

[4] Taken from the ACE 2005 Annotation Guideline

[5] Defined by the annotation guideline.

table, the *Meet* and *Phone-Write* subtypes or topics of *Contact* are quite separate from those of the other types.

7 Related Work

Early research on event extraction has primarily focused on local sentence-level representations in a pipelined architecture (Grishman et al., 2005; Ahn, 2006). Afterward, higher level features have been found to improve the performance (Ji and Grishman, 2008; Gupta and Ji, 2009; Patwardhan and Riloff, 2009; Liao and Grishman, 2010; Liao and Grishman, 2011; Hong et al., 2011; McClosky et al., 2011; Huang and Riloff, 2012; Li et al., 2013). Some recent research has proposed joint models for EE, including the methods based on Markov Logic Networks (Riedel et al., 2009; Poon and Vanderwende, 2010; Venugopal et al., 2014), structured perceptron (Li et al., 2013; Li et al., 2014b), and dual decomposition (Riedel et al. (2009; 2011b)).

The application of neural networks to EE is very recent. In particular, Zhou et al. (2014) and Boros et al. (2014) use neural networks to learn word embeddings from a corpus of specific domains and then directly utilize these embeddings as features in statistical classifiers. Chen et al. (2015) apply dynamic multi-pooling CNNs for EE in a pipelined framework, while Nguyen et al. (2016) propose joint event extraction using recurrent neural networks.

Finally, domain adaptation and transfer learning have been studied extensively for various NLP tasks, including part of speech tagging (Blitzer et al., 2006), name tagging (Daume III, 2007), parsing (McClosky et al., 2010), relation extraction (Plank and Moschitti, 2013; Nguyen and Grishman, 2014; Nguyen et al., 2015a), to name a few. For event extraction, Miwa et al. (2013) study instance weighting and stacking models while Riedel and McCallum (2011b) examine joint models with domain adaptation. However, none of them studies the new type extension setting for ED using neural networks like we do.

8 Conclusion

We systematically evaluate the ED models on the new type extension setting. A two-stage algorithm to train the CNN model and transfer knowledge is introduced, yielding the state-of-the-art performance for the extension task. In the future, we plan to apply the two-stage algorithm to other tasks such as relation extension to further verify its effectiveness.

References

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

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *EMNLP*.

Emanuela Boros, Romaric Besançon, Olivier Ferret, and Brigitte Grau. 2014. Event role extraction using domain-relevant word representations. In *EMNLP*.

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

Hal Daume III. 2007. Frustratingly easy domain adaptation. In *ACL*.

Ralph Grishman, David Westbrook, and Adam Meyers. 2005. Nyus english ace 2005 system description. In *ACE 2005 Evaluation Workshop*.

Prashant Gupta and Heng Ji. 2009. Predicting unknown time arguments based on cross-event propagation. In *ACL-IJCNLP*.

Yu Hong, Jianfeng Zhang, Bin Ma, Jianmin Yao, Guodong Zhou, and Qiaoming Zhu. 2011. Using cross-entity inference to improve event extraction. In *ACL*.

Ruihong Huang and Ellen Riloff. 2012. Modeling textual cohesion for event extraction. In *AAAI*.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *ACL*.

Jing Jiang and ChengXiang Zhai. 2007. A two-stage approach to domain adaptation for statistical classfiers. In *CIKM*.

Jing Jiang. 2009. Multi-task transfer learning for weakly-supervised relation extraction. In *ACL-IJCNLP*.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *ACL*.

Qi Li, Heng Ji, Yu Hong, and Sujian Li. 2014b. Constructing information networks using one single model. In *EMNLP*.

Shasha Liao and Ralph Grishman. 2010. Using document level cross-event inference to improve event extraction. In *ACL*.

Shasha Liao and Ralph Grishman. 2011. Acquiring topic features to improve event extraction: in preselected and balanced collections. In *RANLP*.

David McClosky, Eugene Charniak, , and Mark Johnson. 2010. Automatic domain adaptation for parsing. In *NAACL-HLT*.

David McClosky, Mihai Surdeanu, and Christopher Manning. 2011. Event extraction as dependency parsing. In *BioNLP Shared Task Workshop*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *ICLR*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *NIPS*.

Makoto Miwa, Paul Thompson, and Sophia Ananiadou. 2013. Boosting automatic event extraction from the literature using domain adaptation and coreference resolution. In *Bioinformatics*.

Thien Huu Nguyen and Ralph Grishman. 2014. Employing word representations and regularization for domain adaptation of relation extraction. In *ACL*.

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *ACL-IJCNLP*.

Thien Huu Nguyen, Barbara Plank, and Ralph Grishman. 2015a. Semantic representations for domain adaptation: A case study on the tree kernel-based method for relation extraction. In *ACL-IJCNLP*.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *NAACL-HLT*.

Siddharth Patwardhan and Ellen Riloff. 2009. A unified model of phrasal and sentential evidence for information extraction. In *EMNLP*.

Barbara Plank and Alessandro Moschitti. 2013. Embedding semantic similarity in tree kernels for domain adaptation of relation extraction. In *ACL*.

Hoifung Poon and Lucy Vanderwende. 2010. Joint inference for knowledge extraction from biomedical literature. In *NAACL-HLT*.

Sebastian Riedel and Andrew McCallum. 2011. Fast and robust joint models for biomedical event extraction. In *EMNLP*.

Sebastian Riedel and Andrew McCallum. 2011b. Robust biomedical event extraction with dual decomposition and minimal domain adaptation. In *BioNLP Shared Task 2011 Workshop*.

Sebastian Riedel, Hong-Woo Chun, Toshihisa Takagi, and Jun'ichi Tsujii. 2009. A markov logic approach to bio-molecular event extraction. In *BioNLP 2009 Workshop*.

Joseph Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *ACL*.

Deepak Venugopal, Chen Chen, Vibhav Gogate, and Vincent Ng. 2014. Relieving the computational bottleneck: Joint inference for event extraction with high-dimensional features. In *EMNLP*.

Deyu Zhou, Dayou Zhong, and Yulan He. 2014. Event trigger identification for biomedical events extraction using domain knowledge. In *Bioinformatics Journal*.