Event Detection and Domain Adaptation with Convolutional Neural Networks

Thien Huu Nguyen
Computer Science Department
New York University
New York, NY 10003 USA
thien@cs.nyu.edu

Ralph Grishman
Computer Science Department
New York University
New York, NY 10003 USA
grishman@cs.nyu.edu

Abstract

We study the event detection problem using convolutional neural networks (CNNs) that overcome the two fundamental limitations of the traditional feature-based approaches to this task: complicated feature engineering for rich feature sets and error propagation from the preceding stages which generate these features. The experimental results show that the CNNs outperform the best reported feature-based systems in the general setting as well as the domain adaptation setting without resorting to extensive external resources.

1 Introduction

We address the problem of event detection (ED): identifying instances of specified types of events in text. Associated with each event mention is a phrase, the event trigger (most often a single verb or nominalization), which evokes that event. Our task, more precisely stated, involves identifying event triggers and classifying them into specific types. For instance, according to the ACE 2005 annotation guideline[1], in the sentence "*A police **officer** was **killed** in **New Jersey today***", an event detection system should be able to recognize the word "*killed*" as a trigger for the event "*Die*". This task is quite challenging, as the same event might appear in the form of various trigger expressions and an expression might represent different events in different contexts. ED is a crucial component in the overall task of event extraction, which also involves event argument discovery.

Recent systems for event extraction have employed either a pipeline architecture with separate classifiers for trigger and argument labeling (Ji and Grishman, 2008; Gupta and Ji, 2009; Patwardhan and Rilof, 2009; Liao and Grishman, 2011; McClosky et al., 2011; Huang and Riloff, 2012; Li et al., 2013a) or a joint inference architecture that performs the two subtasks at the same time to benefit from their inter-dependencies (Riedel and McCallum, 2011a; Riedel and McCallum, 2011b; Li et al., 2013b; Venugopal et al., 2014). Both approaches have coped with the ED task by elaborately hand-designing a large set of features (*feature engineering*) and utilizing the existing supervised natural language processing (NLP) toolkits and resources (i.e name tagger, parsers, gazetteers etc) to extract these features to be fed into statistical classifiers. Although this approach has achieved the top performance (Hong et al., 2011; Li et al., 2013b), it suffers from at least two issues:

(i) The choice of features is a manual process and requires linguistic intuition as well as domain expertise, implying additional studies for new application domains and limiting the capacity to quickly adapt to these new domains.

(ii) The supervised NLP toolkits and resources for feature extraction might involve errors (either due to the imperfect nature or the performance loss of the toolkits on new domains (Blitzer et al., 2006; Daumé III, 2007; McClosky et al., 2010)), probably propagated to the final event detector.

This paper presents a convolutional neural network (LeCun et al., 1988; Kalchbrenner et al., 2014) for the ED task that automatically learns features from sentences, and minimizes the dependence on supervised toolkits and resources for features, thus alleviating the error propagation and improving the performance for this task. Due to the emerging interest of the NLP community in deep learning recently, CNNs have been studied extensively and applied effectively in various tasks: semantic parsing (Yih et al., 2014), search query retrieval (Shen et al., 2014), semantic matching (Hu et al., 2014), sentence modeling and classification (Kalchbrenner et al., 2014; Kim,

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

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers), pages 365–371,
Beijing, China, July 26-31, 2015.

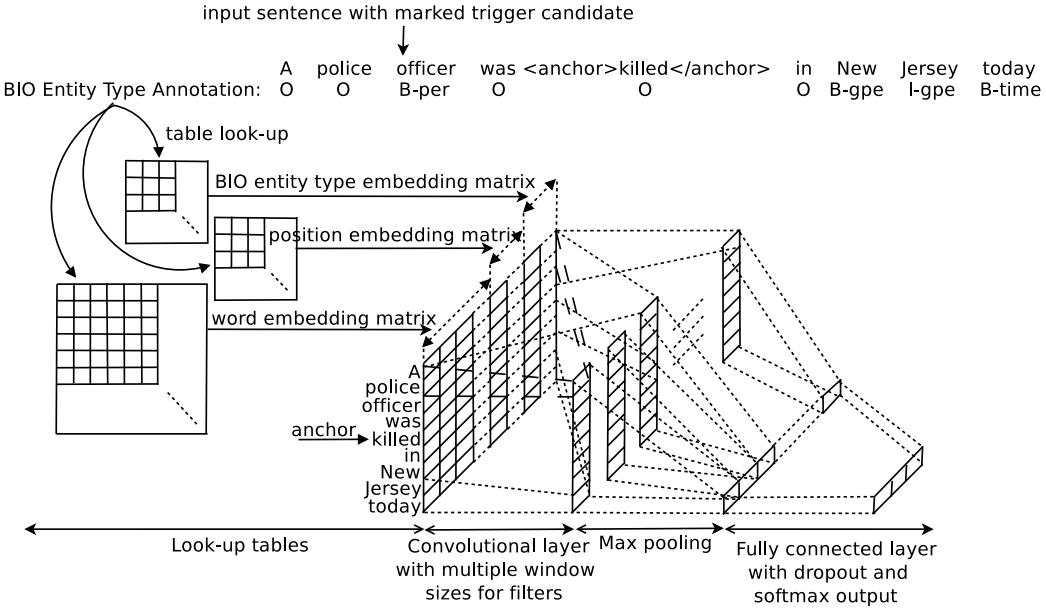

Figure 1: Convolutional Neural Network for Event Detection.

2014), name tagging and semantic role labeling (Collobert et al., 2011), relation classification and extraction (Zeng et al., 2014; Nguyen and Grishman, 2015). However, to the best of our knowledge, this is the first work on event detection via CNNs so far.

First, we evaluate CNNs for ED in the general setting and show that CNNs, though not requiring complicated feature engineering, can still outperform the state-of-the-art feature-based methods extensively relying on the other supervised modules and manual resources for features. Second, we investigate CNNs in a domain adaptation (DA) setting for ED. We demonstrate that CNNs significantly outperform the traditional feature-based methods with respect to generalization performance across domains due to: (i) their capacity to mitigate the error propagation from the preprocessing modules for features, and (ii) the use of word embeddings to induce a more general representation for trigger candidates. We believe that this is also the first research on domain adaptation using CNNs.

2 Model

We formalize the event detection problem as a multi-class classification problem. Given a sentence, for every token in that sentence, we want to predict if the current token is an event trigger: i.e, does it express some event in the pre-defined event set or not (Li et al., 2013b)? The current token along with its context in the sentence constitute an event trigger candidate or an example in multi-class classification terms. In order to prepare for the CNNs, we limit the context 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 the CNNs, each token x_i is transformed into a real-valued vector by looking up the following embedding tables to capture different characteristics of the token:

- Word Embedding Table (initialized by some pre-trained word embeddings): to capture the hidden semantic and syntactic properties of the tokens (Collobert and Weston, 2008; Turian et al., 2010).

- Position Embedding Table: to embed the relative distance i of the token x_i to the current token x_0. In practice, we initialize this table randomly.

- Entity Type Embedding Table: If we further know the entity mentions and their entity types[2] in the sentence, we can also capture this information for each token by looking up the entity type embedding table (initialized randomly) using the entity type associated with each token. We employ the BIO annotation scheme to assign entity type labels to each token in the trigger candidate

[2]For convenience, when mentioning entities in this paper, we always include ACE timex and values.

using the heads of the entity mentions.

For each token x_i, the vectors obtained from the three look-ups above are concatenated into a single vector $\mathbf{x}_i$ to represent the token. As a result, the original event trigger x is transformed into a matrix $\mathbf{x} = [\mathbf{x}_{-w}, \mathbf{x}_{-w+1}, \ldots, \mathbf{x}_0, \ldots, \mathbf{x}_{w-1}, \mathbf{x}_w]$ of size $m_t \times (2w+1)$ (m_t is the dimensionality of the concatenated vectors of the tokens).

The matrix representation $\mathbf{x}$ is then passed through a convolution layer, a max pooling layer and a softmax at the end to perform classification (like (Kim, 2014; Kalchbrenner et al., 2014)). In the convolution layer, we have a set of feature maps (filters) $\{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n\}$ for the convolution operation. Each feature map $\mathbf{f}_i$ corresponds to some window size k and can be essentially seen as a weight matrix of size $m_t \times k$. Figure 1 illustrates the proposed CNN.

The gradients are computed using back-propagation; regularization is implemented by a dropout (Kim, 2014; Hinton et al., 2012), and training is done via stochastic gradient descent with shuffled mini-batches and the AdaDelta update rule (Zeiler, 2012; Kim, 2014). During the training, we also optimize the weights of the three embedding tables at the same time to reach an effective state (Kim, 2014).

3 Experiments

3.1 Dataset, Hyperparameters and Resources

As the benefit of multiple window sizes in the convolution layer has been demonstrated in the previous work on sentence modeling (Kalchbrenner et al., 2014; Kim, 2014), in the experiments below, we use window sizes in the set {2, 3, 4, 5} to generate feature maps. We utilize 150 feature maps for each window size in this set. The window size for triggers is set to 31 while the dimensionality of the position embeddings and entity type embeddings is 50[3].We inherit the values for the other parameters from Kim (2014), i.e, the dropout rate $\rho = 0.5$, the mini-batch size = 50, the hyperparameter for the l_2 norms = 3. Finally, we employ the pre-trained word embeddings `word2vec` with 300 dimensions from Mikolov et al. (2013) for initialization.

We evaluate the presented CNN over the ACE 2005 corpus. For comparison purposes, we utilize the same test set with 40 newswire articles (672 sentences), the same development set with 30 other documents (836 sentences) and the same training set with the remaning 529 documents (14,849 sentences) as the previous studies on this dataset (Ji and Grishman, 2008; Liao and Grishman, 2010; Li et al., 2013b). The ACE 2005 corpus has 33 event subtypes that, along with one class "*None*" for the non-trigger tokens, constitutes a 34-class classification problem.

In order to evaluate the effectiveness of the position embeddings and the entity type embeddings, Table 1 reports the performance of the proposed CNN on the development set when these embeddings are either included or excluded from the systems. With the large margins of performance, it is very clear from the table that the position embeddings are crucial while the entity embeddings are also very useful for CNNs on ED.

Systems		P	R	F
-Entity Types	-Position	16.8	12.0	14.0
	+Position	75.0	63.0	**68.5**
+Entity Types	-Position	17.0	15.0	15.9
	+Position	75.6	66.4	**70.7**

Table 1: Performance on the Development Set.

For the experiments below, we examine the CNNs in two scenarios: excluding the entity type embeddings (CNN1) and including the entity type embeddings (CNN2). We always use position embeddings in these two scenarios.

3.2 Performance Comparison

The state-of-the-art systems for event detection on the ACE 2005 dataset have followed the traditional feature-based approach with rich hand-designed feature sets, and statistical classifiers such as MaxEnt and perceptron for structured prediction in a joint architecture (Hong et al., 2011; Li et al., 2013b). In this section, we compare the proposed CNNs with these state-of-the-art systems on the blind test set. Table 2 presents the overall performance of the systems with gold-standard entity mention and type information[4].

As we can see from the table, considering the systems that only use sentence level information, CNN1 significantly outperforms the MaxEnt classifier as well as the joint beam search with local features from Li et al. (2013b) (an improvement of 1.6% in F1 score), and performs comparably

[3]These values are chosen for their best performance on the development data.

[4]Entity mentions and types are used to introduce more features into the systems.

Methods	P	R	F
Sentence-level in Hong et al (2011)	67.6	53.5	59.7
MaxEnt with local features in Li et al. (2013b)	74.5	59.1	65.9
Joint beam search with local features in Li et al. (2013b)	73.7	59.3	65.7
Joint beam search with local and global features in Li et al. (2013b)	73.7	62.3	67.5
Cross-entity in Hong et al. (2011) †	72.9	64.3	68.3
CNN1: CNN without any external features	71.9	63.8	67.6
CNN2: CNN augmented with entity types	71.8	66.4	**69.0**

Table 2: Performance with Gold-Standard Entity Mentions and Types. † beyond sentence level.

with the joint beam search approach using both local and global features (Li et al., 2013b). This is remarkable since CNN1 does not require any external features[5], in contrast to the other feature-based systems that extensively rely on such external features to perform well. More interestingly, when the entity type information is incorporated into CNN1, we obtain CNN2 that still only needs sentence level information but achieves the state-of-the-art performance for this task (an improvement of 1.5% over the best system with only sentence level information (Li et al., 2013b)).

Except for CNN1, all the systems reported in Table 2 employ the gold-standard (perfect) entities mentions and types from manual annotation which might not be available in reality. Table 3 compares the performance of CNN1 and the feature-based systems in a more realistic setting, where entity mentions and types are acquired from an automatic high-performing name tagger and information extraction system (Li et al., 2013b). Note that CNN1 is eligible for this comparison as it does not utilize any external features, thus avoiding usage of the name tagger and the information extraction system to identify entity mentions and types.

3.3 Domain Adaptation Experiment

In this section, we aim to further compare the proposed CNNs with the feature-based systems under the domain adaptation setting for event detection.

The ultimate goal of domain adaptation research is to develop techniques taking training data in some *source domain* and learning models that can work well on *target domains*. The target domains are supposed to be so dissimilar from the source domain that the learning techniques would suffer from a significant performance loss when trained on the source domain and applied to the target domains. To make it clear, we address the unsupervised DA problem in this section, i.e no training data in the target domains (Blitzer et al., 2006; Plank and Moschitti, 2013). The fundamental reason for the performance loss of the feature-based systems on the target domains is twofold:

(i) The behavioral changes of features across domains: As domains differ, some features might be informative in the source domain but become less relevant in the target domains and vice versa.

(ii) The propagated errors of the pre-processing toolkits for lower-level tasks (POS tagging, name tagging, parsing etc) to extract features: These pre-processing toolkits are also known to degrade when shifted to target domains (Blitzer et al., 2006; Daumé III, 2007; McClosky et al., 2010), introducing noisy features into the systems for higher-level tasks in the target domains and eventually impairing the performance of these higher-level systems on the target domains.

For ED, we postulate that CNNs are more useful than the feature-based approach for DA for two reasons. First, rather than relying on the symbolic and concrete forms (i.e words, types etc) to construct features as the traditional feature-based systems (Ji and Grishman, 2008; Li et al., 2013b) do, CNNs automatically induce their features from word embeddings, the general distributed representation of words that is shared across domains. This helps CNNs mitigate the lexical sparsity, learn more general and effective feature representation for trigger candidates, and thus bridge the gap between domains. Second, as CNNs minimize the reliance on the supervised pre-processing toolkits for features, they can alleviate the error

Methods	F
Sentence level in Ji and Grishman (2008)	59.7
MaxEnt with local features in Li et al. (2013b)	64.7
Joint beam search with local features in Li et al. (2013b)	63.7
Joint beam search with local and global features in Li et al. (2013b)	65.6
CNN1: CNN without any external features	**67.6**

Table 3: Performance with Predicted Entity Mentions and Types.

[5]External features are the features generated from the supervised NLP modules and manual resources such as parsers, name tagger, entity mention extractors (either automatic or manual), gazetteers etc.

System	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 beam search in Li et al. (2013b)												
Joint+Local	73.5	62.7	67.7	70.3	57.2	63.1	64.9	50.8	57.0	59.5	38.4	46.7
Joint+Local+Global	72.9	63.2	67.7	68.8	57.5	62.6	64.5	52.3	57.7	56.4	38.5	45.7
CNN1	70.9	64.0	67.3	71.0	61.9	66.1†	64.0	55.0	59.1	53.2	38.4	44.6
CNN2	69.2	67.0	**68.0**	70.2	65.2	**67.6**†	68.3	58.2	**62.8**†	54.8	42.0	**47.5**

Table 4: In-domain (first column) and Out-of-domain Performance (columns two to four). Cells marked with †designate CNN models that significantly outperform ($p < 0.05$) all the reported feature-based methods on the specified domain.

propagation and be more robust to domain shifts.

3.3.1 Dataset

We also do the experiments in this part over the ACE 2005 dataset but focus more on the difference between domains. The ACE 2005 corpus comes with 6 different domains: broadcast conversation (bc), broadcast news (bn), telephone conversation (cts), newswire (nw), usenet (un) and webblogs (wl). Following the common practice of domain adaptation research on this dataset (Plank and Moschitti, 2013; Nguyen and Grishman, 2014), we use **news** (the union of **bn** and **nw**) as the source domain and **bc**, **cts**, **wl** as three different target domains. We take half of bc as the development set and use the remaining data for testing. We note that the distribution of event subtypes and the vocabularies of the source and target domains are quite different (Plank and Moschitti, 2013).

3.3.2 Domain Adaptation Results

Table 4 presents the performance of five systems: the MaxEnt classifier with the local features from Li et al. (2013b) (called *MaxEnt*); the state-of-the-art joint beam search systems with: (i) only local features (called *Joint+Local*); and (ii) both local and global features (called *Joint+Local+Global*) in Li et al. (2013b) (the baseline systems); CNN1 and CNN2 via 5-fold cross validation. For each system, we train a model on the training set of the source domain and report the performance of this model on the test set of the source domain (in-domain performance) as well as the performance of the model on the three target domains bc, cts and wl (out-of-domain performance)[6].

The main conclusions from the table include: (i) The baseline systems *MaxEnt*, *Joint+Local*, *Joint+Local+Global* achieve high performance on the source domain, but degrade dramatically on the target domains due to the domain shifts. (ii) Comparing CNN1 and the baseline systems, we see that CNN1 performs comparably with the baseline systems on the source domain (in-domain performance) (as expected), substantially outperform the baseline systems on two of the three target domains (i.e, bc and cts), and is only less effective than the joint beam search approach on the wl domain; (iii) Finally and most importantly, we consistently achieve the best adaptation performance across all the target domains with CNN2 by only introducing entity type information into CNN1. In fact, CNN2 significantly outperforms the feature-based systems with $p < 0.05$ and large margins of about 5.0% on the domains bc and cts, clearly confirming our argument in Section 3.3 and testifying to the benefits of CNNs on DA for ED.

[6]The performance of the feature-based systems *MaxEnt*, *Joint+Local* and *Joint+Local+Global* are obtained from the actual systems in Li et al. (2013b).

4 Conclusion

We present a CNN for event detection that automatically learns effective feature representations from pre-trained word embeddings, position embeddings as well as entity type embeddings and reduces the error propagation. We conducted experiments to compare the proposed CNN with the state-of-the-art feature-based systems in both the general setting and the domain adaptation setting. The experimental results demonstrate the effectiveness as well as the robustness across domains of the CNN. In the future, our plans include: (i) to explore the joint approaches for event extraction with CNNs; (ii) and to investigate other neural network architectures for information extraction.

Acknowledgments

We would like to thank Qi Li for providing the performance of the feature-based systems on the domain adaptation experiments. Thank you to Yifan He, Kai Cao, and Xiang Li for useful discussions on the task as well as the anonymous reviewers for their valuable feedback.

References

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. *Domain Adaptation with Structural Correspondence Learning*. In Proceedings of EMNLP.

Ronan Collobert and Jason Weston. 2008. *A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning*. In Proceedings of ICML.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu and Pavel Kuksa. 2011. *Natural Language Processing (Almost) from Scratch*. Journal of Machine Learning Research 12:24932537.

Hal Daumé III. 2007. *Frustratingly Easy Domain Adaptation*. In Proceedings of ACL.

Prashant Gupta and Heng Ji. 2009. *Predicting Unknown Time Arguments Based on Cross-Event Propagation*. In Proceedings of ACL-IJCNLP.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2012. *Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors*. CoRR, abs/1207.0580.

Yu Hong, Jianfeng Zhang, Bin Ma, Jian-Min Yao, Guodong Zhou, and Qiaoming Zhu. 2011. *Using Cross-entity Inference to Improve Event Extraction*. In Proceedings of ACL.

Baotian Hu, Zhengdong Lu, Hang Li, Qingcai Chen. 2014. *Convolutional Neural Network Architectures for Matching Natural Language Sentences*. In Proceedings of NIPS.

Ruihong Huang and Ellen Riloff. 2012. *Modeling Textual Cohesion for Event Extraction*. In Proceedings of AAAI.

Heng Ji and Ralph Grishman. 2008. *Refining Event Extraction through Cross-Document Inference*. In Proceedings of ACL.

Nal Kalchbrenner, Edward Grefenstette and Phil Blunsom. 2014. *A Convolutional Neural Network for Modeling Sentences*. In Proceedings of ACL.

Yoon Kim. 2014. *Convolutional Neural Networks for Sentence Classification*. In Proceedings of EMNLP.

Yann LeCun, Léon Bottou, Yoshua Bengio and Patrick Haffner. 1988. *Gradient-based Learning Applied to Document Recognition*. In Proceedings of the IEEE, 86(11):22782324.

Peifeng Li, Qiaoming Zhu, and Guodong Zhou. 2013a. *Argument Inference from Relevant Event Mentions in Chinese Argument Extraction*. In Proceedings of ACL.

Qi Li, Heng Ji, and Liang Huang. 2013b. *Joint Event Extraction via Structured Prediction with Global Features*. In Proceedings of ACL.

Shasha Liao and Ralph Grishman. 2010. *Using Document Level Cross-event Inference to Improve Event Extraction*. In Proceedings of ACL.

Shasha Liao and Ralph Grishman. 2011. *Acquiring Topic Features to Improve Event Extraction: in Preselected and Balanced Collections*. In Proceedings RANLP.

David McClosky, Eugene Charniak, and Mark Johnson. 2010. *Automatic Domain Adaptation for Parsing*. In Proceedings of HLT-NAACL.

David McClosky, Mihai Surdeanu, and Chris Manning. 2011. *Event Extraction as Dependency Parsing*. In Proceedings of ACL-HLT.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. *Distributed Representations of Words and Phrases and their Compositionality*. In Proceedings of NIPS.

Thien Huu Nguyen and Ralph Grishman. 2014. *Employing Word Representations and Regularization for Domain Adaptation of Relation Extraction*. In Proceedings of ACL.

Thien Huu Nguyen and Ralph Grishman. 2015. *Relation Extraction: Perspective from Convolutional Neural Networks*. In Proceedings of the NAACL Workshop on Vector Space Modeling for NLP (VSM).

Siddharth Patwardhan and Ellen Rilof. 2009. *A Unified Model of Phrasal and Sentential Evidence for Information Extraction*. In Proceedings of EMNLP.

Barbara Plank and Alessandro Moschitti. 2013. *Embedding Semantic Similarity in Tree Kernels for Domain Adaptation of Relation Extraction*. In Proceedings of ACL.

Sebastian Riedel and Andrew McCallum. 2011. *Fast and Robust Joint Models for Biomedical Event Extraction*. In Proceedings of EMNLP.

Sebastian Riedel and Andrew McCallum. 2011. *Robust Biomedical Event Extraction with Dual Decomposition and Minimal Domain Adaptation*. In Proceedings of the BioNLP Shared Task 2011 Workshop.

Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng and Grégoire Mesnil. 2014. *Learning Semantic Representations Using Convolutional Neural Networks for Web Search*. In Proceedings of WWW.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. *Word Representations: A Simple and General Method for Semi-supervised Learning*. In Proceedings of 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 Proceedings of EMNLP.

Wen-tau Yih, Xiaodong He, and Christopher Meek. 2014. *Semantic Parsing for Single-Relation Question Answering*. In Proceedings of ACL.

Matthew D. Zeiler. 2012. *ADADELTA: An Adaptive Learning Rate Method*. CoRR, abs/1212.5701.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou and Jun Zhao. 2014. *Relation Classification via Convolutional Deep Neural Network*. In Proceedings of COLING.