Improving Event Detection via Information Sharing Among Related Event Types

Shulin Liu[1,2](✉), Yubo Chen[1], Kang Liu[1], Jun Zhao[1,2], Zhunchen Luo[3], and Wei Luo[3]

[1] National Laboratory of Pattern Recognition Institute of Automation,
Chinese Academy of Sciences, Beijing 100190, China
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] China Defense Science and Technology Information Center,
Beijing 100142, China
{shulin.liu,yubo.chen,kliu,jzhao}@nlpr.ia.ac.cn
zhunchenluo@gmail.com, htqxjj@126.com

Abstract. Event detection suffers from data sparseness and label imbalance problem due to the expensive cost of manual annotations of events. To address this problem, we propose a novel approach that allows for information sharing among related event types. Specifically, we employ a fully connected three-layer artificial neural network as our basic model and propose a type-group regularization term to achieve the goal of information sharing. We conduct experiments with different configurations of type groups, and the experimental results show that information sharing among related event types remarkably improves the detecting performance. Compared with state-of-the-art methods, our proposed approach achieves a better F_1 score on the widely used ACE 2005 event evaluation dataset.

1 Introduction

In the ACE (Automatic Context Extraction) event extraction program, an event is represented as a structure consisting of an event trigger and a set of arguments. This paper tackles with the task of event detection (ED), which is a crucial component in the overall task of event extraction. The goal of ED is to identify event triggers and their corresponding event types from the given documents.

The dominative approaches to ED follow the supervised learning paradigm which exploits a set of labeled instances to train diverse models. However, the available annotated data is insufficient and highly imbalanced due to the expensive cost of manual annotations of events. ACE event evaluation program defines 33 event types (under eight coarse types), whereas the widely used dataset ACE 2005 corpus only contains 599 annotated documents, which is insufficient to train satisfying models. Even worse, ACE 2005 is highly imbalanced due to the significant occurrence difference between common and uncommon events. Table 1 shows the statistical information about the most frequent and infrequent labeled events in ACE 2005 corpus. From the table, we observed that the frequency of

M. Sun et al. (Eds.): CCL 2017 and NLP-NABD 2017, LNAI 10565, pp. 122–134, 2017.
https://doi.org/10.1007/978-3-319-69005-6_11

the most frequent events is 73 times (3009/41) more than that of the infrequent events. For common events, which typically occurs frequently in the real world, such as *Attack* and *Transport*, there are hundreds of labeled instances. By contrast, there are only few instances for uncommon events, where types *Extradite*, *Acquit* and *Release-Parole* contain even less than 10 labeled samples. Apparently, it is difficult to yield a satisfying performance using such a small scale of training data.

Table 1. The most frequent and infrequent event types and their frequencies of labeled samples in ACE 2005 corpus.

Frequent events		Infrequent events	
Type	Frequency	Type	Frequency
Attack	1367	*Merge-Org*	14
Transport	659	*Nominate*	12
Die	540	*Extradite*	7
Meet	262	*Acquit*	6
End-Position	181	*Release-Parole*	2
Total	3009	Total	41

In the ED task, events are associated with each other. For example, *Injure* and *Die* events are more likely to co-occur with *Attack* events than others, whereas *Marry* and *Born* events are unlikely to co-occur with *Attack* events. This information is very useful for the ED task. For example, in the sentence "*He* ***left*** *the company, and planned to* ***go*** *home directly.*", the trigger word **left** may trigger a *Transport* (a person left a place) event or an *End-Position* (a person retired from a company) event. However, if we take the following event triggered by **go** into consideration, we are confident to judge it as a *Transport* event rather than an *End-Position* event. Several existing approaches have been proposed to exploit the aforementioned information for the ED task. [21] proposed a two-pass cross-event method to employ event-event association information. [20] proposed a sentence-level joint model to capture the combinational features of triggers and arguments. [23] proposed a two staged approach based on the probabilistic soft logic model (PSL) [2,18] to utilize the association information among events. In these methods, the aforementioned information is encoded as features and learnt from the training data.

The main weakness of these methods is that they could not tackle with the data sparseness and label imbalance problem. The reasons are twofold. On the one hand, all these methods encode the event association information as features and learn them from the training data, however it is difficult to learn useful information for sparse events. On the other hand, from the perspective of the model (ignoring the features they used), all these methods treat events of various types independently and ignore the event-event association. On the contrast, in this paper we propose an approach to exploit the event-event association information from the perspective of the model, which allows related events to share information in the procedure of training. Specifically, we first divide all the event types into several groups. Then, we employ a three-layer Artificial Neural Networks (ANNs) [13] based event detection model to automatically learn features and propose a type-group regularization term to encourage events in the same group to share information in training process. In this way, events of sparse types are expected to benefit from that of dense types in the same group. Our idea is inspired by multi-task learning approaches [6,10], where multiple related prediction tasks are learned jointly, sharing information across the tasks.

Recently, [22] addressed this problem by leveraging FrameNet [3,11]. Contrast with their approach, our solution does not need any external resources. Moreover, our approach can be applied to theirs (see Sect. 2.2 for details). In summary, the contributions of this paper are as follows.

- To our knowledge, this is the first work to address the data sparseness and imbalance problem without using external resource for the ED task.
- We propose two event-type grouping strategies and apply them in our proposed detecting model. We also conduct a set of experiments to illustrate their performances.
- We conduct experiments using the widely used ACE 2005 dataset and its expanded version published by [22]. The experimental results on both datasets demonstrate that the proposed appraoch is effective for the ED task. Our approach outperforms state-of-the-art methods.

2 Background

2.1 Task Description

The ED task is a subtask of the ACE event evaluations. We first introduce the ACE event extraction task. In ACE evaluations, an event is defined as a specific occurrence involving one or more participants. Event extraction task requires that certain specified types of events, which are mentioned in the source language data, be detected. We introduce some ACE terminologies to facilitate the understanding of this task:

Entity: an object or a set of objects in one of the semantic categories of interests.

Entity mention: a reference to an entity (typically, a noun phrase).

Event trigger: the main word that most clearly expresses an event occurrence.

Event arguments: the mentions that are involved in an event (participants).

Event mention: a phrase or sentence within which an event is described, including the trigger and arguments.

The 2005 ACE evaluation included 8 supertypes of events, with 33 types. Consider the following sentence:

He ***died*** *in the hospital.*

An event extractor should detect a *Die* event mention, along with the trigger word "*died*", the victim "*He*" and the place "*hospital*".

Unlike the standard ACE event extraction task, event detection task concentrates only on trigger identification and event type classification, which implies that in the previous example, our task is to identify that the token "*died*" is an event trigger and that its type is *Die*.

2.2 Related Work

Event extraction is an increasingly hot and challenging research topic in NLP. Many approaches have been proposed to this task. Nearly all of the reported ACE event extraction approaches use supervised paradigm. We further divide supervised approaches to feature-based methods, structure-based methods and representation-based methods.

In feature-based methods, a diverse set of strategies have been exploited to convert classification clues (such as sequences and parse trees) into feature vectors. [1] uses the lexical features (e.g., full word, pos tag), syntactic features (e.g., dependency features) and external-knowledge features (WordNet) to extract events. Inspired by the hypothesis of "One Sense Per Discourse" [16,28] combined global evidence from related documents with local decisions for the event extraction. To capture more clues from the texts, [12,15,21] proposed the cross-event and cross-entity inference for the ACE event task. [23] proposed a global inference approach to employ both latent and global information for event detection. Although these approaches achieve high performance, feature-based methods suffer from the problem of selecting a suitable feature set when converting the classification clues into feature vectors.

In structure-based methods, researchers treat event extraction as the task of predicting the structure of the event in a sentence. [24] cast the problem of biomedical event extraction as a dependency parsing problem. [20] presented a joint framework for ACE event extraction based on structured perceptron with beam search. To use more information from the sentence, [19] proposed to extract entity mentions, relations and events in ACE task based on the unified structure.

In representation-based methods, candidate event mentions are represented by embedding, which typically are fed into neural networks. Several related approaches have been proposed to event detection [8,26,27]. [27] employed Convolutional Neural Networks (CNNs) to automatically extract sentence-level features for event detection. [8] proposed dynamic multi-pooling operation on CNNs to better capture sentence-level features. These methods yield relatively high performance. However, they all ignored the data sparseness and label imbalance problem.

Recently, [22] leveraged FrameNet to alleviate the data sparseness problem for event detection. They added the events automatically detected from FrameNet to the ACE corpus to achieve the goal of alleviating the data sparseness problem. They used FrameNet because of the highly similar structures and definitions of frames and events. The idea is simple but effective. Contrast with them, we try to solve the data sparseness problem by exploiting event-type consistency rather than using external resources. Moreover, our approach could be applied to theirs (via applying the proposed approach on the expanded ACE 2005 corpus generated by their method).

3 Methodology

[7] proved that performing trigger identification and classification in a unified manner is superior to handling them separately. Similar to previous work, we model these activities as a word classification task. Each word of a sentence is a trigger candidate, and our objective is to classify each of these candidates into one of the target classes (including a NEGATIVE class).

3.1 Basic Event Detection Model

We employ a fully connected three-layer (a input layer, a hidden layer and a soft-max output layer) Artificial Neural Networks (ANNs) [13] as the basic event detection model, which has been demonstrated very effective for the event detection task by [22].

Embedding Learning. Word embeddings learned from large amount of unlabeled data have been shown to be able to capture the meaningful semantic regularities of words [5,9]. This paper uses unsupervised pre-trained word embedding as the source of base features. In this work, we use the Skip-gram model [25] to pre-train the word embedding. This model is the state-of-the-art model in many NLP tasks [4,8]. The Skip-gram model trains the embeddings of words $w_1, w_2, ..., w_m$ by maximizing the average log probability,

$$\frac{1}{m}\sum_{t=1}^{m}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t) \tag{1}$$

where c is the size of the training window, m is the size of the unlabeled text. In this paper, we use the NYT corpus[1] to train word embeddings.

Model Training. Given a sentence, we concatenate the embedding vector of the candidate trigger and the average embedding vector of the words in the sentence as the input to the basic event detection model. Finally, for a given input sample $\mathbf{x}$, the ANNs with parameter θ outputs a vector $\mathbf{O}$, where the i-th value o_i in $\mathbf{O}$ is the confident score of classifying $\mathbf{x}$ to the i-th event type. To obtain the conditional probability $p(i|\mathbf{x}, \theta)$, we apply a softmax operation over all event types:

$$p(i|\mathbf{x}, \theta) = \frac{e^{o_i}}{\sum_{k=1}^{m} e^{o_k}} \tag{2}$$

[1] https://catalog.ldc.edu/LDC2008T19.

Given all of our (suppose T) training instances $(\mathbf{x^{(i)}}; y^{(i)})$, we can then define the negative log-likelihood loss function

$$J(\theta) = -\sum_{i=1}^{T} \log p(y^{(i)}|\mathbf{x^{(i)}}, \theta). \tag{3}$$

We train the model using a simple optimization technique called stochastic gradient descent (SGD) over shuffled mini-batches with the Adadelta rule [29]. Regularization is implemented by a dropout [14,17].

3.2 Type Group Regularization

As mentioned in Introduction, we want to encourage related types (which are indicated by the given type groups, and the grouping strategy will be introduced in the next subsection) to share information when training the model. To achieve this goal, a regularization term is proposed to the loss function,

$$R(\theta) = \sum_{\mathbf{g}\in\mathbf{G}} \frac{1}{|\mathbf{g}|} \sum_{k=1}^{|\mathbf{g}|} \frac{1}{\log(n^{(g,k)})} ||\mathbf{W_o^{(g,k)}} - \overline{\mu}_{\mathbf{g}}||^2 \tag{4}$$

$$\overline{\mu}_{\mathbf{g}} = \frac{1}{|g|} \sum_{j=1}^{|g|} \mathbf{W_o^{(g,j)}} \tag{5}$$

where $\mathbf{G}$ is type groups; $\mathbf{g}$ is one group in $\mathbf{G}$; $n^{(g,k)}$ is the instance amount of the k-th event type in $\mathbf{g}$; $\mathbf{W_o}$ is the weight matrix in the output layer (soft-max layer); $\overline{\mu}_{\mathbf{g}}$ is the average weight vector of all types in $\mathbf{g}$ (see Eq. 5) and $\mathbf{W_o^{(g,j)}}$ is the weight vector of the j-th event type in $\mathbf{g}$.

The hypothesis behind this intuition is that similar event types should have similar weight vectors in $\mathbf{W_o}$. The quadratic term in Eq. 4 encourages weight vectors of types in the same group to be similar. And the coefficient of it states that types with more labeled instances are less penalized by this term, which means that types with sufficient labeled instances should keep their own weight vectors. By contrast, for types which have less labeled instances, they should learn more from the group. In this way, sparse types are expected to benefit from tense types, which enables our model to alleviate the data sparseness and label imbalance problems for event detection.

Our final loss function $J^{'}(\theta)$ is defined as follows:

$$J^{'}(\theta) = J(\theta) + \alpha R(\theta) \tag{6}$$

where α is a hyper-parameter for trade-off between J and R. Akin to the basic event detection model, we minimize the loss function $J^{'}(\theta)$ using SGD over shuffled mini-batches with the Adadelta update rule.

3.3 Grouping Event Types

In this paper, we focus on alleviating the data sparseness and label imbalance problem, which means that our main goal is to improve the performance of sparse types. We can not learn type groups from the labeled corpus because it contains only a few instances of our target event types. A good choice is to manually group all the event types based on prior knowledge about events and their types. We propose two grouping strategies as follows.

G1: Positive vs. Negative. Our first grouping strategy is based on the hypothesis that all the positive events share some common characteristics to a certain extent compared with negative events (labeled with NEGATIVE). Thus, we divide all the event types into two groups. One of them contains all the positive event types, and the other only contains the NEGATIVE type.

G2: ACE Event Taxonomy. It is obvious that the first grouping strategy is too coarse, because not all the positive events share common characteristics to the same extent. For example, *Start-Org* events should share more common characteristics with *End-Org* events than with *Marry* events. Based on the above observation, we propose our second grouping strategy. We use the event taxonomy defined by ACE to group the event types.

All 33 positive event types in the ACE 2005 event evaluation program are grouped into eight supertypes (see Table 2). We obtained our event groups via slightly modifying these groups by moving the event types *Die* and *Injure* from supertype *Life* to *Conflict* because events of these two types often co-occur with events of type *Attack* and *Demonstrate*, which are in the supertype *Conflict*.

Table 2. Event taxonomy in ACE 2005 corpus.

Supertype	Type
Personal	*Start-Position, End-Position, Nominate, Elect*
Life	*Be-Born, Marry, Divorce, Injure, Die*
Movement	*Transport*
Contact	*Meet, Phone-Write*
Conflict	*Attack, Demonstrate*
Business	*Start-Org, End-Org, Merge-Org, Declare-Bankruptcy*
Transaction	*Transfer-Money, Transfer-Ownership*
Justice	*Arrest-Jail, Execute, Pardon, Release-Parole, Fine, Convict, Acquit, Appeal, Trial-Hearing, Charge-Indict, Sentence, Sue, Extradite*

4 Experiments

4.1 Data Set and Experimental Setup

ACE 2005 Corpus. We performed experiments on the ACE 2005 corpus. For the purpose of comparison, we followed the evaluation of [8,20,23]: randomly selected 30 articles from different genres as the development set, and subsequently conducted a blind test on a separate set of 40 ACE 2005 newswire documents. We used the remaining 529 articles as our training data set.

ExtACE 2005 Corpus. [22] used the events automatically detected from FrameNet as extra training data to alleviate the datasparseness problem for event detection. For simplicity sake, we denoted the ACE2005 corpus extended with FrameNet as ExtACE 2005 corpus. To investigate the effects of applying our approach to theirs, we also perform experiments on ExtACE2005 corpus. [22] published the events automatically detected from FrameNet, which can be easily obtained[2]. Note that, the development and test datasets hold the same as introduced in ACE 2005 corpus.

Evaluation Metrics. Following previous work [8,20,27], we use the following criteria to evaluate the results:

(1) A trigger is correctly identified if its offset matches a reference trigger.
(2) A trigger is correctly classified if both its event type and offset match a reference trigger. Finally, we use *Precision (P), Recall (R) and F meansure* (F_1) as the evaluation metrics.

Hyper-parameter Setting. Hyper-parameters are tuned by grid search on the development data set. We observed an interesting phenomenon when tuning parameters. For CNNs, updating word embeddings in the training procedure usually improves performances [8,27]. However, it is false for ANNs. Figure 1 shows the training curves on development data set. We observe that *UWE* (***U****pdating* ***W****ord* ***E****mbedding*) outperforms *NUWE* (***N****ot* ***U****pdating* ***W****ord* ***E****mbedding*) in the first five iterations. However, the situation is opposite in the remaining iterations. We believe the reason is that updating word embedding causes ANNs overly fit the training data and thus hurts the performances on development data. We apply regularization strategies to try to address this issue, but still fail to make *UWE* achieve good performances. In this work, word embeddings are not updated in training process.

In our experiments, we set the size of the hidden layer to 300, the size of word embeddings to 200, the batch size to 100 and the dropout rate to 0.5. The hyper-parameter α in Eq. 6 is various for different grouping strategies, we will give its setting in the next section.

[2] https://github.com/subacl/acl16.

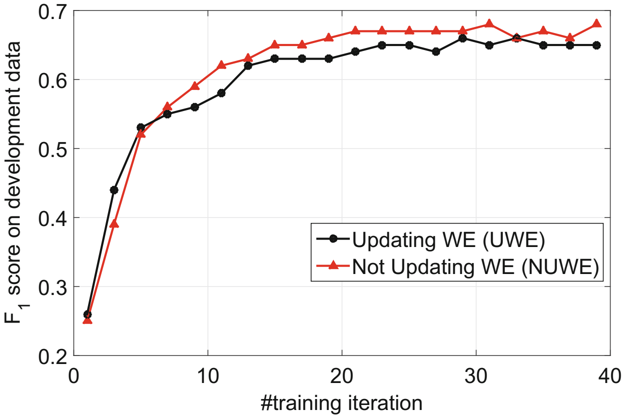

Fig. 1. Training curves on development data. UWE is short for "updating word embedding" whereas NUWE is short for "not updating word embedding".

4.2 Systems

In this section, we introduce the systems implemented in this work.

ANN is the basic event detection model, in which the hyper-parameter α is set to 0. In this system, event types do not share information.

ANN-G1 uses the first type grouping strategy G1 introduced in Subsect. 3.3. We use the development data set to tune the hyper-parameter α, and the final assignment is 2.56e-4.

ANN-G2 uses the second type grouping strategy G2 and the hyper-parameter α is set to 5.12e-5.

4.3 Experiments on ACE 2005 Corpus

We select the following state-of-the-art methods for comparison.

(1) *Li's joint model* is the method proposed by [20], which extracts events based on structure prediction. It is the best-reported structured-based system.
(2) *Ngyuen's CNN* is the method proposed by [27], which employs CNNs to detect events.
(3) *Chen's DMCNN* is the method proposed by [8], which employs dynamic multi-pooling operations on CNNs to extract events.
(4) *Liu's PSL* is the method proposed by [23], which employ both latent local and global information for event detection. It is the best-reported feature-based system.

Table 3 presents the experimental results on ACE 2005 corpus. The first group illustrates the performances of state-of-the-art approaches, and the second group illustrates the performances of our systems. Based on these results, we make the following observations:

Table 3. Experimental results on ACE 2005 corpus

Methods	Identification (%)			Classification (%)		
	P	R	F_1	P	R	F_1
Li's joint model (2013)	76.9	65.0	70.4	73.7	62.3	67.5
Nguyen's CNN (2015)	N/A			71.8	66.4	69.0
Chen's DMCNN (2015)	80.4	67.7	73.5	75.6	63.6	69.1
Liu's PSL (2016)	N/A			75.3	64.4	69.4
ANN (Ours)	83.1	63.5	72.0	79.7	60.9	69.0
ANN-G1 (Ours)	81.7	67.1	**73.7**	76.7	63.0	69.2
ANN-G2 (Ours)	82.0	64.7	72.3	78.9	62.2	**69.6**

(1) Information sharing among event types makes both *ANN-G1* and *ANN-G2* outperform the basic event detection model *ANN*, which demonstrates the effectiveness of our proposed approach.

(2) It is evident that the first grouping strategy G1 enables the event detection model to achieve more improvements for identification (whether it triggers an event or not) than for classification (what event type it triggers) (1.7% vs. 0.2%), and the second grouping strategy G2 is versa (0.3% vs. 0.6%). This phenomenon is easy to understand. Since G1 only differentiate positive events from negative events, it is reasonable to bring more improvements for identification than for classification. Whereas, G2 contains detail information for specific event types, thus it is more helpful for classification.

(3) Compared with state-of-the-art approaches, *ANN-G2* outperforms all of them with remarkable improvements. We also perform a t-test ($p \leqslant 0.05$), which indicates that our method significantly outperforms all of the compared methods.

4.4 Experiments on ExtACE 2005 Corpus

Recently, [22] used the events automatically detected from FrameNet as extra training data to alleviate the data sparseness problem for event detection. To investigate the effects of applying our method to theirs, we also perform experiments on ExtACE 2005 corpus, which is obtained by adding the events automatically detected from FrameNet to the ACE 2005 training data. Table 4 presents the experimental results. Consistent with the results reported in the above section, G1 makes *ANN-G1* achieve remarkable improvements for identification compared with *ANN* (74.0% vs. 72.9%). However, G2 fails to bring as much improvements as it performs on ACE 2005 corpus. The reason may be that the data sparseness problem in ExtACE 2005 corpus is less serious than that in ACE 2005. Nevertheless, the results demonstrate that information sharing among event types is also helpful for the ExtACE 2005 corpus.

Table 4. Experimental results on ExtACE 2005 corpus

Methods	Identification (%)			Classification (%)		
	P	R	F_1	P	R	F_1
ANN	79.2	67.5	72.9	76.8	65.5	70.7
ANN-G1	77.4	70.9	**74.0**	73.7	67.5	70.5
ANN-G2	78.5	69.1	73.5	75.6	66.6	**70.8**

4.5 Performances on Sparse Event Types

Our proposed approach allows for information sharing among related event types, which is expected to help the sparse types to benefit from dense types. To demonstrate the effectiveness of this intuition, we evaluate the proposed approach on the top 15 sparse event types[3].

Table 5. Performances of *ANN/ANN-G1/ANN-G2* on the top 15 sparse event types.

Dataset	Methods	Identification (%)			Classification (%)		
		P	R	F_1	P	R	F_1
ACE 2005	ANN	93.5	49.2	64.4	90.3	47.5	62.2
	ANN-G1	92.4	52.2	**66.7**	87.1	49.2	62.9
	ANN-G2	93.0	50.9	65.8	90.9	49.7	**64.3**
ExtACE 2005	ANN	91.8	50.6	65.2	88.8	48.9	63.1
	ANN-G1	91.4	53.3	***67.3***	86.6	50.5	63.8
	ANN-G2	92.0	52.6	66.9	89.0	50.8	***64.7***

Table 5 shows the experimental results, from which we could observe the following two results. (1) all systems achieve poor recall scores on sparse events. This is not difficult to understand: few training labeled data prevents the model to predict test samples to sparse types, which consequently causes poor recall scores. (2) Compared with *ANN*, *ANN-G1* and *ANN-G2* respectively improve the performances of identification and classification with remarkable gains on both datasets, which demonstrates that our approach is effective for sparse types.

5 Conclusions

We propose a novel approach for event detection that allows for information sharing among related event types. The proposed method uses given event type groups to decide which events should share information. In this paper, we explore

[3] *Appeal, Start-Org, Fine, Divorce, Execute, Merge-Org, Nominate, Extradite, Acquit, Declare-Bankruptcy, Pardon, End-Org, Be-Born, Sue* and *Release-Parole.*

two strategies, which are respectively denoted by G1 and G2, to group event types. To demonstrate the effectiveness of the proposed method, we conduct experiments on ACE 2005 corpus and its expanded version named ExtACE 2005. The results on both datasets demonstrate that the proposed approach is effective for the event detection task, and our approach outperforms all compared methods.

Acknowledgments. This work was supported by the Natural Science Foundation of China (No. 61533018) and the National Basic Research Program of China (No. 2014CB340503). And this research work was also supported by Google through focused research awards program.

References

1. Ahn, D.: The stages of event extraction. In: Proceedings of the Workshop on Annotating and Reasoning about Time and Events, pp. 1–8 (2006)
2. Bach, S.H., Huang, B., London, B., Getoor, L.: Hinge-loss Markov random fields: Convex inference for structured prediction. In: Proceedings of Uncertainty in Artificial Intelligence (UAI) (2013)
3. Baker, C.F., Fillmore, C.J., Lowe, J.B.: The berkeley framenet project. In: Proceedings of 17th Annual Meeting of the Association for Computational Linguistics, pp. 86–90 (1998)
4. Baroni, M., Dinu, G., Kruszewski, G.: Dont count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, vol. 1, pp. 238–247 (2014)
5. Bengio, Y., Ducharme, R., Vincent, P., Janvin, C.: A neural probabilistic language model. J. Mach. Learn. Res. **3**, 1137–1155 (2003)
6. Caruana, R.: Multitask learning. Mach. Learn. **28**(1), 41–75 (1997)
7. Chen, C., Ng, V.: Joint modeling for Chinese event extraction with rich linguistic features. In: COLING, pp. 529–544 (2012)
8. Chen, Y., Xu, L., Liu, K., Zeng, D., Zhao, J.: Event extraction via dynamic multi-pooling convolutional neural networks, pp. 167–176. Association for Computational Linguistics (2015)
9. Erhan, D., Bengio, Y., Courville, A., Manzagol, P.A., Vincent, P., Bengio, S.: Why does unsupervised pre-training help deep learning? J. Mach. Learn. Res. **11**, 625–660 (2010)
10. Evgeniou, T., Micchelli, C.A., Pontil, M.: Learning multiple tasks with kernel methods. J. Mach. Learn. Res. **6**(4), 615–637 (2005)
11. Fillmore, C.J., Johnson, C.R., Petruck, M.R.: Background to framenet. Int. J. Lexicogr. **16**(3), 235–250 (2003)
12. Gupta, P., Ji, H.: Predicting unknown time arguments based on cross-event propagation. In: Proceedings of ACL-IJCNLP, pp. 369–372 (2009)
13. Hagan, M.T., Demuth, H.B., Beale, M.H., et al.: Neural Network Design. PWS Publishing, Boston (1996)
14. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint. arXiv:1207.0580 (2012)

15. Hong, Y., Zhang, J., Ma, B., Yao, J., Zhou, G., Zhu, Q.: Using cross-entity inference to improve event extraction. In: Proceedings of ACL, pp. 1127–1136 (2011)
16. Ji, H., Grishman, R.: Refining event extraction through cross-document inference. In: Proceedings of ACL, pp. 254–262 (2008)
17. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp. 1746–1751 (2014)
18. Kimmig, A., Bach, S., Broecheler, M., Huang, B., Getoor, L.: A short introduction to probabilistic soft logic. In: Proceedings of NIPS Workshop, pp. 1–4 (2012)
19. Li, Q., Ji, H., Hong, Y., Li, S.: Constructing information networks using one single model. Association for Computational Linguistics (2014)
20. Li, Q., Ji, H., Huang, L.: Joint event extraction via structured prediction with global features. In: Proceedings of ACL, pp. 73–82 (2013)
21. Liao, S., Grishman, R.: Using document level cross-event inference to improve event extraction. In: Proceedings of ACL, pp. 789–797 (2010)
22. Liu, S., Chen, Y., He, S., Liu, K., Zhao, J.: Leveraging framenet to improve automatic event detection. In: Proceedings of ACL (2016)
23. Liu, S., Liu, K., He, S., Zhao, J.: A probabilistic soft logic based approach to exploiting latent and global information in event classification. In: Proceedings of the thirtieth AAAI Conference on Artificail Intelligence (2016)
24. McClosky, D., Surdeanu, M., Manning, C.D.: Event extraction as dependency parsing, pp. 1626–1635. Association for Computational Linguistics (2011)
25. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint. arXiv:1301.3781 (2013)
26. Nguyen, H.T., Grishman, R.: Modeling skip-grams for event detection with convolutional neural networks. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 886–891. Association for Computational Linguistics (2016)
27. Nguyen, T.H., Grishman, R.: Event detection and domain adaptation with convolutional neural networks. Association for Computational Linguistics (2015)
28. Yarowsky, D.: Unsupervised word sense disambiguation rivaling supervised methods. In: Proceedings of ACL, pp. 189–196 (1995)
29. Zeiler, M.D.: ADADELTA: An adaptive learning rate method. arXiv preprint. arXiv:1212.5701 (2012)