A Language-Independent Neural Network for Event Detection

Xiaocheng Feng[1], **Lifu Huang**[2], **Duyu Tang**[1], **Bing Qin**[1], **Heng Ji**[2], **Ting Liu**[1]
[1] Harbin Institute of Technology, Harbin, China
{*xcfeng, dytang, qinb, tliu*}@ir.hit.edu.cn
[2] Rensselaer Polytechnic Institute, Troy, USA
{*huangl7, jih*}@rpi.edu

Abstract

Event detection remains a challenge due to the difficulty at encoding the word semantics in various contexts. Previous approaches heavily depend on language-specific knowledge and pre-existing natural language processing (NLP) tools. However, compared to English, not all languages have such resources and tools available. A more promising approach is to automatically learn effective features from data, without relying on language-specific resources. In this paper, we develop a hybrid neural network to capture both sequence and chunk information from specific contexts, and use them to train an event detector for multiple languages without any manually encoded features. Experiments show that our approach can achieve robust, efficient and accurate results for multiple languages (English, Chinese and Spanish).

1 Introduction

Event detection aims to extract event triggers (most often a single verb or noun) and classify them into specific types precisely. It is a crucial and quite challenging sub-task of event extraction, because the same event might appear in the form of various trigger expressions and an expression might represent different event types in different contexts. Figure 1 shows two examples. In S1, "***release***" is a verb concept and a trigger for "*Transfer-Money*" event, while in S2, "***release*** " is a noun concept and a trigger for "*Release-Parole*" event.

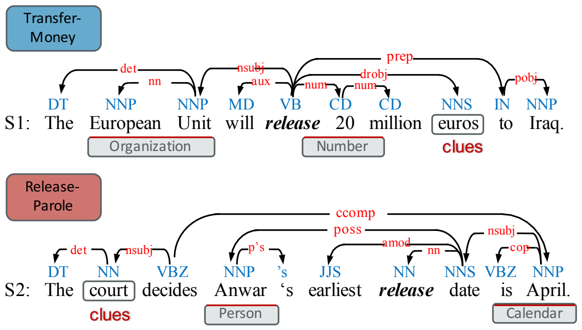
Transfer-Money
det
nn
nsubj
aux
prep
num
drobj
num
pobj
DT NNP NNP MD VB CD CD NNS IN NNP
S1: The European Unit will release 20 million euros to Iraq.
Organization
Number
clues
Release-Parole
ccomp
poss
nsubj
det
nsubj
p's
amod
nn
cop
DT NN VBZ NNP 's JJS NN NNS VBZ NNP
S2: The court decides Anwar 's earliest release date is April.
clues
Person
Calendar

Figure 1: Event type and syntactic parser results of an example sentence.

Most of previous methods (Ji et al., 2008; Liao et al., 2010; Hong et al., 2011; Li et al., 2013; Li et al., 2015b) considered event detection as a classification problem and designed a lot of lexical and syntactic features. Although such approaches perform reasonably well, features are often derived from language-specific resources and the output of pre-existing natural language processing toolkits (e,g., name tagger and dependency parser), which makes these methods difficult to be applied to different languages. Sequence and chunk are two types of meaningful language-independent structures for event detection. For example, in S2, when predicting the type of a trigger candidate " ***release***", the forward sequence information such as "court" can help the classifier label "***release***" as a trigger of a "*Release-Parole*" event. However, for feature engineering methods, it is hard to establish a relation between "court" and "release", because there is no direct dependency path between them. In addition, considering S1, "European Union" and "20 million euros" are two chunks, which indicate that this sentence is related to an organization and financial activities. These cluese are very helpful to infer "***release***" as a trigger of a "*Transfer-Money*" event. However, chunkers and parsers are only available for a few high-resource languages and their performance varies a lot.

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 66–71,
Berlin, Germany, August 7-12, 2016. ©2016 Association for Computational Linguistics

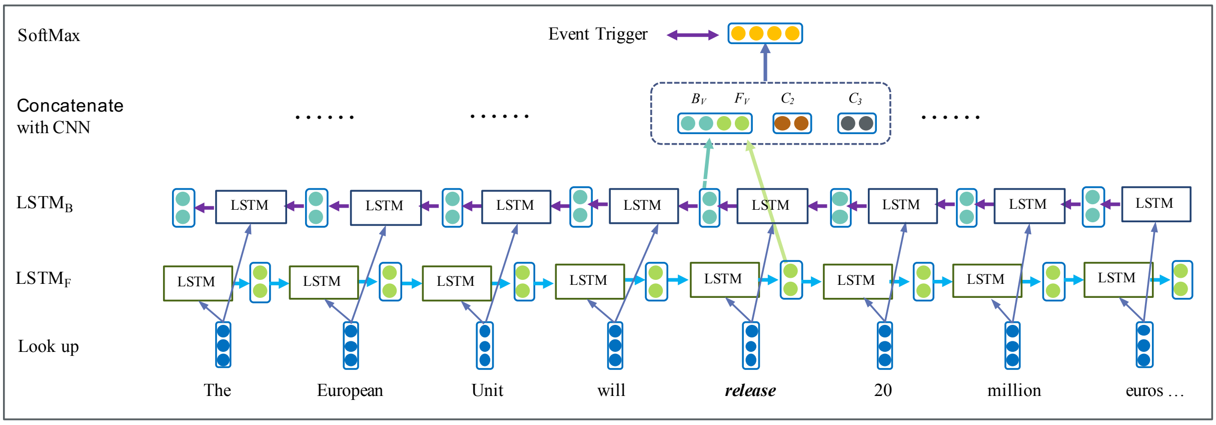

Figure 2: An illustration of our model for event trigger extraction (here the trigger candidate is "*release*"). F_v and B_v are the output of Bi-LSTM and C_2, C_3 are the output of CNN with convolutional filters with widths of 2 and 3.

Recently, deep learning techniques have been widely used in modeling complex structures and proven effective for many NLP tasks, such as machine translation (Bahdanau et al., 2014), relation extraction (Zeng et al., 2014) and sentiment analysis (Tang et al., 2015a). Bi-directional long short-term memory (Bi-LSTM) model (Schuster et al., 1997) is a two-way recurrent neural network (RNN) (Mikolov et al., 2010) which can capture both the preceding and following context information of each word. Convolutional neural network (CNN) (LeCun et al., 1995) is another effective model for extracting semantic representations and capturing salient features in a flat structure (Liu et al., 2015), such as chunks. In this work, we develop a hybrid neural network incorporating two types of neural networks: Bi-LSTM and CNN, to model both sequence and chunk information from specific contexts. Taking advantage of word semantic representation, our model can get rid of hand-crafted features and thus be easily adapted to multiple languages.

We evaluate our system on the event detection task for various languages for which ground-truth event detection annotations are available. In English event detection task, our approach achieved 73.4% F-score with average 3.0% absolute improvement compared to state-of-the-art. For Chinese and Spanish, the experiment results are also competitive. We demonstrate that our combined model outperforms traditional feature-based methods with respect to generalization performance across languages due to: (i) its capacity to model semantic representations of each word by capturing both sequence and chunk information. (ii) the use of word embeddings to induce a more general representation for trigger candidates.

2 Our Approach

In this section, we introduce a hybrid neural networks, which combines Bi-directional LSTM (Bi-LSTM) and convolutional neural networks to learn a continuous representation for each word in a sentence. This representation is used to predict whether the word is an event trigger or not. Specifically, we first use a Bi-LSTM to encode semantics of each word with its preceding and following information. Then, we add a convolutional neural network to capture structure information from local contexts.

2.1 Bi-LSTM

In this section we describe a Bidirectional LSTM model for event detection. Bi-LSTM is a type of bidirectional recurrent neural networks (RNN), which can simultaneously model word representation with its preceding and following information. Word representations can be naturally considered as features to detect triggers and their event types. As show in (Chen et al., 2015), we take all the words of the whole sentence as the input and each token is transformed by looking up word embeddings. Specifically, we use the Skip-Gram model to pre-train the word embeddings to represent each word (Mikolov et al., 2013; Bahdanau et al., 2014).

We present the details of Bi-LSTM for event trigger extraction in Figure 2. We can see that Bi-LSTM is composed of two LSTM neural networks, a forward LSTM$_F$ to model the preced-

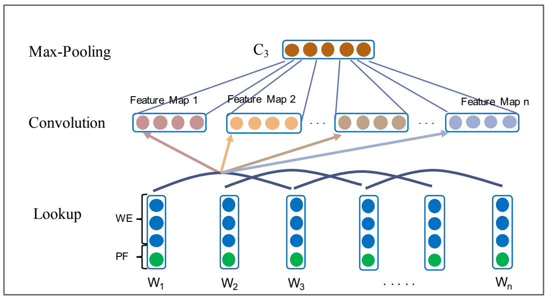

Figure 3: CNN structure.

ing contexts, and a backward LSTM$_B$ to model the following contexts respectively. The input of LSTM$_F$ is the preceding contexts along with the word as trigger candidate, and the input of LSTM$_B$ is the following contexts plus the word as trigger candidate. We run LSTM$_F$ from the beginning to the end of a sentence, and run LSTM$_B$ from the end to the beginning of a sentence. Afterwards, we concatenate the output F_v of LSTM$_F$ and B_v of LSTM$_B$ as the output of Bi-LSTM. One could also try averaging or summing the last hidden vectors of LSTM$_F$ and LSTM$_B$ as alternatives.

2.2 Convolution Neural Network

As the convolutional neural network (CNN) is good at capturing salient features from a sequence of objects (Liu et al., 2015), we design a CNN to capture some local chunks. This approach has been used for event detection in previous studies (Nguyen and Grishman, 2015; Chen et al., 2015). Specifically, we use multiple convolutional filters with different widths to produce local context representation. The reason is that they are capable of capturing local semantics of n-grams of various granularities, which are proven powerful for event detection. In our work, multiple convolutional filters with widths of 2 and 3 encode the semantics of bigrams and trigrams in a sentence. This local information can also help our model fix some errors due to lexical ambiguity.

An illustration of CNN with three convolutional filters is given in Figure 3. Let us denote a sentence consisting of n words as $\{w_1, w_2, ...w_i, ...w_n\}$, and each word w_i is mapped to its embedding representation $e_i \in \mathbb{R}^d$. In addition, we add a position feature (PF), which is defined as the relative distance between the current word and the trigger candidate. A convolutional filter is a list of linear layers with shared parameters. We feed the output of a convolutional filter to a MaxPooling layer and obtain an output vector with fixed length.

2.3 Output

At the end, we concatenate the bidirectional sequence features: F and B, which are learned from the Bi-LSTM, and local context features: C_2 and C_3, which are the output of CNN with convolutional filters with width of 2 and 3, as a single vector $O = [F, B, C_2, C_3]$. Then, we exploit a softmax approach to identify trigger candidates and classify each trigger candidate as a specific event type.

2.4 Training

In our model, the loss function is the cross-entropy error of event trigger identification and trigger classification. We initialize all parameters to form a uniform distribution $U(-0.01, 0.01)$. We set the widths of convolutional filters as 2 and 3. The number of feature maps is 300 and the dimension of the PF is 5. Table 1 illustrates the setting parameters used for three languages in our experiments (Zeiler, 2012).

3 Experiments

In this section, we will describe the detailed experimental settings and discuss the results. We evaluate the proposed approach on various languages (English, Chinese and Spanish) with Precision (P), Recall (R) and F-measure (F). Table 1 shows the detailed description of the data sets used in our experiments. We abbreviate our model as HNN (Hybrid Neural Networks).

3.1 Baseline Methods

We compare our approach with the following baseline methods.

(1) *MaxEnt*, a basesline feature-based method, which trains a Maximum Entropy classifier with some lexical and syntactic features (Ji et al., 2008).

(2) *Cross-Event* (Liao et al., 2010), using document-level information to improve the performance of ACE event extraction.

(3) *Cross-Entity* (Hong et al., 2011), extracting events using cross-entity inference.

(4) *Joint Model* (Li and Ji, 2014), a joint structured perception approach, incorporating multi-level linguistic features to extract event triggers and arguments at the same time so that local predictions can be mutually improved.

Language	Word Embedding		Gradient Learning Method		Data Sets			
	corpus	dim	method	parameters	Corpus	Train	Dev	Test
English	NYT	300	SGD	learning rate $r = 0.03$	ACE2005	529	30	40
Chinese	Gigaword	300	Adadelta	$p = 0.95, \delta = 1e^{-6}$	ACE2005	513	60	60
Spanish	Gigaword	300	Adadelta	$p = 0.95, \delta = 1e^{-6}$	ERE	93	12	12

Table 1: Hyperparameters and # of documents used in our experiments on three languages.

Model	Trigger Identification			Trigger Classification		
	P	R	F	P	R	F
MaxEnt	76.2	60.5	67.4	74.5	59.1	65.9
Cross-Event	N/A	N/A	N/A	68.7	68.9	68.8
Cross-Entity	N/A	N/A	N/A	72.9	64.3	68.3
Joint Model	76.9	65.0	70.4	73.7	62.3	67.5
PR	N/A	N/A	N/A	68.9	**72.0**	70.4
CNN	80.4	67.7	73.5	75.6	63.6	69.1
RNN	73.2	63.5	67.4	67.3	59.9	64.2
LSTM	78.6	67.4	72.6	74.5	60.7	66.9
Bi-LSTM	80.1	69.4	74.3	81.6	62.3	70.6
HNN	**80.8**	**71.5**	**75.9**	**84.6**	64.9	**73.4**

Table 2: Comparison of different methods on English event detection.

(5) *Pattern Recognition* (Miao and Grishman, 2015), using a pattern expansion technique to extract event triggers.

(6) *Convolutional Neural Network* (Chen et al., 2015), which exploits a dynamic multi-pooling convolutional neural network for event trigger detection.

3.2 Comparison On English

Table 2 shows the overall performance of all methods on the ACE2005 English corpus. We can see that our approach significantly outperforms all previous methods. The better performance of HNN can be further explained by the following reasons: (1) Compared with feature based methods, such as *MaxEnt*, *Cross-Event*, *Cross-Entity*, and *Joint Model*, neural network based methods (including *CNN*, *Bi-LSTM*, *HNN*) performs better because they can make better use of word semantic information and avoid the errors propagated from NLP tools which may hinder the performance for event detection. (2) Moreover, *Bi-LSTM* can capture both preceding and following sequence information, which is much richer than dependency path. For example, in S2, the semantic of "court" can be delivered to release by a forward sequence in our approach. It is an important clue which can help to predict "release" as a trigger for "*Release-Parole*". For explicit feature based methods, they can not establish a relation between "court" and "release", because they belong to different clauses, and there is no direct dependency path between them. While in our approach, the semantics of "court" can be delivered to release by a forward sequence. (3) *Cross-entity* system achieves higher recall because it uses not only sentence-level information but also document-level information. It utilizes event concordance to predict a local trigger's event type based on cross-sentence inference. For example, an "attack" event is more likely to occur with "killed" or "die" event rather than "marry" event. However, this method heavily relies on lexical and syntactic features, thus the precision is lower than neural network based methods. (4) *RNN* and *LSTM* perform slightly worse than Bi-LSTM. An obvious reason is that *RNN* and *LSTM* only consider the preceding sequence information of the trigger, which may miss some important following clues. Considering S1 again, when extracting the trigger "*releases*", both models will miss the following sequence "20 million euros to Iraq". This may seriously hinder the performance of RNN and LSTM for event detection.

3.3 Comparison on Chinese

For Chinese, we follow previous work (Chen et al., 2012) and employ Language Technology Platform (Liu et al., 2011) to do word segmentation.

Table 3 shows the comparison results between our model and the state-of-the-art methods (Li et al., 2013; Chen et al., 2012). *MaxEnt* (Li et al., 2013) is a pipeline model, which employs human-designed lexical and syntactic features. *Rich-C* is developed by Chen et al. (2012), which also incorporates Chinese-specific features to improve Chinese event detection. We can see that our method outperforms methods based on human designed features for event trigger identification and achieves comparable F-score for event classification.

3.4 Spanish Extraction

Table 4 presents the performance of our method on the Spanish ERE corpus. The results show that

Model	Trigger Identification			Trigger Classification		
	P	R	F	P	R	F
MaxEnt	50.0	**77.0**	60.6	47.5	73.1	57.6
Rich-C	62.2	71.9	66.7	58.9	**68.1**	**63.2**
HNN	**74.2**	63.1	**68.2**	**77.1**	53.1	63.0

Table 3: Results on Chinese event detection.

HNN approach performed better than LSTM and Bi-LSTM. It indicates that our proposed model could achieve the best performance in multiple languages than other neural network methods. We did not compare our system with other systems (Tanev et al., 2009), because they reported the results on a non-standard data set .

Model	Trigger Identification			Trigger Classification		
	P	R	F	P	R	F
LSTM	62.2	52.9	57.2	56.9	32.6	41.6
Bi-LSTM	76.2	63.1	68.7	61.5	42.2	50.1
HNN	**81.4**	**65.2**	**71.6**	**66.3**	**47.8**	**55.5**

Table 4: Results on Spanish event detection.

4 Related Work

Event detection is a fundamental problem in information extraction and natural language processing (Li et al., 2013; Chen et al., 2015), which aims at detecting the event trigger of a sentence (Ji et al., 2008). The majority of existing methods regard this problem as a classification task, and use machine learning methods with hand-crafted features, such as lexical features (e.g., full word, pos tag), syntactic features (e.g., dependency features) and external knowledge features (WordNet). There also exists some studies leveraging richer evidences like cross-document (Ji et al., 2008), cross-entity (Hong et al., 2011) and joint inference (Li and Ji, 2014).

Despite the effectiveness of feature-based methods, we argue that manually designing feature templates is typically labor intensive. Besides, feature engineering requires expert knowledge and rich external resources, which is not always available for some low-resource languages. Furthermore, a desirable approach should have the ability to automatically learn informative representations from data, so that it could be easily adapted to different languages. Recently, neural network emerges as a powerful way to learn text representation automatically from data and has obtained promising performances in a variety of NLP tasks.

For event detection, two recent studies (Nguyen and Grishman, 2015; Chen et al., 2015) explore neural network to learn continuous word representation and regard it as the feature to infer whether a word is a trigger or not. Nguyen (2015) presented a convolutional neural network with entity type information and word position information as extra features. However, their system limits the context to a fixed window size which leads the loss of word semantic representation for long sentences.

We introduce a hybrid neural network to learn continuous word representation. Compared with feature-based approaches, the method here does not require feature engineering and could be directly applied to different languages. Compared with previous neural models, we keep the advantage of convolutional neural network (Nguyen and Grishman, 2015) in capturing local contexts. Besides, we also incorporate a Bi-directional LSTM to model the preceding and following information of a word as it has been commonly accepted that LSTM is good at capturing long-term dependencies in a sequence (Tang et al., 2015b; Li et al., 2015a).

5 Conclusions

In this work, We introduce a hybrid neural network model, which incorporates both bidirectional LSTMs and convolutional neural networks to capture sequence and structure semantic information from specific contexts, for event detection. Compared with traditional event detection methods, our approach does not rely on any linguistic resources, thus can be easily applied to any languages. We conduct experiments on various languages (English, Chinese and Spanish. Empirical results show our approach achieved state-of-the-art performance in English and competitive results in Chinese. We also find that bi-directional LSTM is powerful for trigger extraction in capturing preceding and following contexts in long distance.

6 Acknowledgments

The authors give great thanks to Ying Lin (RPI) and Shen Liu for (HIT) the fruitful discussions. We also would like to thank three anonymous reviewers for their valuable comments and suggestions. RPI co-authors were supported by the U.S. DARPA LORELEI Program No. HR0011-15-C-0115, DARPA DEFT Program No. FA8750-13-2-0041 and NSF CAREER Award IIS-1523198.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Chen Chen, V Incent Ng, and et al. 2012. Joint modeling for chinese event extraction with rich linguistic features. In *In COLING*. Citeseer.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, volume 1, pages 167–176.

Yu Hong, Jianfeng Zhang, Bin Ma, Jianmin Yao, Guodong Zhou, and Qiaoming Zhu. 2011. Using cross-entity inference to improve event extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1127–1136. Association for Computational Linguistics.

Heng Ji, Ralph Grishman, and et al. 2008. Refining event extraction through cross-document inference. In *ACL*, pages 254–262.

Yann LeCun, Yoshua Bengio, and et al. 1995. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10).

Qi Li and Heng Ji. 2014. Incremental joint extraction of entity mentions and relations. In *Proceedings of the Association for Computational Linguistics*.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *ACL (1)*, pages 73–82.

Jiwei Li, Dan Jurafsky, and Eudard Hovy. 2015a. When are tree structures necessary for deep learning of representations? *arXiv preprint arXiv:1503.00185*.

Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. 2015b. A hierarchical neural autoencoder for paragraphs and documents. *arXiv preprint arXiv:1506.01057*.

Shasha Liao, Ralph Grishman, and et al. 2010. Using document level cross-event inference to improve event extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 789–797. Association for Computational Linguistics.

Ting Liu, Wanxiang Che, and Zhenghua Li. 2011. Language technology platform. *Journal of Chinese Information Processing*, 25(6):53–62.

Yang Liu, Furu Wei, Sujian Li, Heng Ji, Ming Zhou, and Houfeng Wang. 2015. A dependency-based neural network for relation classification. *arXiv preprint arXiv:1507.04646*.

Fan Miao and Ralph Grishman. 2015. Improving event detection with active learning. In *EMNLP*.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH*, volume 2, page 3.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. *Volume 2: Short Papers*, page 365.

Mike Schuster, Kuldip K Paliwal, and et al. 1997. Bidirectional recurrent neural networks. *Signal Processing, IEEE Transactions on*, 45(11):2673–2681.

Hristo Tanev, Vanni Zavarella, Jens Linge, Mijail Kabadjov, Jakub Piskorski, Martin Atkinson, and Ralf Steinberger. 2009. Exploiting machine learning techniques to build an event extraction system for portuguese and spanish. *Linguamática*, 1(2):55–66.

Duyu Tang, Bing Qin, and Ting Liu. 2015a. Document modeling with gated recurrent neural network for sentiment classification. EMNLP.

Duyu Tang, Bing Qin, and Ting Liu. 2015b. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432.

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao, et al. 2014. Relation classification via convolutional deep neural network. In *COLING*, pages 2335–2344.