Biomedical Event Trigger Identification Using Bidirectional Recurrent Neural Network Based Models

Patchigolla V S S Rahul, Sunil Kumar Sahu, Ashish Anand
Department of Computer Science and Engineering
Indian Institute of Technology Guwahati, Assam, India
rahul.rahul.pvss@gmail.com
{sunil.sahu, anand.ashish}@iitg.ernet.in

Abstract

Biomedical events describe complex interactions between various biomedical entities. Event trigger is a word or a phrase which typically signifies the occurrence of an event. Event trigger identification is an important first step in all event extraction methods. However many of the current approaches either rely on complex hand-crafted features or consider features only within a window. In this paper we propose a method that takes the advantage of recurrent neural network (RNN) to extract higher level features present across the sentence. Thus hidden state representation of RNN along with word and entity type embedding as features avoid relying on the complex hand-crafted features generated using various NLP toolkits. Our experiments have shown to achieve state-of-art F1-score on Multi Level Event Extraction (MLEE) corpus. We have also performed category-wise analysis of the result and discussed the importance of various features in trigger identification task.

1 Introduction

Biomedical events play an important role in improving biomedical research in many ways. Some of its applications include pathway curation (Ohta et al., 2013) and development of domain specific semantic search engine (Ananiadou et al., 2015). So as to gain attraction among researchers many challenges such as BioNLP'09 (Kim et al., 2009), BioNLP'11 (Kim et al., 2011), BioNLP'13 (Nédellec et al., 2013) have been organized and many novel methods have also been proposed addressing these tasks.

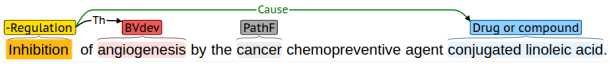

Figure 1: Example of a complex biomedical event

An event can be defined as a combination of a trigger word and arbitrary number of arguments. Figure 1 shows two events with trigger words as "*Inhibition*" and "*Angiogenesis*" of trigger types "*Negative Regulation*" and "*Blood Vessel Development*" respectively. Pipelined based approaches for biomedical event extraction include event trigger identification followed by event argument identification. Analysis in multiple studies (Wang et al., 2016b; Zhou et al., 2014) reveal that more than 60% of event extraction errors are caused due to incorrect trigger identification.

Existing event trigger identification models can be broadly categorized in two ways: *rule based approaches* and *machine learning based approaches*. Rule based approaches use various strategies including pattern matching and regular expression to define rules (Vlachos et al., 2009). However, defining these rules are very difficult, time consuming and requires domain knowledge. The overall performance of the task depends on the quality of rules defined. These approaches often fail to generalize for new datasets when compared with machine learning based approaches. Machine learning based approaches treat the trigger identification problem as a word level classification problem, where many features from the data are extracted using various NLP toolkits (Pyysalo et al., 2012; Zhou et al., 2014) or learned automatically (Wang et al., 2016a,b).

In this paper, we propose an approach using RNN to learn higher level features without the requirement of complex feature engineering. We thoroughly evaluate our proposed approach on

MLEE corpus. We also have performed category-wise analysis and investigate the importance of different features in trigger identification task.

2 Related Work

Many approaches have been proposed to address the problem of event trigger identification. Pyysalo et al. (2012) proposed a model where various hand-crafted features are extracted from the processed data and fed into a Support Vector Machine (SVM) to perform final classification. Zhou et al. (2014) proposed a novel framework for trigger identification where embedding features of the word combined with hand-crafted features are fed to SVM for final classification using multiple kernel learning. Wei et al. (2015) proposed a pipeline method on BioNLP'13 corpus based on Conditional Random Field (CRF) and Support vector machine (SVM) where the CRF is used to tag valid triggers and SVM is finally used to identify the trigger type. The above methods rely on various NLP toolkits to extract the hand-crafted features which leads to error propagation thus affecting the classifier's performance. These methods often need to tailor different features for different tasks, thus not making them generalizable. Most of the hand-crafted features are also traditionally sparse one-hot features vector which fail to capture the semantic information.

Wang et al. (2016b) proposed a neural network model where dependency based word embeddings (Levy and Goldberg, 2014) within a window around the word are fed into a feed forward neural network (FFNN) (Collobert et al., 2011) to perform final classification. Wang et al. (2016a) proposed another model based on convolutional neural network (CNN) where word and entity mention features of words within a window around the word are fed to a CNN to perform final classification. Although both of the methods have achieved good performance they fail to capture features outside the window.

3 Model Architecture

We present our model based on bidirectional RNN as shown in Figure 2 for the trigger identification task. The proposed model detects trigger word as well as their type. Our model uses embedding features of words in the input layer and learns higher level representations in the subsequent layers and

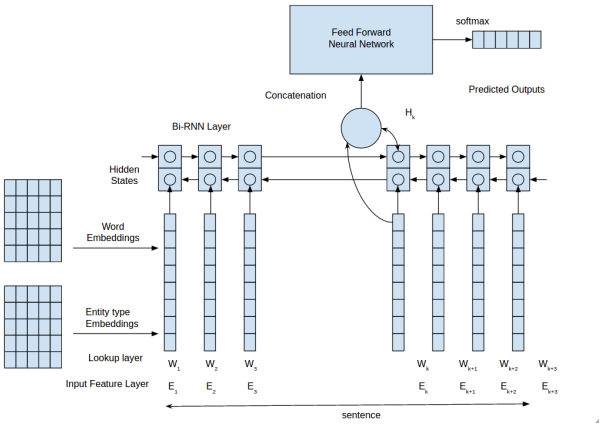

Figure 2: Model Architecture

makes use of both the input layer and higher level features to perform the final classification. We now briefly explain about each component of our model.

3.1 Input Feature Layer

For every word in the sentence we extract two features, exact word $w \in$ W and entity type $e \in E$. Here W refers the dictionary of words and E refers to dictionary of entities. Apart from all the entities, E also contains a $None$ entity type which indicates absence of an entity. In some cases the entity might span through multiple words, in that case we assign every word spanned by that entity the same entity type.

3.2 Embedding or Lookup Layer

In this layer every input feature is mapped to a dense feature vector. Let us say that E_w and E_e be the embedding matrices of W and E respectively. The features obtained from these embedding matrices are concatenated and treated as the final word-level feature (l) of the model.

The $E_w \in \mathbb{R}^{n_w \times d_w}$ embedding matrix is initialized with pre-trained word embeddings and $E_e \in \mathbb{R}^{n_e \times d_e}$ embedding matrix is initialized with random values. Here n_w, n_e refer to length of the word dictionary and entity type dictionary respectively and d_w, d_e refer to dimension of word and entity type embedding respectively.

3.3 Bidirectional RNN Layer

RNN is a powerful model for learning features from sequential data. We use both LSTM (Hochreiter and Schmidhuber, 1997) and GRU (Chung et al., 2014) variants of RNN in our ex-

periments as they handle the vanishing and exploding gradient problem (Pascanu et al., 2012) in a better way. We use bidirectional version of RNN (Graves, 2013) where for every word forward RNN captures features from the past and the backward RNN captures features from future, inherently each word has information about whole sentence.

3.4 Feed Forward Neural Network

The hidden state of the bidirectional RNN layer acts as sentence-level feature (g), the word and entity type embeddings (l) act as a word-level features, are both concatenated (1) and passed through a series of hidden layers (2), (3) with dropout (Srivastava et al., 2014) and an output layer. In the output layer, the number of neurons are equal to the number of trigger labels. Finally we use $Softmax$ function (4) to obtain probability score for each class.

$$f = g^k \oplus l^k \quad (1)$$

$$h_0 = \tanh(W_0 f + b_0) \quad (2)$$

$$h_i = \tanh(W_i h_{i-1} + b_i) \quad (3)$$

$$p(y|x) = Softmax(W_o h_i + b_o) \quad (4)$$

Here k refers to the k^{th} word of the sentence, i refers to the i^{th} hidden layer in the network and $\oplus$ refers to concatenation operation. W_i,W_o and b_i,b_o are parameters of the hidden and output layers of the network respectively.

3.5 Training and Hyperparameters

We use cross entropy loss function and the model is trained using stochastic gradient descent. The implementation[1] of the model is done in python language using $Theano$ (Bergstra et al., 2010) library. We use pre-trained word embeddings obtained by Moen et al. (2013) using *word2vec* tool (Mikolov et al., 2013).

We use training and development set for hyperparameter selection. We use word embeddings of 200 dimension, entity type embeddings of 50 dimension, RNN hidden state dimension of 250 and 2 hidden layers with dimension 150 and 100. In both the hidden layers we use dropout of 0.2.

[1]Implementation is available at https://github.com/rahulpatchigolla/EventTriggerDetection

4 Experiments and discussion

4.1 Dataset Description

We use MLEE (Pyysalo et al., 2012) corpus for performing our trigger identification experiments. Unlike other corpora on event extraction it covers events across various levels from molecular to organism level. The events in this corpus are broadly divided into 4 categories namely "*Anatomical*", "*Molecular*", "*General*", "*Planned*" which are further divided into 19 sub-categories as shown in Table 1. Here our task is to identify correct subcategory of an event. The entity types associated with the dataset are summarized in Table 2.

Category	Trigger label	Train count	Test count
Anatomical	Cell Proliferation (CELLP)	82	43
	Development (DEV)	202	98
	Blood Vessel Development (BVD)	540	305
	Death (DTH)	57	36
	Breakdown (BRK)	44	23
	Remodeling (REMDL)	22	10
	Growth (GRO)	107	56
Molecular	Synthesis (SYN)	13	4
	Gene Expression (GENEXP)	210	132
	Transcription (TRANS)	16	7
	Catabolism (CATA)	20	4
	Phosphorylation (PHO)	26	3
	Dephosphorylation (DEPHO)	2	1
General	Localization (LOC)	282	133
	Binding (BIND)	102	56
	Regulation (REG)	362	178
	Positive Regulation (PREG)	654	312
	Negative Regulation (NREG)	450	233
Planned	Planned Process (PLP)	407	175

Table 1: Statistics of event triggers in MLEE corpus

Category	Entity label	Train count	Test count
Molecule	Drug or Compound	637	307
	Gene or Gene Product	1961	1001
Anatomy	Organism Subdivision	27	22
	Anatomical System	10	8
	Organ	123	53
	Multi-tissue Structure	348	166
	Tissue	304	122
	Cell	866	332
	Cellular Component	105	40
	Developing Anatomical Structure	4	2
	Organism Substance	82	60
	Immaterial Anatomical Entity	11	4
	Pathological Formation	553	357
Organism	Organism	485	237

Table 2: Statistics of entities in MLEE corpus

4.2 Experimental Design

The data is provided in three parts as training, development and test sets. Hyperparameters are tuned using development set and then final model is trained on the combined set of training and development sets using the selected hyperparameters. The final results reported here are the best results over 5 runs.

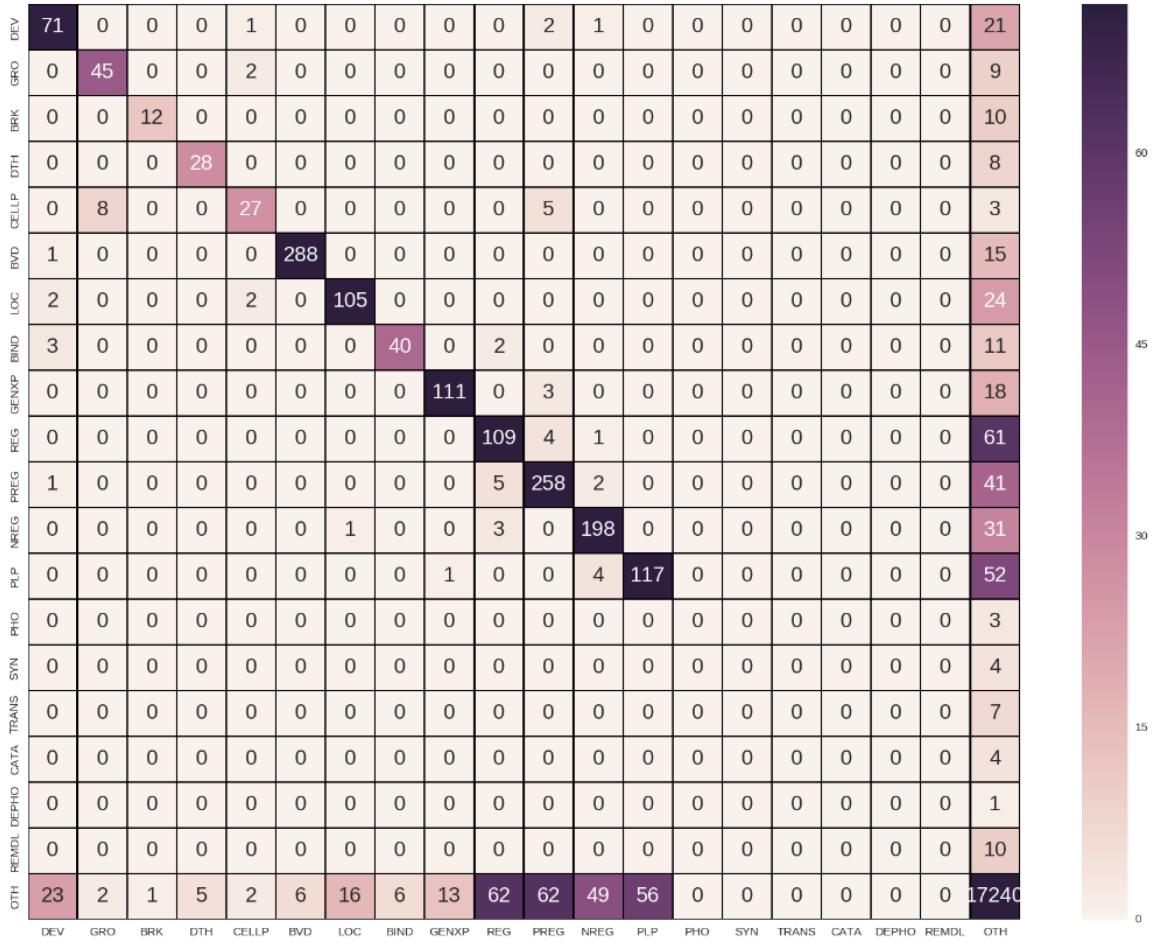

Figure 3: Confusion matrix of trigger classes with abbreviations mentioned in Table 1

We have used micro averaged F1-score as the evaluation metric and evaluated the performance of the model by ignoring the trigger classes with counts ≤ 10 in test set while training and considered them directly as false-negative while testing.

4.3 Performance comparison with Baseline Models

We compare our results with baseline models shown in Table 3. Pyysalo et al. (2012) defined a SVM based classifier with hand-crafted features. Zhou et al. (2014) also defined a SVM based classifier with word embeddings and hand-crafted features. Wang et al. (2016a) defined window based CNN classifier. Apart from the proposed models we also compare our results with two more baseline methods FFNN and CNN$^{\psi}$ which are our implementations. Here FFNN is a window based feed forward neural network where embedding features of words within the window are used to predict the trigger label (Collobert et al., 2011). We chose window size as 3 (one word from left and one word from right) after tuning it in validation set. CNN$^{\psi}$ is our implementation of window based CNN classifier proposed by Wang et al. (2016a) due to unavailability of their code in public domain. Our proposed model have shown slight improvement in F1-score when compared with baseline models. The proposed model's ability to capture the context of the whole sentence is likely to be one of the reasons of improvement in performance.

We perform one-side t-test over 5 runs of F1-Scores to verify our proposed model's performance when compared with FFNN and CNN$^{\Psi}$. The p value of the proposed model (GRU) when compared with FFNN and CNN$^{\psi}$ are 8.57×10^{-07} and 1.178×10^{-10} respectively. This indicates statistically superior performance of the proposed model.

4.4 Category Wise Performance Analysis

The category wise performance of the proposed model is shown in Table 4. It can be observed that

Method	Precision	Recall	F1-Score
SVM (Pyysalo et al., 2012)	81.44	69.48	75.67
SVM+W_e (Zhou et al., 2014)	80.60	74.23	77.82
CNN (Wang et al., 2016a)	80.67	76.76	78.67
FFNN	77.53	75.55	76.53
CNN$^{\psi}$	80.75	69.36	74.62
Proposed (LSTM)	78.58	78.84	**78.71**
Proposed (GRU)	79.78	78.45	**79.11**

Table 3: Comparison of performance of our model with baseline models

model's performance in *anatomical* and *molecular* categories are better than *general* and *planned* categories. We can also infer from the confusion matrix shown in Figure 3 that *positive regulation*, *negative regulation* and *regulation* among *general* category and *planned* category triggers are causing many false positives and false negatives thus degrading the model's performance.

Trigger Category	Precision	Recall	F1-Score
Anatomical	88.86	83.06	85.87
Molecular	88.80	73.51	80.43
General	75.69	78.53	77.09
Planned	67.63	67.24	67.43
Overall	79.78	78.45	79.11

Table 4: Category wise performance of the model

4.5 Further Analysis

In this section we investigate the importance of various features and model variants as shown in Table 5. Here E_w and E_e refer to using word and entity type embedding as a feature in the model, l and g refer to using word-level and sentence-level feature respectively for the final prediction. For example, $E_w + E_e$ and g means using both word and entity type embedding as the input feature for the model and g means only using the global feature (hidden state of RNN) for final prediction.

Index	Method	F1-Score
1	E_w and g	76.52
2	E_w and $l + g$	77.59
3	$E_w + E_e$ and g	78.70
4	$E_w + E_e$ and $l + g$	79.11

Table 5: Affect on F1-Score based on feature analysis and model variants

Examples in Table 6 illustrate importance of features used in best performing models. In phrase 1 the word "*knockdown*", is a part of an entity namely "*round about knockdown endothelial cells*" of type "*Cell*" and in phrase 2 it is trigger word of type "*Planned Process*", methods 1 and 2 failed to differentiate both of them because of no knowledge about the entity type. In phrase 3 "*impaired*" is a trigger word of type "*Negative Regulation*" methods 1 and 3 failed to correctly identify but when reinforced with word-level feature the model succeeded in identification. So, we can say that E_e feature and $l + g$ model variant help in improving the model's performance.

Index	Phrase
1	silencing of directional migration in *round about knockdown endothelial cells*
2	we show that PSMA inhibition *knockdown* or deficiency decrease
3	display altered maternal hormone concentrations indicative of an *impaired* trophoblast capacity

Table 6: Example phrases for Further Analysis

5 Conclusion and Future Work

In this paper we have proposed a novel approach for trigger identification by learning higher level features using RNN. Our experiments have shown to achieve state-of-art results on MLEE corpus. In future we would like to perform complete event extraction using deep learning techniques.

References

Sophia Ananiadou, Paul Thompson, Raheel Nawaz, John McNaught, and Douglas B Kell. 2015. Event-based text mining for biology and functional genomics. *Briefings in functional genomics* 14(3):213–230.

James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. 2010. Theano: A cpu and gpu math compiler in python. In *Proc. 9th Python in Science Conf*. pages 1–7.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR* abs/1412.3555.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* 12:2493–2537.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *CoRR* abs/1308.0850.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.* pages 1735–1780.

Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. 2009. Overview of bionlp'09 shared task on event extraction. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*. Association for Computational Linguistics, pages 1–9.

Jin-Dong Kim, Sampo Pyysalo, Tomoko Ohta, Robert Bossy, Ngan Nguyen, and Jun'ichi Tsujii. 2011. Overview of bionlp shared task 2011. In *Proceedings of the BioNLP Shared Task 2011 Workshop*. Association for Computational Linguistics, pages 1–6.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *ACL 2014*. pages 302–308.

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.

Hans Moen, Sampo Pyysalo, Filip Ginter, Tapio Salakoski, and Sophia Ananiadou. 2013. Distributional semantics resources for biomedical text processing.

Claire Nédellec, Robert Bossy, Jin-Dong Kim, Jung-Jae Kim, Tomoko Ohta, Sampo Pyysalo, and Pierre Zweigenbaum. 2013. Overview of bionlp shared task 2013. In *Proceedings of the BioNLP Shared Task 2013 Workshop*. Association for Computational Linguistics, pages 1–7.

Tomoko Ohta, Sampo Pyysalo, Rafal Rak, Andrew Rowley, Hong-Woo Chun, Sung-Jae Jung, Changhoo Jeong, Sung-pil Choi, and Sophia Ananiadou. 2013. Overview of the pathway curation (pc) task of bionlp shared task 2013 .

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2012. Understanding the exploding gradient problem. *CoRR, abs/1211.5063* .

Sampo Pyysalo, Tomoko Ohta, Makoto Miwa, Han-Cheol Cho, Jun'ichi Tsujii, and Sophia Ananiadou. 2012. Event extraction across multiple levels of biological organization. *Bioinformatics* 28(18):575–581.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15(1):1929–1958.

Andreas Vlachos, Paula Buttery, Diarmuid O Séaghdha, and Ted Briscoe. 2009. Biomedical event extraction without training data. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*. Association for Computational Linguistics, pages 37–40.

Jian Wang, Honglei Li, Yuan An, Hongfei Lin, and Zhihao Yang. 2016a. Biomedical event trigger detection based on convolutional neural network. *International Journal of Data Mining and Bioinformatics* 15(3):195–213.

Jian Wang, Jianhai Zhang, Yuan An, Hongfei Lin, Zhihao Yang, Yijia Zhang, and Yuanyuan Sun. 2016b. Biomedical event trigger detection by dependency-based word embedding. *BMC Medical Genomics* 9(2):45.

Xiaomei Wei, Qin Zhu, Chen Lyu, Kai Ren, and Bo Chen. 2015. A hybrid method to extract triggers in biomedical events. *Journal of Digital Information Management* 13(4):299.

Deyu Zhou, Dayou Zhong, and Yulan He. 2014. Event trigger identification for biomedical events extraction using domain knowledge. *Bioinformatics* 30(11):1587–1594.