NEC and Tokyo Institute of Technology in TAC KBP 2017: Multichannel Encoding and Stochastic Voting for Event Detection Model

Kai Ishikawa
Data Science Research Laboratories, NEC Corporation

Hiroya Takamura
Institute of Innovative Research, Tokyo Institute of Technology

Manabu Okumura
Institute of Innovative Research, Tokyo Institute of Technology

Abstract

This is the first time we participate in the KBP evaluation tasks of the Text Analysis Conference (TAC). This year, we developed an event detection system and submitted to Event Nugget Detection task in English. Our system is language independent, and can outperform conventional methods in event detection accuracy without parameter tuning specific to the dataset. This advantage is enabled by combining the following technologies: (1) muti-channel encoding of target token as a modification of conventional single window method to enhance the prediction accuracy of phrase position, (2) stochastic voting to synthesize reliable prediction results based on multiple predictions generated by multiple prediction models.

1 Overview

In Event Nugget Detection task, a system detects event phrases of 18 determined event subtypes from raw text data. As this task includes detection of event phrase position (in character offset) and recognizing event realis status, we focus on enhancing the total performance of our system. Figure 1 shows the overview of our event nugget detection system in KBP2017. The system first generates a sequence of tokens from a set of input documents in Tokenizer module. Then, Event Detection module applies binary classification to each of single tokens to obtain hypotheses of event tokens. Both of Event Classification module and Realis Status Classification module input the obtained event tokens and output a pair of event subtype and event realis status for each token as a combined event nugget information.

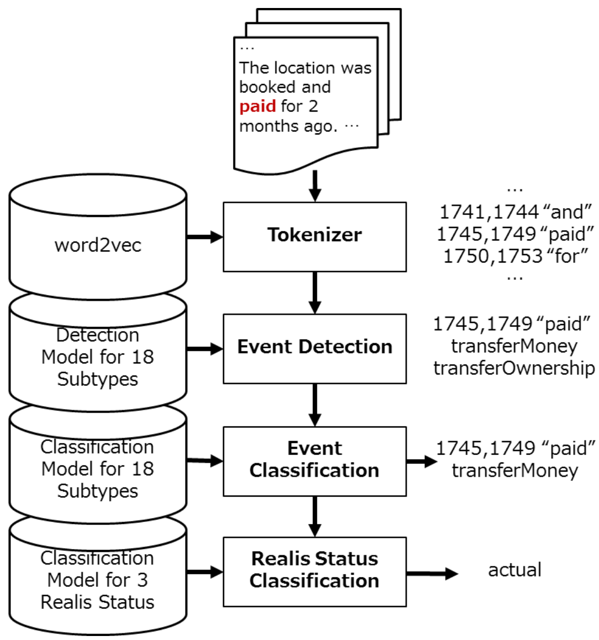

Figure 1: Our system for Event Nugget Detection

The role of Event Classification is to decide a unique subtype to each token by disambiguating the multiple event subtypes assigned by binary classifications of subtypes in the previous module. The classification model in Realis Classification module assumes that all the input tokens are event relevant tokens and each of those tokens should be assigned one of the predefined event realis statuses to make an event nugget. Beside model based classification, Realis Classification uses a heuristic rule to assign default realis status "other" when the classification model fails to assign any of realis status to a token. All these detection and classification modules use neural networks as classification models and the models were trained from development dataset automatically.

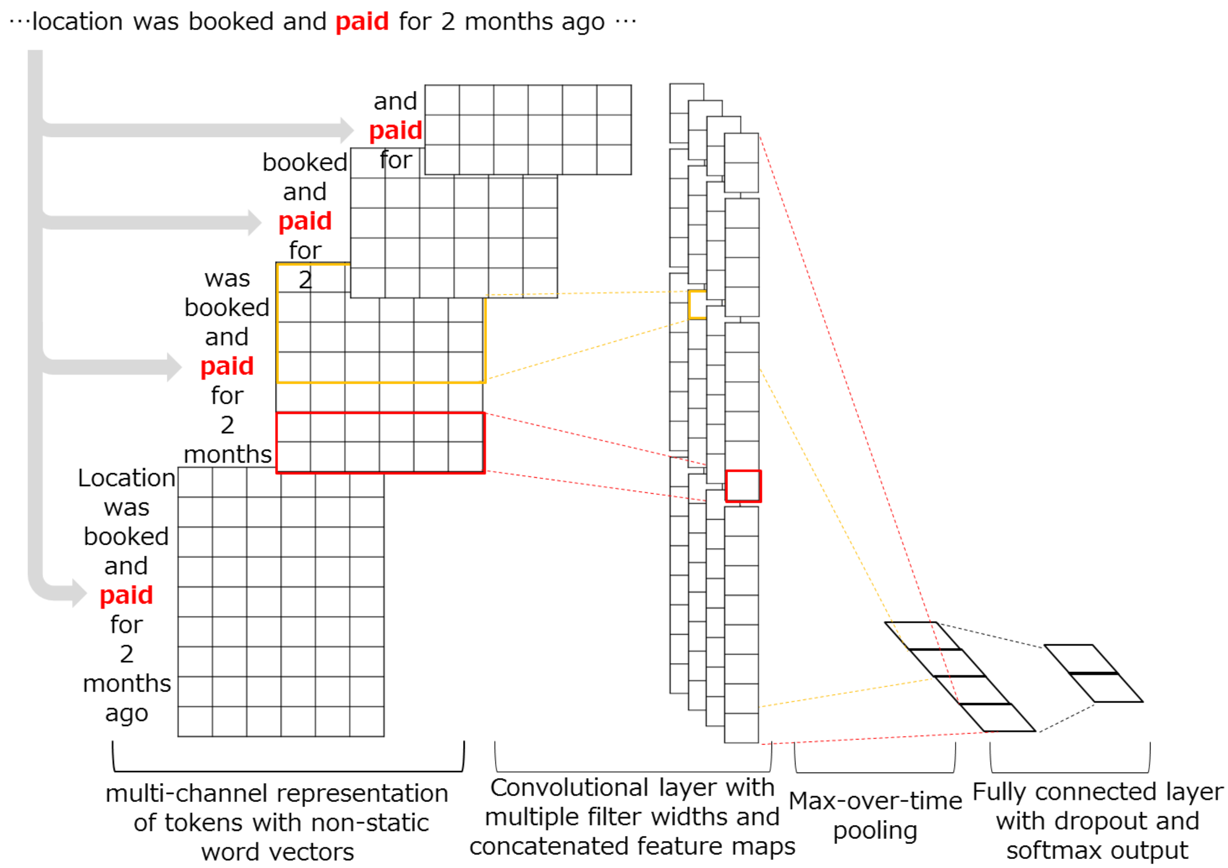

Figure 2: Convolutional network with multi-channel encoding

2 Detection Model

Event Detection module inputs a sequence of tokens generated by Tokenizer and applies binary classifications to individual single tokens to detect hypotheses of 18 event subtypes. As a classification model for event detection, we use a CNN architecture for sentence classification of Kim (2014), a slight modification of Collobert et al. (2011), is used to classify single tokens. For encoding tokens, we use k-dimensional word vector representation obtained by word2vec (Mikolov, 2013).

In this paper, we propose a classification model using multi-channel encoding of target token as a modification of conventional single window method to enhance the prediction accuracy of phrase position.

2.1 Single Window (SW: conventional)

Single Window introduces a window of size 2w+1, limit the context to a window size by trimming longer sentences and padding shorter sentences with a special token when necessary.

Let x_i be a k-dimensional word vector corresponding to i-th word in the window. Then, an encode of event mention candidate can be represented as a matrix in the following formula,

$$x_{-w:w} = x_{-w} \oplus \ldots \oplus x_0 \oplus \ldots \oplus x_w. \qquad (1)$$

Where, $\oplus$ is the concatenation operator (Kim, 2014).

The encoding of target token using single window can deal with longer context in event detection of tokens by setting the window size larger. However, the accuracy of predicting the position of event token will be degraded by using large window size. This trade-off problem is due to the disagreement between the unit of classification, single token, and the length of context considered in classification.

To avoid this problem of using conventional single window, we introduce multi-channel encoding framework that considers context information without losing position information of event token.

2.2 Gradational Windows (GW)

The first proposed method is to generate a multi-channel encoding of a token by using a set of multiple windows with different window sizes and a center on the token.

By using a set of wind sizes {2w+1, 2w-1, …, 1},we obtain w+1 encoding representations for the target token in the following.

$$\{x_{-w:w}, x_{-w+1:w-1}, \ldots, x_{-1:1}, x_0\}. \qquad (2)$$

In the training process of network, we use the encoding representation as w+1 multi-channel inputs for the convolutional layer of the neural network as shown in Figure 2.

We concatenate multiple feature maps generated from multiple encoding representations, and input into fully connected layer through max-over-time pooling. In this way, we train the neural network as an integrated network with multi-channel encoding of target token.

2.3 Pseudo Dependencies (PD)

The second proposed method is to introduce pseudo dependency relations between the target token and all the other tokens in a distance of w and generate a multi-channel encoding of the target token by using the pseudo dependencies. Then, we obtain 2w encoding representations of the target token in the following.

$$\{x_{-w} \oplus x_0, \ldots, x_{-1} \oplus x_0, \ldots, x_0 \oplus x_w\}. \qquad (3)$$

In the training process of network, we use the encoding representation as 2w multi-channel inputs for the convolutional layer of the neural network in the same way as in Gradational Window (in Figure 2). In this way, we train the neural network as an integrated network with multi-channel encoding of target token.

2.4 Stochastic Voting

In Event Detection module, neural network generates a probability of $P_{model}(subtype|w)$ for each prediction of a token as an event token of a subtype. By interpreting the probability as a reliability measure, we can obtain a more reliable prediction by synthesizing from multiple predictions (results of binary classification) from different models.

The following formula shows our approach of model selection.

$$\begin{aligned} & model \\ & = \operatorname*{argmax}_{model}\{\max[P_{model}(subtype|w), \\ & 1 - P_{model}(subtype|w)]\} \end{aligned} \qquad (4)$$

This approach selects the model with the maximum probability (dealing with both positive and negative prediction) for every prediction of target token.

3 Classification Model

3.1 Event Classification

As binary classification is used, Event Detection assigns multiple event subtypes to some tokens. Therefore, we need a disambiguation process for those event tokens to disambiguate among multiple event subtypes.

Another reason of introducing event classification after event detection is that we can handle some pair of event subtypes that are difficult to distinguish by the detection model. For these reason, we train a classification model from a dataset consists of event tokens only to obtain better disambiguation performance.

Here, we use the following formula to obtain more reliable prediction for subtype of a token based on multiple predictions generated by different models.

$$\begin{aligned} & subtype \\ & = \operatorname*{argmax}_{subtype}\left\{\max_{model} P_{model}(subtype|w)\right\} \end{aligned} \qquad (5)$$

3.2 Realis Classification

The approach introduced in Event Classification is also applicable to Realis Classification. We use the following formula to obtain more reliable prediction for realis status of a token based on multiple predictions generated by different models.

$$\begin{aligned} & r\_status \\ & = \operatorname*{argmax}_{r\_status}\left\{\max_{model} P_{model}(r\_status|w)\right\} \end{aligned} \qquad (6)$$

Table 1: F-1 Scores of Event Detection Models using SW, GW, and PD

Subtypes used in KBP2017	Gradational Windows	Pseudo Dependencies	Single Window (conventional)		
	size=1,3,5,7,9,11	size=11	size=1	size=7	size=11
attack	**0.614**	0.580	0.590	0.522	0.248
demonstrate	**0.780**	0.674	0.730	0.579	0.211
broadcast	0.352	0.336	**0.370**	0.307	0.057
contact	0.322	**0.349**	0.224	0.256	0.065
correspondence	0.265	0.219	0.229	**0.287**	0.117
meet	**0.488**	0.485	0.383	0.343	0.165
arrestjail	0.732	0.736	**0.755**	0.618	0.290
die	0.682	0.683	**0.693**	0.589	0.262
injure	0.481	**0.600**	0.500	0.291	0.165
artifact	**0.529**	0.387	0.390	0.082	0.057
transportartifact	0.311	**0.485**	0.329	0.210	0.074
transportperson	**0.591**	0.585	0.460	0.532	0.254
elect	**0.580**	0.552	0.000	0.316	0.155
endposition	0.692	**0.729**	0.590	0.492	0.218
startposition	**0.487**	0.482	0.404	0.266	0.280
transaction	0.175	**0.197**	0.182	0.070	0.053
transfermoney	**0.620**	0.591	0.553	0.542	0.265
transferownership	0.521	**0.545**	0.510	0.455	0.210
Macro Average	0.512	0.512	0.438	0.375	0.175

4 Datasets and Experimental Setup

Table 2 shows all the dataset used to build our event nugget detection system. They are all provided by LDC, and we used English source articles and corresponding annotations related to event nuggets to build a development dataset.

Table 2: Datasets Used for Development

Catalog ID	Title
LDC2017E02	TAC KBP Event Nugget Detection and Coreference - Comprehensive Training and Evaluation Data 2014-2016
LDC2016E31	DEFT Rich ERE English Training Annotation R3
LDC2015E68	DEFT Rich ERE English Training Annotation R2 V2
LDC2015E29	DEFT Rich ERE English Training Annotation V2

All the detection models and classification models are developed only from the development dataset.

4.1 Hyper-parameters and Training

With regard to the hyper parameters of convolutional neural network, we use the same set of hyper-parameters for all the detection and classification models. We use filter windows of 2, 3, 4, 5 with 100 feature maps each, dropout rate of 0.5, and mini-batch size of 50.
Training is done through stochastic gradient descent over shuffled mini-batches with the Adadelta update rule (Zeiler, 2012).
We do not perform any data specific tuning other than early stopping (randomly selected 10% of the training data is used for evaluation).

Table 3: F-1 Scores of Event Classification Models using SW, GW, and PD

Subtypes used in KBP2017	Gradational Windows	Pseudo Dependencies	Single Window
	size=1,3,5,7,9,11	size=11	size=1
attack	0.805	**0.817**	0.805
demonstrate	**0.916**	0.903	0.875
broadcast	**0.706**	0.655	0.538
contact	0.528	**0.546**	0.508
correspondence	**0.508**	0.456	0.283
meet	0.648	**0.697**	0.560
arrestjail	**0.969**	0.928	0.899
die	0.789	0.759	**0.803**
injure	0.639	**0.721**	0.625
artifact	0.875	**0.897**	0.737
transportartifact	0.385	0.389	**0.426**
transportperson	**0.835**	0.831	0.724
elect	0.939	**0.984**	0.884
endposition	**0.877**	0.827	0.776
startposition	**0.787**	0.711	0.646
transaction	0.357	0.118	**0.414**
transfermoney	0.800	**0.835**	0.739
transferownership	0.604	**0.670**	0.614
Macro Average	0.720	0.708	0.659

Table 4: F-1 Scores of Realis Status Classification Models using SW, GW, and PD

Subtypes used in KBP2017	Gradational Windows	Pseudo Dependencies	Single Window
	size=1,3,5,7,9,11	size=11	size=1
actual	**0.897**	0.889	0.784
generic	**0.734**	0.729	0.400
other	**0.779**	0.758	0.517
Macro Average	0.803	0.792	0.567

4.2 Pre-trained Word Vectors

As word vectors, we use the publicly available word2vec vectors that were trained on 100 billion words from Google News. The vectors have dimensionality of 300 and were trained using the continuous bag-of-words architecture (Mikolov et al., 2013). Words not presented in the set of pre trained words are initialized randomly.

Then, the pre-trained word vectors from word2vec are fine-tuned via back-propagation for each data set using the non-static model (Kim, 2014).

5 Results and Discussion

5.1 Detection Model in Development Set

To develop our event detection models for the KBP2017 official submission, we randomly selected 10% of tokens from the development set as an evaluation set for event detection task. The remaining 90% of tokens was used for training detection models. Then, we developed event detection models using three methods, i.e., Single Window, Gradational Windows, and Pseudo Dependencies. The performance of these models in event detection are evaluated in F-1 score and shown in Table 1.

With regard to window size, size of 11 was commonly used for all the three methods. As for Single Window, we trained detection models for window size of 1 and 7 additionally.

By comparing the Macro Average values for the methods, you see both Gradational Windows and Pseudo Dependencies outperform Single Window by about 7% points.

Among the Macro Average values of Single Window, size 1 achieved the highest F-1 score, the score of 7 is in the second, and the score of size 11 is the lowest among all the results.

By comparing F-1 scores for individual event subtypes, you see the number of subtypes in which Gradational Windows achieved the best score is 8, while Pseudo Dependencies won 6 subtypes, and Single Window with size of 1 won 3 subtypes.

As a consequence, both of our proposed method with multi-channel encoding, Gradational Windows and Pseudo Dependencies, outperformed Single Window. However, there are some cases where Single Window achieves better score than the other models in some event subtypes.

5.2 Classification Model in Development Set

To develop our event and realis classification models for the official submission, we prepared a subset of development dataset consists of only event tokens to enhance disambiguation performance. Then, we randomly selected 10% of tokens from the subset as an evaluation set for event and realis classification task. The remaining 90% of tokens was used for training classification models.

Using the dataset, we developed event and realis classification models using three methods, i.e., Single Window, Gradational Windows, and Pseudo Dependencies. The performance of these models in event classification and realis classification are evaluated in F-1 score and shown in Table 3 and Table 4 respectively.

The scores of event classification (in Table 3) are observed higher than those of event detection (in Table 1), because the positive rate of evaluation dataset is higher in classification task. However, the overall trend observed in event detection and event classification are quite similar. Both Gradational Windows and Pseudo Dependencies outperform Single Window in overall performance. However, there are some cases where Single Window achieves better score than the other models in some event subtypes.

On the contrarily, results of realis status classification shows a clear superiority of Gradational Windows and Pseudo Dependencies against Single Window. The score of Gradational Window is the best in all realis statuses.

5.3 Official Submission in KBP2017

According to the evaluation of Event Detection, Event Classification, and Realis Classification on development dataset, we decided to use a combination of three models, i.e., Single Window (size of 0), Gradational Window, and Pseudo Dependencies for all the detection and classification tasks. We submitted two system using different way of combining the three methods, SW, GW, and PD as explained in Table 5.

Table 5: Submitted Systems

ID	Specification
System 1 (dsln_nlptt1)	Micro combination of SW, GW, and PD by Stochastic Voting for each token.
System 2 (dsln_nlptt2)	Macro combination of SW, GW, and PD by selecting F-1 best model for each subtype and realis status.

The F-1 scores of KBP2017 official results for the systems are shown in the following Table 6.

Table 6: Official Results of KBP2017 (F-1 scores)

System	Plain	Mention Type	Realis	Type & Realis
1	56.12	48.56	42.47	36.81
2	53.94	46.59	41.29	35.41

Here, "Plain" means the performance of event detection without considering their subtypes. The F-1 scores of Plain shows that system 1 outperform system 2 in all the scores by 2.18% point. This demonstrates the effectiveness of Stochastic Voting used in system 1.

System 1 outperforms system 2 in all the scores. "Type & Realis" means the overall performance of

Event Nugget Detection task. The score of system 1 in Type & Realis is higher than that of system 2 by 1.4% point.

From these results, the effectiveness and advantage of Stochastic Voting and Classification using multiple prediction results of different models is clearly demonstrated.

Reference

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

Ye Zhang, Byron Wallace. 2017. A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. IJCNLP.

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

Teruko Mitamura, Zhengzhong Liu, and Eduard Hovy. 2016. Overview of TAC-KBP2016 Event Nugget Track. In TAC.

Thien Huu Nguyen, Adam Meyers, and Ralph Grishman. 2016. New York University 2016 System for KBP Event Nugget: A Deep Learning Approach. In TAC.

M. Zeiler. 2012. Adadelta: An adaptive learning rate method. CoRR abs/1212.5701.

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuglu, P. Kuksa. 2011. Natural Language Processing (Almost) from Scratch. Journal of Machine Learning Research12.