KYOTOU at TAC KBP 2017 Event Track: Neural Network-based Event Sequence Classification Model

Tomohide Shibata and **Hongkai Li** and **Tomohiro Sakaguchi** and **Sadao Kurohashi**
Graduate School of Informatics, Kyoto University
Yoshida-honmachi, Sakyo-ku, Kyoto, 606-8501, Japan
{shibata, li, sakaguchi, kuro}@nlp.ist.i.kyoto-u.ac.jp

Abstract

We describe Kyoto University's system in the Event Sequencing task at TAC KBP 2017 (Team ID: KYOTOU). We take a neural network based approach for the event sequence classification with external knowledge about events. Bi-directional GRU (Gated Recurrent Unit) is first applied for the input encoding, and then Multi-layer perceptron (MLP) takes the representations of two events as well as other features as an input, and outputs an event sequence class. In order to eliminate the class imbalance, an undersampling technique is used. Our system achieved F-score of 12.6 for the official evaluation, which ranked first among two teams.

1 Overview

The Event Sequencing task is a task to detect temporal relations of events, focusing on a stereotypical sequence of events that occur as part of a whole event. In this task, the two kinds of links are annotated, *subevent* (parent-child) and *after* link[1]. In the following example, there are subevent links in "*attacked* → *hit*" and "*attacked* → *stabbed*", and an after link in "*hit* → *stabbed*".

(1) The 17 year old high school student was *attacked* on the street yesterday. He was *hit* and then *stabbed* with a knife.

There are several approaches to estimate a temporal relation between events. One is a feature based machine learning approach, which utilizes hand-crafted rules, event attributes and external resources (D'Souza and Ng, 2013; Chambers et al., 2014). Another is a neural network based approach, which performs comparable without using hand-crafted features or external knowledge (Cheng and Miyao, 2017; Choubey and Huang, 2017). We take a neural network based approach for the event sequence classification with external knowledge about events.

Among the combinations of all events, only a small portion of the relations have a temporal relation, mostly *NONE*. In order to eliminate this class imbalance, an undersampling technique is used. Our system achieved F-score of 12.6 for the official evaluation, which ranked first among two teams.

2 Related Work

Understanding temporal information in text is important for many NLP tasks such as question answering, summarization and information extraction. Temporal relation classification, estimating relations between event-event pair or event-time pair, is one of the most important task to understand temporal information. The TimeBank Corpus (Pustejovsky et al., 2003) and TempEval competitions (Verhagen et al., 2007; Verhagen et al., 2010; UzZaman et al., 2013) have contributed to the development of classification techniques.

Feature based approaches use hand-crafted rules, event attributes and external resources such as WordNet (Miller, 1995) and VerbOcean (Chklovski and Pantel, 2004). Mani et al. (2006) built a Maximum Entropy classifier using annotated features in corpus

[1] http://cairo.lti.cs.cmu.edu/kbp/2017/event/TAC_KBP_2017_Event_Coreference_and_Sequence_Annotation_Guidelines_v1.1.pdf

and outperformed rule-based approaches. D'Souza and Ng (2013) combined rule-based and data-based approaches, using lexical relation, semantic and discourse features. Chambers et al. (2014) introduced a sieve-based architecture for event ordering.

Neural network based approaches perform comparable without using hand-crafted efforts or external resources. Since the dependency path based neural network methods perform well in relation extraction tasks (Socher et al., 2011; Xu et al., 2015a; Xu et al., 2015b), the techniques are introduced to the temporal relation classification. Choubey and Huang (2017) proposed a BiLSTM model to classify intra-sentence events. They generate three sequences of dependency path: the word sequence, the POS tag sequence and the dependency relation sequence. They apply BiLSTMs for each sequence and concatenate the outputs to estimate the relationship. Cheng and Miyao (2017) applied BiLSTM to dependency paths, and estimated cross-sentence relationships. To estimate the relationship between two entities, they make two sequences, each entity to the common root of the entities, and apply BiLSTMs to them. For each sequence, the concatenation of word, POS and dependency relation embeddings is used.

3 Model

The input of the system is the (gold) event pairs e_1 and e_2 (e_2 appears after e_1 in a document). The annotated directed links are normalized to an event sequence class, which is a relation from e_2 to e_1, for ease of the direction handling. The output of the system is a sequence class, which includes *BEFORE*, *AFTER*, *PARENT*, *CHILD*, and *NONE*. Figure 1 shows the architecture of the system.

3.1 Network Architecture

Let $\boldsymbol{x}_i$ be the embedding corresponding to the i-th word, which is represented as a concatenation of word embedding and POS (part-of-speech) embedding. First, to obtain the contextual word representation, bi-directional GRU (Gated Recurrent Unit) (Chung et al., 2014) is applied to a sequence of words for each sentence as follows:

$$\overrightarrow{\boldsymbol{h}}_i = \overrightarrow{GRU}(\boldsymbol{x}_i, \overrightarrow{\boldsymbol{h}}_{i-1}), \quad (1)$$

$$\overleftarrow{\boldsymbol{h}}_i = \overleftarrow{GRU}(\boldsymbol{x}_i, \overleftarrow{\boldsymbol{h}}_{i+1}), \quad (2)$$

and the representation for the i-th word is a concatenation of these hidden states as follows:

$$\boldsymbol{h}_i = [\overrightarrow{\boldsymbol{h}}_i; \overleftarrow{\boldsymbol{h}}_i]. \quad (3)$$

The input vector $\boldsymbol{v}_{in}$ for the classification is a concatenation of $\boldsymbol{v}_{e_1}$ and $\boldsymbol{v}_{e_2}$ (the representations of e_1 and e_2), a path embedding $\boldsymbol{v}_p$ and a feature vector $\boldsymbol{v}_f$ of e_1 and e_2. A word sequence between e_1 and e_2 can be a clue for the classification. GRU reads the word sequence, and the final hidden layer is adopted as the path embedding. The feature vector includes the followings:

- Event subtype of e_1 and e_2

  The events in this task are based on the definition in DEFT Rich ERE Event Annotation Guidelines[2], and type and subtype are annotated for each event. There are 8 types, such as *Business* and *Conflict*, and 38 subtypes, such as *Declare–Bankrupt* and *Attack*. The (gold) event subtypes of e_1 and e_2 are utilized.

- Realis of e_1 and e_2

  The (gold) realis status (ACTUAL, GENERIC and OTHER) of e_1 and e_2 is used.

- Sentence distance between e_1 and e_2

  A binary vector of sentence distance between e_1 and e_2 is used.

- Exact match of lemmas between e_1 and e_2

- Existence of a semantic relation between e_1 and e_2 in external knowledge

  The semantic relation of event-pair obtained from external knowledge is used. The details are described in Section 3.2.

The input vector $\boldsymbol{v}_{in} \in \mathbb{R}^{d_{in}}$ (d_{in} denotes the dimension of the input vector) is fed into a Multi-layer

[2] https://tac.nist.gov/2016/KBP/guidelines/summary_rich_ere_v4.2.pdf

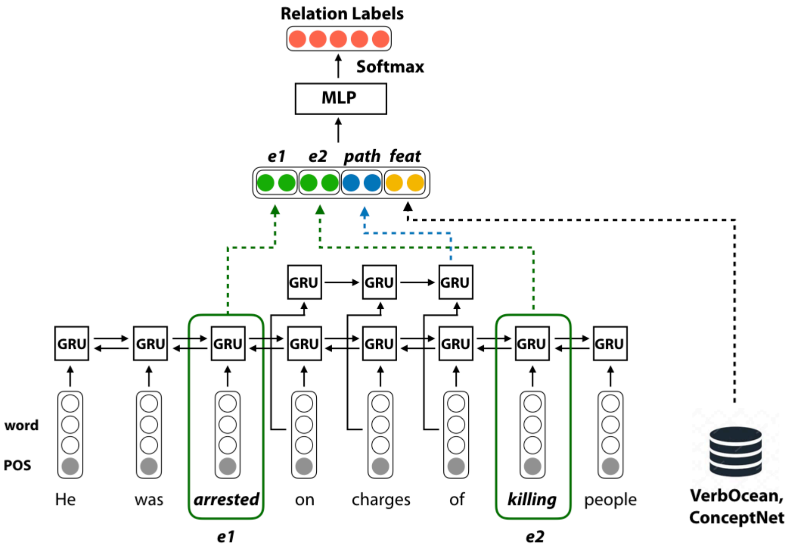

Figure 1: The system architecture.

perceptron (MLP). A hidden state $\boldsymbol{h}_c$ (for the classification) is calculated as follows:

$$\boldsymbol{h}_c = f(W_1 \boldsymbol{v}_{in}) \tag{4}$$

where $W_1 \in \mathbb{R}^{d_{h_c} \times d_{in}}$ (d_{h_c} denotes the dimension of the hidden layer) is a weight matrix from the input layer to the hidden layer, and f is an activation function ($tanh$ is used in our experiments). The predicted probability distribution $\boldsymbol{y}$ is calculated as follows:

$$\boldsymbol{y} = softmax(W_2 \boldsymbol{h}_c) \tag{5}$$

where $W_2 \in \mathbb{R}^{d_{out} \times d_{h_c}}$ (d_{out} denotes the number of event class) is a weight matrix from the hidden layer to the output layer. The objective is to minimize the cross entropy between predicted and true distributions.

3.2 External Knowledge

Since the training data is small, external knowledge of event pairs is necessary. Two resources, VerbOcean (Chklovski and Pantel, 2004) and ConceptNet (Speer and Havasi, 2012), are utilized. In this system, whether the relationships described in external knowledge exist between e_1 and e_2 is represented as a binary vector.

3.2.1 VerbOcean

VerbOcean is a resource of fine-grained semantic relations between verbs, which is extracted from Web using a semi-automatic method. There are five relations, *similar*, *stronger-than*, *opposite-of*, *can-result-in* and *happens-before*, and about 22,000 relations are extracted. For example, the pair of *attack* and *destroy* has a *happens-before* relation.

These semantic relations can be a clue in the task. In the following example, there is a *happens-before* relation between *arrested* and *extradited*, and it is a clue to estimate a *BEFORE* class.

(2) [...] you ask them to ***arrest*** that person and have them ***extradited***.

In the same way, *similar* relation between the events in the following example would be a clue to estimate a *PARENT* class.

(3) I called the RE's office and ***spoke*** with our nurse. She ***said*** a lot of couples opt to take a break because it is very stressful .

3.2.2 ConceptNet

ConceptNet provides a large semantic graph that describes general human knowledge, and 21 interlingual relations are defined, such as *IsA* and *PartOf*. In this system, three relations which are related

	all			after			subevent		
	P	R	F	P	R	F	P	R	F
dev	16.8	19.6	18.1	16.4	19.4	17.8	22.8	17.5	19.8
test	14.8	12.5	13.6	14.5	12.4	13.3	18.1	9.40	12.4

Table 1: Experimental results (before official evaluation).

to events, *HasSubevent*, *HasLastSubevent* and *HasFirstSubevent*, are used as a binary vector. In the following example, the semantic relation *HasSubevent* between the event pair is a clue to estimate a *PARENT* class.

(4) In 1963, Sen. Arnon de Mello ***shot*** dead a fellow legislator on the Senate floor, only to escape imprisonment, since the ***killing*** was considered an accident because he was aiming at another senator.

3.3 Training

Adam (Kingma and Ba, 2014) is adopted as the optimizer, and weight decay is used for regularization (0.0001). Dropout is applied for Multi-layer Perceptron. The word embeddings are initialized using pre-trained word embeddings[3], whose dimension is 300, and POS embeddings are randomly initialized, whose dimension is 10. The dimension of hidden layer is 100.

Since the combination of event pair is enormous, event pairs within three sentences are targeted. Event pairs that have a gold coreference relation are not utilized for training and testing.

The number of *NONE* class instances is much larger compared to other classes. To handle the class imbalance, an undersampling method is used; a part of *NONE* class instances at a specified ratio are used (the rest of instances are discarded). The undersampling ratio is determined by using a development set.

Our system is implemented using *Chainer* (Tokui et al., 2015). Stanford CoreNLP[4] is used for tokenization, sentence segmentation, lemmatization and POS tagging. When looking up VerbOcean and ConceptNet, a verbal noun is converted to its corresponding noun using NLTK (Natural Language Processing Toolkits) (e.g., negotiation → negotiate).

undersampling ratio	P	R	F
1.00	43.0	0.242	0.480
0.10	27.4	5.12	8.63
0.05	21.6	9.59	13.3
0.03	19.0	16.5	17.6
0.02	16.8	19.6	**18.1**
0.01	8.52	24.4	12.6

Table 2: Experimental results for development set where undersampling ratio varies.

4 Experiments

4.1 Corpus

We used the corpus LDC2016E130 for our experiments, which consists of 158 training documents and 202 testing documents. 30 documents among the training documents were used for the development. For the official evaluation, the system was trained using the same corpus, and submitted our three runs.

4.2 Experimental Result

Table 1 shows our experimental results (before the official evaluation), where the undersampling ratio was set to 0.02. The evaluation measures are precision, recall, and F-measure by the official scorer provided by the organizers.

Table 2 shows our results for development set where the undersampling ratio varies. When all possible classes are used, that is, when the undersampling ratio is 1.0, F-score is 0.48, but when the undersampling ratio is 0.02 (98% of *NONE* classes are randomly abandoned), it becomes 18.1. The table shows that the recall is improved by reducing undersampling ratio.

[3] Downloaded from `https://nlp.stanford.edu/projects/glove/`.

[4] `https://stanfordnlp.github.io/CoreNLP/`

undersampling ratio	all			after			subevent		
	P	R	F	P	R	F	P	R	F
0.02 (RUN2)	13.3	**12.0**	**12.6**	7.5	**15.0**	**10.0**	16.9	**11.0**	**13.3**
0.03 (RUN1)	15.5	7.7	10.2	12.5	4.4	6.5	15.8	8.5	11.1
0.05 (RUN3)	**23.0**	4.2	7.1	**15.7**	4.8	7.4	**26.6**	4.2	7.1

Table 3: Experimental results (official evaluation).

4.3 Official Evaluation Result

We submitted the following three runs for the official evaluation where a undersampling ratio just varied (Run1: 0.03, Run2: 0.02, Run3: 0.05). Table 3 shows our official evaluation result. Run2 performed the best, and we ranked first among two teams.

5 Discussion

In the following example, the system correctly outputted *BEFORE* class between the event pair.

(5) Biros ***killed*** the 22 year old Engstrom near Warren in 1991 after offering to drive her home from a bar, then ***scattered*** her body parts in Ohio and Pennsylvania .

Although there is no relation described in external knowledge between events, it is supposed that the word "then" between events, which is considered by the path embedding, could be used for a clue.

In the following example, while the gold class is *NONE*, the system outputted *BEFORE* class.

(6) That way, you are completely finished with the car ***payment***, are only out the difference (instead of the entire amount left that is owed), and have ***purchased*** something cheap in cash.

In the following example, the system did not output the correct label *PARENT* but outputted *NONE*.

(7) During testimony last month Al Jayouzi threw his shoes at prosecutors when the ***death*** of his comrades during a fire ***fight*** was discussed.

In the above example, the relation is not described in the external knowledge. Thus, we are planning to acquire event knowledge from a large raw corpus, and integrate it into our system.

	F	Δ
Our method	18.1	
- VerbOcean	15.5	-2.6
- ConceptNet	17.0	-1.1
- GRU	16.1	-2.0
w/ LSTM	16.1	-2.0

Table 4: Ablation study on the development set.

To reveal the importance of each clue for the classification, each clue was ablated. Table 4 shows the result on the development set. We found that external knowledge (both VerbOcean and ConceptNet) was effective. "- GRU" represents GRU was not used, and just word embeddings were used for the word representation. GRU was effective for capturing the context. "w/ LSTM" represents LSTM was used instead of GRU. The performance of LSTM was worse than one of GRU. That is because LSTM has more parameters to train in comparison with GRU, and the evaluation corpus is relatively small for the parameters training.

6 Conclusion

In this paper, we have described the Kyoto University's system in the Event Sequencing task at TAC KBP 2017. The system is based on neural network approach using external knowledge. Since the most of the classes are *NONE*, an undersampling method was used. Our system achieved F-score of 12.6 for the official evaluation, which ranked first among two teams.

Acknowledgment

This work was supported by JST CREST Grant Number JPMJCR1301, Japan.

References

Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. 2014. Dense event ordering with a multi-pass architecture. *TACL*, 2:273–284.

Fei Cheng and Yusuke Miyao. 2017. Classifying temporal relations by bidirectional lstm over dependency paths. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–6. Association for Computational Linguistics.

Timothy Chklovski and Patrick Pantel. 2004. VerbOcean: Mining the web for fine-grained semantic verb relations. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-04)*, pages 33–40.

Prafulla Kumar Choubey and Ruihong Huang. 2017. A sequential model for classifying temporal relations between intra-sentence events. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1803. Association for Computational Linguistics.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio, 2014. *Empirical evaluation of gated recurrent neural networks on sequence modeling*.

Jennifer D'Souza and Vincent Ng. 2013. Classifying temporal relations with rich linguistic knowledge. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 918–927.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Inderjeet Mani, Marc Verhagen, Ben Wellner, Chong Min Lee, and James Pustejovsky. 2006. Machine learning of temporal relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 753–760, Stroudsburg, PA, USA. Association for Computational Linguistics.

George A. Miller. 1995. Wordnet: A lexical database for english. *COMMUNICATIONS OF THE ACM*, 38:39–41.

James Pustejovsky, Patrick Hanks, Roser Saurí, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, David Day Beth Sundheim, Lisa Ferro, and Marcia Lazo. 2003. The timebank corpus. In *Proc. Corpus Linguistics 2003*, pages 647–656.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 151–161, Stroudsburg, PA, USA. Association for Computational Linguistics.

Robert Speer and Catherine Havasi. 2012. Representing general relational knowledge in conceptnet 5. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *LREC*, pages 3679–3686. European Language Resources Association (ELRA).

Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. 2015. Chainer: a next-generation open source framework for deep learning. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*.

Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. 2013. SemEval-2013 Task 1: TempEval-3: Evaluating Time Expressions, Events, and Temporal Relations. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9. Association for Computational Linguistics.

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. 2007. SemEval-2007 Task 15: TempEval Temporal Relation Identification. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 75–80. Association for Computational Linguistics.

Marc Verhagen, Roser Sauri, Tommaso Caselli, and James Pustejovsky. 2010. SemEval-2010 Task 13: TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 57–62. Association for Computational Linguistics.

Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2015a. Semantic relation classification via convolutional neural networks with simple negative sampling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 536–540, Lisbon, Portugal, September. Association for Computational Linguistics.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015b. Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1785–1794, Lisbon, Portugal, September. Association for Computational Linguistics.