TTI-COIN at SemEval-2017 Task 10: Investigating Embeddings for End-to-End Relation Extraction from Scientific Papers

Tomoki Tsujimura, **Makoto Miwa**, and **Yutaka Sasaki**
COmputational INtelligence Laboratory
Toyota Technological Institute
2-12-1 Hisakata, Tempaku-ku, Nagoya, Aichi, 468-8511, Japan
{sd16420, makoto-miwa, yutaka.sasaki}@toyota-ti.ac.jp

Abstract

This paper describes our TTI-COIN system that participated in SemEval-2017 Task 10. We investigated appropriate embeddings to adapt a neural end-to-end entity and relation extraction system LSTM-ER to this task. We participated in the full task setting of the entity segmentation, entity classification and relation classification (scenario 1) and the setting of relation classification only (scenario 3). The system was directly applied to the scenario 1 without modifying the codes thanks to its generality and flexibility. Our evaluation results show that the choice of appropriate pre-trained embeddings affected the performance significantly. With the best embeddings, our system was ranked third in the scenario 1 with the micro F1 score of 0.38. We also confirm that our system can produce the micro F1 score of 0.48 for the scenario 3 on the test data, and this score is close to the score of the 3rd ranked system in the task.

1 Introduction

Semantic relationships between entities are useful for building knowledge bases and semantic search engines. Their automatic extraction has been widely studied in the natural language processing (NLP) field (Li and Ji, 2014; Miwa and Sasaki, 2014; Miwa and Bansal, 2016). SemEval-2017 Task 10 (Augenstein et al., 2017) deals with relation extraction from scientific papers.

While entity detection and relation extraction have often been treated as separate tasks, several studies show that joint treatment of these tasks can improve extraction performance on both tasks (Li and Ji, 2014; Miwa and Sasaki, 2014). We employed the state-of-the-art neural network-based end-to-end entity and relation extraction system LSTM-ER[1] (Miwa and Bansal, 2016). The model of the system is built on pre-trained word embeddings, and it has a tree-structured bidirectional long short-term memory-based recurrent neural network (LSTM-RNN) layer stacked on a sequential bidirectional LSTM-RNN layer. It predicts entities and relations in an end-to-end manner with shared parameters, and the parameters in word embeddings and both LSTM-RNNs are updated simultaneously during training.

We first checked the applicability of the system in our evaluation. The system was originally developed for ACE (automatic content extraction) corpora, but it does not depend on specific tasks and has high configurability (Miwa and Ananiadou, 2015) since it prepares a separate configuration file, where task specific settings like hyperparameters can be specified. The system was successfully applied to the end-to-end relation extraction setting (scenario 1 in the task) without modifying the original codes in our experiments, although small modifications to the inputs were required. This shows the generality of the system.

Using this system, we also investigated how the pre-trained word embeddings affect the overall performance. Miwa and Bansal (2016) mostly focused on the model architectures and paid little attention to the differences in pre-trained embeddings. Our results show that selecting the appropriate initial embeddings is crucial since changing the pre-trained embeddings greatly affected the overall performance.

2 System description

In this section, we describe our base neural network system and pre-trained word embeddings

[1] https://github.com/tticoin/LSTM-ER

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 985–989,
Vancouver, Canada, August 3 - 4, 2017.

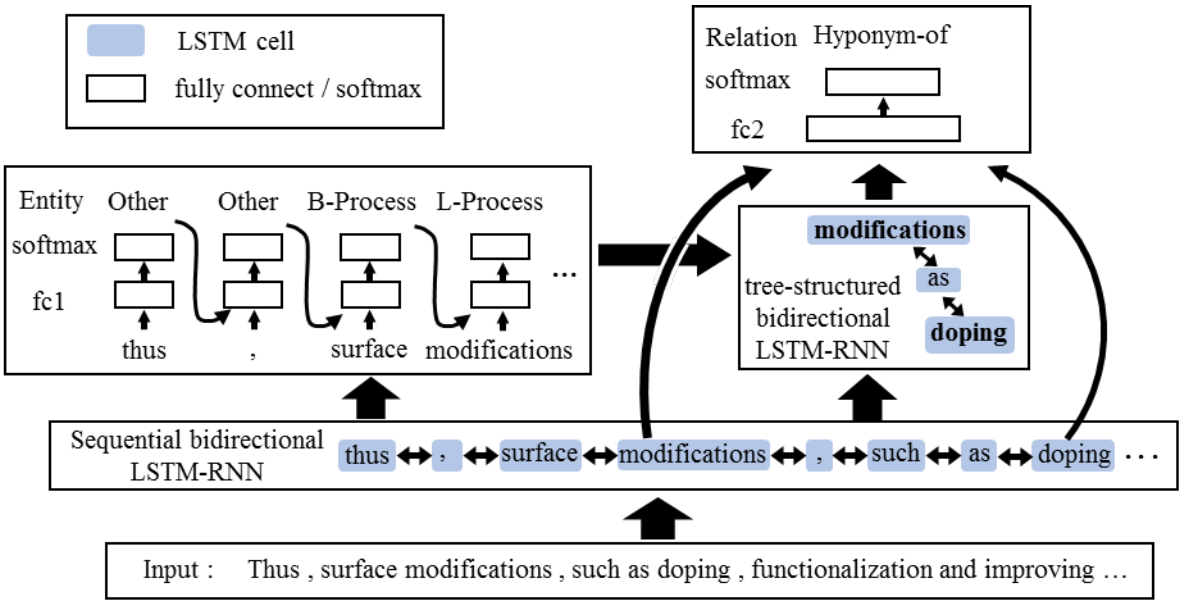

Figure 1: Model overview in extracting a *Hypernym-of* relation between *Process* entities "surface modifications" and "doping". Word and POS embeddings are input to the sequential bidirectional LSTM-RNN.

used for the initialization.

2.1 Base system

We employ the LSTM-ER system that implements a neural network-based model (Miwa and Bansal, 2016). The system has a high configurability, and it can be applied to new tasks with minimum manual efforts. Figure 1 illustrates the overview of the model. The model represents relations using tree-structured LSTM-RNNs and entities using a sequential LSTM-RNN, and stack these LSTM-RNNs to realize their end-to-end extraction.

This model first inputs the concatenation of word embeddings and part-of-speech (POS) embeddings to a sequential bidirectional LSTM-RNN. The output of the sequential bidirectional LSTM-RNN is obtained by concatenating the output of the forward sequential LSTM-RNN and the output of the backward sequential LSTM-RNN.

These outputs together with the embedding of the previous label are passed to the fully connected layer `fc1` with an activation `tanh` function, and then the model calculate the probabilities of the entity labels through a softmax layer. The model uses the BILOU (Begin, Inside, Last, Outside, Unit) scheme (Ratinov and Roth, 2009) for representing entities and assigns the labels to each word. Each entity label is decided in a greedy manner, starting from the beginning of a sentence to the end of the sentence. During this greedy decision, the model avoids illegal label assignments. For example, if a t-th word is determined as *I-Material*, only either *I-Material* or *L-Material* can be selected for the next $(i+1)$-th word and other illegal labels like *B-Task* are not chosen.

After detecting entities, for each target entity pair, the model picks up words that are contained in the shortest path between the last words of the pair. The model passes the outputs of the sequential bidirectional LSTM-RNN together with the entity-label and dependency embeddings relating to these words to a bidirectional tree-structured LSTM-RNN. In the bidirectional tree-structured LSTM-RNN, each cell of the bottom-up tree-structured LSTM-RNN receives multiple children states and calculates the parent state from them, while each cell of the top-down one takes a parent state and calculates the states of the children.

The output of the bi-directional tree-structured LSTM-RNN is obtained by concatenating the output of the root word from the bottom-up tree-structured LSTM-RNN and the outputs of last words of the entities from the top-down tree-structured LSTM-RNN. This output of the bi-directional tree-structured LSTM-RNN is passed to the fully connected layer `fc2` with the activation `tanh` function, together with the outputs of the last words in each target entity in the sequential bi-directional LSTM-RNN. The model then calculates the probability of each relation label through a softmax layer as in entity detection.

2.2 Investigating pre-trained word embedding

Word embeddings are frequently used for the inputs to neural network based models. It is well-known that the performance of neural models can be improved by initializing embeddings with pre-trained embeddings. Word2vec (Mikolov et al., 2013) is a widely-used toolkit to obtain such pre-trained embeddings from raw texts. Word2vec implements two models: continuous bag-of-words (CBOW) and skip-gram. CBOW learns embeddings by predicting the distribution of target word from the surrounding words, while skip-gram learns embeddings by predicting the distribution of each surrounding word from the input word. Ling et al. (2015) introduced the structured skip-gram model[2] based on word2vec in order to consider the ordering of co-occurrence words by incorporating an attention mechanism.

To investigate the effects of the pre-trained word embeddings, we employed two unlabeled data sets: Wikipedia articles[3] and PubMed abstracts[4]. After the preliminary experiments, we compared 100 dimensional word embeddings obtained by the skip-gram model on the Wikipedia articles and those by the structured skip-gram model on the PubMed abstracts. We used these embeddings for initialization, and we fine-tuned these embeddings during training.

3 Evaluation

3.1 Task

SemEval-2017 Task 10 (Augenstein et al., 2017) deals with a relation extraction problem that focuses on detecting entities of *Process*, *Task* and *Material* from research papers and extracting the *Synonym-of* and *Hypernym-of* relationships between the entities. In this task, 500 example paragraphs are extracted from the ScienceDirect open access publications, and they are manually annotated with the entities and their static relationships. 350 examples of them are used for training, 50 for development, and 100 for test. Each paragraph consists of multiple sentences. Each sentence may contain multiple entities and relationships, and an entity may overlap other entities. *Hypernym-of* relations are directed, while *Synonym-of* relations are undirected. The tasks consists of three subtasks: A, B and C. In subtask A, the participants need to detect segmentations of all entities without considering the entity types. In subtask B, the types of entities need to be labeled. Subtask C focuses on extracting relations between the entities. Three scenarios are provided With the subtasks: the system needs to solve subtasks A, B, and C in scenario 1, B and C in scenario 2, and C in scenario 3. We focus on scenarios 1 and 3.

Parameter	dimension
word embeddings	100
POS embeddings	10
bidirectional seq-LSTM	50×2
FC1	100
bidirectional tree-LSTM	100×2
FC2	100

Table 1: Dimensions of layers.

3.2 Evaluation settings

Pre-trained word embeddings: We obtained pre-trained word embeddings by the skip-gram model and the structured skip-gram model with the same setting. We set the window size to 2, the number of negative samples to 10, the down-sampling rate to 1e-4. We also ignored the words that appear less than 26 times. Other parameters are kept as the default of the original word2vec toolkit[5].

POS tagging and dependency parsing: To deal with the data with the LSTM-ER system, we obtained POS tags and dependency trees for all training, development and test data sets by using the Stanford parser (Chen and Manning, 2014). Since the texts contained Unicode, we processed the data as Unicode texts.

Relation modification: We treated each relation as a directed relation. The *Synonym-of* relations are undirected, but we treated them as they always have left-to-right directions.

Out-of-vocabulary words: For the robustness, we treated out-of-vocabulary words as follows. We first counted the frequencies of words in the training dataset. We then picked up words that only appear once in the training dataset and replaced 1% of them with a symbol word "UNK" randomly. Embeddings for the words that do not appear in the vocabulary of pre-trained embed-

[2] https://github.com/wlin12/wang2vec

[3] https://dumps.wikimedia.org/enwiki/20150901/

[4] 2014 MEDLINE/PubMed baseline distribution

[5] https://code.google.com/archive/p/word2vec/

Model	Development			Test		
	A	A,B	C	A	A,B	C
End2end (Wikipedia)	0.55	0.46	0.37	0.49	0.36	0.20
End2end (PubMed)	0.58	0.50	0.39	0.50	0.39	0.21
Relation (Wikipedia)	-	-	0.50	-	-	0.43
Relation (PubMed)	-	-	0.52	-	-	0.48 (0.1)

Table 2: Micro F1 scores on the development and the test dataset for three task settings: subtask A, subtask A,B, and subtask C. The number in the parentheses for Rel (PubMed) shows our official score.

Model	Development			Test		
	A	A,B	C	A	A,B	C
structured skip-gram (PubMed)	0.58	0.50	0.39	0.50	0.39	0.21
skip-gram (PubMed)	0.57	0.48	0.36	0.51	0.39	0.22
skip-gram (Wikipedia)	0.55	0.46	0.37	0.49	0.36	0.20

Table 3: Results on the end-to-end model with several pre-trained word vectors.

dings and are not replaced with "UNK" are initialized with random values. The embedding of the "UNK" word is also initialized with random values. We also replaced all the out-of-vocabulary words in the development and test datasets with the "UNK" word.

Nested entities: Entities that were inside of other entity were ignored. The ignored entities were not used as training examples since the base model gives only one entity label to each word. Our system thus did not predict internal entities on the development and test datasets.

Hyper-parameter tuning: We tuned the dimensions of the embeddings and layers, the rates of the dropout, the coefficient of L2 generalization, and the learning late in a greedy manner. We used both the training and development data sets to train the final models for testing. These parameters were tuned by modifying the configuration file of the LSTM-ER system. Table 1 summarizes the dimensions of word/POS embeddings and layers we used for all models.

3.3 Results

Table 2 shows the F1 scores on the development and test datasets for each subtask. We show our (unofficial) evaluation results of our system for the relation classification task (scenario 3)[6]. In all the evaluations, the results with word embeddings obtained by the structured skip-gram model on the PubMed abstracts were better than those by the skip-gram model on the wikipedia abstracts.

Table 3 shows the results between the models using several pre-trained word embeddings.[7] When comparing the training models of pre-trained embeddings on the PubMed abstracts, our system with embeddings by the structured skip-gram model shows better performance than the system with those by the skip-gram model on the develop dataset does, but this is opposite for the test dataset. As for the difference on the training corpora for pre-training using the skip-gram model, the system with embeddings on PubMed shows consistently higher performance on the entities than the system with those on the Wikipedia articles, but there was no consistent performance change for the relations.

There were 11,026 kinds of lowercased words in the 500 examples in the data set in practice. The numbers of out-of-vocabulary words that were not initialized with the pre-trained embeddings were 2,984 for the Wikipedia dataset and 1,697 for the PubMed dataset. The PubMed abstracts are scientific articles in a different domain like the shared task data sets, and this may be one of the reasons why the pre-trained embeddings on PubMed covers more words than those on Wikipedia. In addition, the similarity in the writing between PubMed and ScienceDirect articles may lead the model to fit to the task on entities.

[6]We got this low result due to our mistakes in converting the results into the task format.

[7]We got results on the model with word embeddings by applying the skip-gram model to the PubMed abstracts after the competition.

4 Conclusion

We participated SemEval-2017 Task 10 with the end-to-end entity and relation extraction system proposed by Miwa and Bansal (2016). We successfully applied the system to the task without modifying the codes. We improved all F1 scores by replacing pre-trained word embeddings obtained from the structured skip-gram model on the PubMed abstracts from those obtained from the skip-gram model on the Wikipedia articles. We achieved micro F1 score of 0.50 for the subtask A, 0.39 for the subtask A and B and 0.21 for the subtask C, and our system was ranked 3rd in the end-to-end setting.

References

Isabelle Augenstein, Mrinal Kanti Das, Sebastian Riedel, Lakshmi Nair Vikraman, and Andrew McCallum. 2017. SemEval 2017 Task 10: ScienceIE - Extracting Keyphrases and Relations from Scientific Publications. In *Proceedings of the International Workshop on Semantic Evaluation*. Association for Computational Linguistics, Vancouver, Canada.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of EMNLP*. ACL, Doha, Qatar, pages 740–750.

Qi Li and Heng Ji. 2014. Incremental joint extraction of entity mentions and relations. In *Proceedings of ACL*. ACL, Baltimore, Maryland, pages 402–412.

Wang Ling, Chris Dyer, Alan W Black, and Isabel Trancoso. 2015. Two/too simple adaptations of word2vec for syntax problems. In *Proceedings of ACL*. ACL, Denver, Colorado, pages 1299–1304.

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

Makoto Miwa and Sophia Ananiadou. 2015. Adaptable, high recall, event extraction system with minimal configuration. *BMC Bioinformatics* 16(Suppl 10.):S7.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. In *Proceedings of ACL*. ACL, Berlin, Germany, pages 1105–1116.

Makoto Miwa and Yutaka Sasaki. 2014. Modeling joint entity and relation extraction with table representation. In *Proceedings of EMNLP*. ACL, Doha, Qatar, pages 1858–1869.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of CoNLL*. ACL, Boulder, Colorado, pages 147–155.