Improving Scientific Relation Classification with Task Specific Supersense

Qin Dai[1], **Naoya Inoue**[1,2], **Paul Reisert**[2], **Kentaro Inui**[1,2]
[1]Tohoku University, Japan
[2]RIKEN Center for Advanced Intelligence Project, Japan
{daiqin, naoya-i, preisert, inui}@ecei.tohoku.ac.jp

Abstract

Classifying the relationship between entities is an important natural language processing (NLP) task. Scientific Relation Classification aims at automatically categorizing scientific semantic relationships among entities in scientific documents. Conventionally, only task unspecific supersense, such as supersense (or hyernym) from WordNet (e.g., ANIMAL is the supersense of "*dog*"), is used as a feature for relation classification. In this work, we hypothesize that task specific supersense could also be utilized as an informative feature for relation classification. Specifically, we define a new entity type based on the property of a given task, and facilitate scientific relation classification with the task specific supersense. Our experiments on three different datasets prove the effectiveness of the task specific supersense on relation classification in scientific articles.

1 Introduction

In recent years, along with the number of scientific papers increasing, it is prohibitively time-consuming for researchers to review and fully-comprehend all papers. To effectively and quickly access a large amount of scientific papers and acquire useful knowledge, a wide variety of computational studies for structuralizing scientific papers has been conducted, such as Argumentative Zoning (Teufel and others, 2000), BioNLP Shared Task (2017), ScienceIE Shared Task (Augenstein et al., 2017) and Semantic Relation Extraction and Classification in Scientific Papers (Gábor et al., 2018). One fundamental study is Relation Classification (RC).

In this paper, we tackle the task of RC. RC is the task of capturing predefined semantic relations between entities from text. Thus, our task consists of the following: given a sentence that has been annotated with entity[1] mentions, we aim towards categorizing relations between entities. Suppose the following sentence:

(1) *An efficient* $\underline{\overset{entity}{\textit{bit-vector-based CKY-style parser}}}_X$ *for* $\underline{\overset{entity}{\textit{context-free parsing}}}_Y$ *is presented.*

In Example 1, one of the scientific relations we aim to identify is the relation USAGE(X, Y), which means that *bit-vector-based CKY-style parser* is used for the action of *context-free parsing*. For notational convenience, we refer to a sentence where a relation is identified as a *target sentence*, and we refer to the related entity pair as a *target entity* pair.

Many previous works on RC exist in the general domain (Kumar, 2017; Zhou et al., 2014). The earlier approaches depend on complex feature engineering such as manually prepared lexical-syntactic patterns (Boschee et al., 2005; Suchanek et al., 2006; Chan and Roth, 2010, etc.). Recently, Neural Network (NN)-based approaches achieve close or even better performance to earlier approaches without complicated manually prepared features (Zeng et al., 2014; Zhang and Wang, 2015; Santos et al., 2015). In the context of scientific RC, Ammar et al. (Ammar et al., 2017) enhanced Miwa and Bansal (Miwa

[1]In this work, *entity* refers not merely to concepts denoted by noun or noun phrase, it could be actions denoted by verb or verb phrase, and evaluation denoted by adjective or adverb etc.

and Bansal, 2016)'s end-to-end general relation extraction model by incorporating external knowledge such as gazetteer-like information extracted from Wikipedia. Pratap et al. (Pratap et al., 2018) incorporate WordNet hypernyms as the feature for scientific RC. However, no previous work leverages task specific supersense as a feature for RC.

In this work, we define the task specific supersense (TSS) as a new semantic category that is proposed according to the property of a given RC task, such as the definitions of target relations and selectional tendency of target relations. We hypothesize that TSS can be utilized to improve the performance of scientific RC.

Suppose the following target sentence taken from the SemEval-2018 task 7 dataset (Gábor et al., 2018):

(2) *This paper presents a* $\overset{entity}{\underline{\textit{critical discussion}}}_X$ *of the various approaches that have been used in the* $\overset{entity}{\underline{\textit{evaluation of Natural Language systems}}}_Y$.

In this dataset, the entity mentions are annotated but their types are not tagged. This task asks a RC system to classify the target entity pair into several predefined semantic relations. One of them is TOPIC relation. The relation TOPIC(X, Y) namely means the entity X deals with the topic Y. Therefore, the entity X tends to be a research activity, such as "*analysis*", "*survey*" and "*discussion*" etc. Based on this selectional tendency, we define a TSS to cover these words, called *RESEARCH-PROCESS*. Identifying *RESEARCH-PROCESS* for a given word such as "*discussion*" in Example 2, could help a RC system to correctly classify the target entity pair into TOPIC relation.

Similarly, suppose the following target sentences from the RANIS dataset (Tateisi et al., 2014):

(3) *A* $\overset{DATA-ITEM}{\underline{\textit{verb}}}$ *'s* $\overset{DATA-ITEM}{\underline{\textit{aspectual category}}}_Y$ *can be* $\overset{PROCESS}{\underline{\textit{predicted}}}_X$ ...

(4) ... $\overset{PLAND-OR-PROCESS}{\underline{\textit{statistical generation}}}$ *to* $\overset{PROCESS}{\underline{\textit{combine}}}_X$ $\overset{DATA-ITEM}{\underline{\textit{common phrases}}}_Y$ *into a* $\overset{DATA-ITEM}{\underline{\textit{sentence}}}$.

In this dataset, both entity mentions and entity types (e.g., *PROCESS*, *PLAN*, *DATA-ITEM*) are annotated. The target relations includes relation OUTPUT(X, Y) (as in Example 3), and INPUT(X, Y) (as in Example 4). They namely mean entity Y is the output/input of a process X. Based on the definition, we propose a TSS called *OUTPUT-PROCESS*, verbs like "*show*", "*identify*" and "*extract*" belong to this TSS, because "*a system can show/identify/extract Y*" represents that the system can output Y. If we could correctly identify the *OUTPUT-PROCESS* in a given target sentence, and apply the new specific TSS , it could help a RC system more effectively identify OUTPUT relation, in comparison with only using the original general entity type, *PROCESS*. For instances, in Example 3 and Example 4, both target entities "*predicted*" and "*combine*" belong to the same entity type, *PROCESS*, but the former specifically belongs to the TSS, *OUTPUT-PROCESS*, and the latter does not. Therefore, based on this difference, a RC system could easily distinguish them, and classify the former as OUTPUT relation.

For identifying the TSS, one possibility is to manually annotate the TSS in target sentences. However, manual annotation is time-consuming (Kim et al., 2008) and expensive (Angeli et al., 2014).

To address this issue, in this work, we propose a minimally supervised approach that utilizes supersense embeddings. Specifically, we manually prepare a small number of seed instance words for the predefined supersense (or TSS) (e.g., "*survey*" for *RESEARCH-PROCESS*) and train the embedding of word and supersense in the same vector space, like the method Flekova and Gurevych (Flekova and Gurevych, 2016) proposed, which will be detailed in Section 3. By comparing the emebdding between supersense and a given word, we determine its TSS. Our evaluation empirically demonstrates that incorporating the TSS could improve the performance of scientific RC.

2 Related Work

Conventional approaches for RC rely on human-designed, complex lexical-syntactic patterns (Boschee et al., 2005), statistical co-occurrences (Suchanek et al., 2006) and structuralized knowledge bases such as WordNet (GuoDong et al., 2005; Chan and Roth, 2010). In recent years, exploring Neural Network (NN)-based models has been the dominant approach in the field. Zeng et al. (Zeng et al., 2014) and Xu et al. (Xu et al., 2015) proposed a Convolutional Neural Network (CNN)-based framework, which depends on sentence-level features collected from an entire target sentence and lexical-level features from lexical resources such as WordNet (Fellbaum, 1998). Santos et al. (Santos et al., 2015) proposed a ranking CNN model, which is trained by a pairwise ranking loss function. To improve the ability of sequential modeling, Zhang et al. (Zhang and Wang, 2015) proposed a recurrent neural network (RNN)-based model for RC. Other variants of RNN-based models have been proposed, such as Miwa et al. (Miwa and Bansal, 2016), who proposed a bidirectional tree-structured LSTM model.

Additionally, similar NN-based approaches are used in scientific relation classification. For instance, Gu et al. (Gu et al., 2017) utilized a CNN-based model for identifying *chemical-disease* relations from the abstracts of MEDLINE papers. Hahn-Powell et al. (Hahn-Powell et al., 2016) proposed an LSTM-based RNN model for identifying *causal precedence* relationship between two event mentions in biomedical papers. Ammar et al. (Ammar et al., 2017) enhanced Miwa and Bansal (Miwa and Bansal, 2016)'s relation extraction model via extensions such as gazetteer-like information extracted from Wikipedia. Pratap et al. (Pratap et al., 2018) incorporate WordNet hypernyms as the feature for scientific RC. However, none of these approaches leverage the task specific supersense for RC.

Flekova and Gurevych (Flekova and Gurevych, 2016) integrated supersense into distributional word representation, and trained supersense embedding and word embedding in the same vector space. They used the similarity between supersense embedding and word embedding as a feature to identify supersense. We applied the similar approach to tag the TSS to enhance the performance of scientific RC.

3 Task Specific Supersense Embedding

3.1 Preparing Seed TSS Instances

To learn the TSS embedding, we firstly define a TSS according to the property of a given task, such as what kinds of relation are in the given task, what is the definition of the target relation, what type of entity tends to participate in the target relation, etc, as discussed before. We test our hypothesis on different RC tasks in the computational linguistic domain in which some RC task, like SemEval-2018 task 7 (Gábor et al., 2018), aims to classify relations, such as USAGE, TOPIC and MEDOL-FEATURE, and other task, like RC on RANIS dataset (Tateisi et al., 2014), asks for identifying relations such as INPUT and OUTPUT. Therefore, we come up with four [2] types of TSS, as shown in the first column of Table 1, for distinguishing these relations for a given specific task. For instance, tagging *SYSTEM or METHOD* in target sentences could help USAGE relation recognition. After figuring out TSS for a given RC task , we manually prepare a small number of seed instances for the predefined TSS as shown in the second column of Table 1.

TSS	Seed Instances
SYSTEM or METHOD	*parser, system, learner, decoder, technology, ...*
RESEARCH-PROCESS	*analyze, investigate, study, survey, trial, ...*
OUTPUT-PROCESS	*describe, show, learn, provide, achieve, ...*
INPUT-PROCESS	*combine, compare, convert, transform, divide, ...*

Table 1: TSS and corresponding seed instances

3.2 Building TSS Embeddings

Similar to the method proposed by Flekova and Gurevych (Flekova and Gurevych, 2016), we replace each word in a corpus by its corresponding TSS according to seed instances prepared in the previous step. In this way, besides the original corpus (see Table 2, first row), we obtain an alternative corpus where each word is replaced by its corresponding TSS (see Table 2, second row). We trained the TSS embeddings on the ACL Anthology Reference Corpus (Bird et al., 2008) and its alternative corpus jointly (e.g., both first and second row in Table 2) by the skip-gram NN architecture made available by the Gensim word2vec tool [3]. Thereby, we produce continuous representation of words and the predefined TSS in one vector space [4]. Table 3 shows the most

[2]As a preliminary study, we only select four representative types of TSS, but in the future, we will investigate more types of TSS for scientific RC.

[3]`https://radimrehurek.com/gensim`

[4]The embedding is trained with negative sampling of 25 noise words, minimal word frequency of 10, window size of 2 and alpha of 0.0025, using 15 epochs to generate 300-dimensional vectors.

1	*In the above example , three different **analyses** have been found. Ribas (1994a) reported experimental results obtained from the application of the above technique to **learn** SRs.*
2	*In the above example , three different RESEARCH-PROCESS have been found. Ribas (1994a) reported experimental results obtained from the application of the above technique to OUTPUT-PROCESS SRs.*

Table 2: Example of original corpus (1) and alternative corpus (2)

TSS	
SYSTEM or METHOD	*model, models, system, approach, algorithm method, parser, framework, classifier, module*
RESEARCH-PROCESS	*study, work, research, analysis, investigation, experiment, experiments, studies, paper, investigations*
OUTPUT-PROCESS	*obtain, derive, find, provide, describe, give, show, generate, introduce, demonstrate*
INPUT-PROCESS	*compare, combine, integrate, evaluate, convert, incorporate, augment, analyze, transform, apply*

Table 3: Top 10 most similar word embeddings for each TSS embedding

similar word to each of the predefined TSS based on their embeddings' cosine similarity.

3.3 Identifying TSS for Given Words

Since the TSS is positioned in the same vector space with original words, we could utilize the embedding cosine similarity between TSS and given words to determine their TSS. Specifically, we tag a given word with the TSS, if the cosine similarity is above a predefined threshold score [5]. For instance, given a target sentence Example 5, the TSS identification result would be Figure 1.

(5) *large vocabulary continuous speech recognition (LVCSR) , a unified framework based approach is introduced to exploit multi-level linguistic knowledge*

4 Proposed Model

4.1 Task Setting

In this paper, we create a task setting where, given definitions of target relations and collections of unannotated scientific papers, we come up with a new entity type called TSS and train TSS embedding on the raw corpus. Based on the embedding cosine similarity between TSS and a given word, we identify

[5]We set the threshold score as 0.5 for identifying TSS in SemEval2018 Task7 datasets, and set it as 0.3 for RANIS dataset.

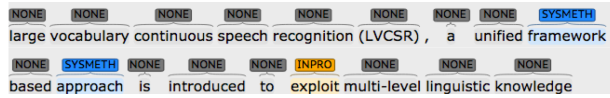

Figure 1: TSS identification example, where NONE means the word does not belong to any TSS. SYSMETH and INPRO stand for *SYSTEM or METHOD* and *INPUT-PROCESS* respectively.

the TSS, and incorporate the TSS information into a state-of-the-art RC model, thereby improve its performance on scientific RC. We execute the problem setting in computational linguistic domain, but we believe that this setting can provide useful guide to other domains, such as RC in biomedical domain.

4.2 Base Model

We choose the RC model that is proposed by Santos et al. (Santos et al., 2015) as our base RE model, since it is simple and strong. As shown in Figure 2, it is composed of three layers. The first layer is an embedding layer, which maps each word of the target sentence into a low-dimensional word vector representation. The embedding layer is calculated via Equations 1-4, where W^{w}_{emb} is a word embedding projection matrix, W^{et}_{emb} is an entity type (ET) projection matrix, x^{w}_{t} is a one-hot word representation and x^{et}_{t} is a one-hot entity type representation. The position vector e^{wp}_{t} encodes the relative distance between the current word and the head of target entity pair. For instance, in Example 6, the relative distance of the word *"for"* is [-1, 2].

(6) *We introduce referential translation machines* $(\overset{entity}{\underline{RTM}}_{A})$ *for* $\overset{entity}{\underline{quality\ estimation}}_{B}$ ...

This relative distance will be encoded into position vectors e^{wp1}_{t} and e^{wp2}_{t}, respectively, via Equation 3, where W^{wp}_{emb} is a word position embedding projection matrix and x^{wp}_{t} is a one-hot representation of the relative distance. Word embedding e^{w}_{t}, entity type embedding e^{et}_{t} and word position embedding e^{wp1}_{t} and e^{wp1}_{t} are concatenated to create the final word representation e_t. If the dataset does not have entity type information, like SemEval-2018 Task 7 dataset, e^{et}_{t} will be ignored.

$$e^{w}_{t} = W^{w}_{emb} x^{w}_{t} \tag{1}$$

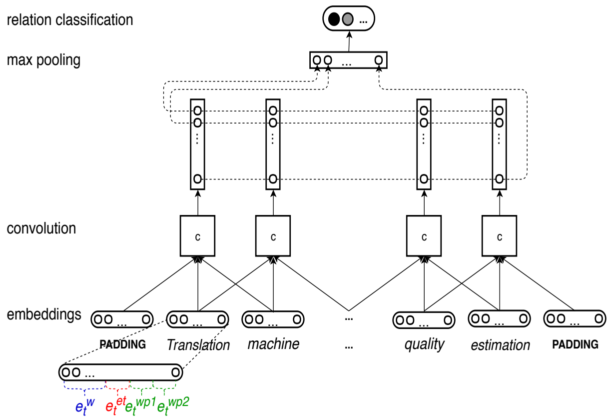

Figure 2: Base model architecture

$$e_t^{et} = W_{emb}^{et} x_t^{et} \tag{2}$$

$$e_t^{wp} = W_{emb}^{wp} x_t^{wp} \tag{3}$$

$$e_t = concat(e_t^{w}, e_t^{et}, e_t^{wp1}, e_t^{wp2}) \tag{4}$$

$$z_t = concat(e_{t-(k-1)/2}, \ldots, e_{t+(k-1)/2}) \tag{5}$$

$$h_t = tanh(W z_t + b) \tag{6}$$

The next layer is a convolutional layer, which generates a distributed convolutional window level vector h_t. h_t is calculated by Equations 5 and 6, where z_t is the concatenated embedding of k words in the convolutional window, k is convolutional window size, and W is the weight matrix of the convolutional layer. In order to address the issue of referencing words with indices outside the sentence boundaries, the target sentence is padded with a special **PADDING** token $(k-1)/2$ times at the beginning and the end.

The third layer is a max pooling layer, which chooses the maximum value from each dimension of the convolutional window level feature and merges them as the sentence level feature r via Equation 7, where i indexes feature dimensions, M is the number of feature dimensions.

$$r_i = \max_t \{(h_t)_i\}, \ \forall i = 1, \ldots, M \tag{7}$$

Finally, the model predicts the semantic relationship between a target entity pair in a target sentence x, by computing the score for a class label $c \in C$ via dot product:

$$S_\theta(x)_c = r^T [W^{class}]_c \tag{8}$$

where C is a set of predefined semantic relationships, r is the sentence level feature vector, and W^{class} is the class embedding matrix. The column of W^{class} represents the distributed vector representation of different class labels. It is worth mentioning that the model uses a logistic loss function, as shown in Equation 9:

$$\begin{aligned} L = {} & log(1 + exp(\gamma(m^+ - s_\theta(x)_{y^+})) \\ & + log(1 + exp(\gamma(m^- + s_\theta(x)_{c^-})) \end{aligned} \tag{9}$$

where $s_\theta(x)_{y^+}$ is the score of correct class label, $s_\theta(x)_{c^-}$ is the score of the most competitive incorrect class label, m^+ and m^- are margins, and γ is a scaling factor. In our experiment, we use $m^+ = 2.5, m^- = 0.5$ and $\gamma = 2$.

4.3 Incorporating TSS

We incorporate TSS information via Equations 10-11, where W_{emb}^{tss} is an TSS projection matrix, and x_t^{tss} is a one-hot TSS representation.

$$e_t^{tss} = W_{emb}^{tss} x_t^{tss} \tag{10}$$

$$e_t = concat(e_t^{w}, e_t^{et}, e_t^{tss}, e_t^{wp1}, e_t^{wp2}) \tag{11}$$

5 Data

5.1 SemEval-2018 Task 7 dataset

We evaluate the effectiveness of TSS for scientific RC on three different datasets. The first and second dataset we use in evaluation are the SemEval-2018 Task 7.1.1 & 7.1.2 datasets (Gábor et al., 2018), which are in computational linguistic domain. This task handles 6 semantic relations in scientific paper abstracts. The datasets of subtasks 1.1 and 1.2 contains titles and abstracts of papers where entity mentions are either manually annotated (Subtask 1.1), as Example 7, or automatically annotated (Subtask 1.2), as Example 8. The target semantic relations in dataset 1.1 and 1.2 are manually annotated. There are 1228/1248 training examples and 355/255 testing examples in dataset 1.1/1.2. These samples are classified into one of the following semantic relations: USAGE, RESULT, MODEL-FEATURE, PART-WHOLE, TOPIC, COMPARISON. The official evaluation metric is macro-F1 score.

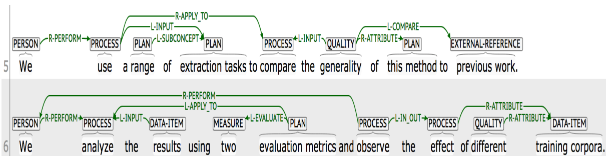

Figure 3: Annotation example shown in brat rapid annotation tool. To more clearly illustrate the direction of relation, we add directional tag "L-" and "R-" before each relation tag.

(7) Recently the LATL has undertaken the development of a <entity id="L08-1579.1">multilingual translation system</entity> based on a <entity id="L08-1579.2">symbolic parsing technology</entity> (...)

(8) The aim of this <entity id="L08-1239.17">paper</entity> is at investigating the <entity id="L08-1239.18">relationships</entity> (...)

5.2 RANIS dataset

The third dataset we use is RANIS corpus (Tateisi et al., 2014), a collection of computer science paper abstracts. The type of entity (referred to as Entity Type (ET) hereafter) and domain specific relation in the RANIS corpus has already been annotated with the annotation scheme proposed by (Tateisi et al., 2014), as Figure 3. The dataset consists of ETs such as QUALITY, PROCESS and DATA-ITEM and domain specific scientific relations, such as INPUT, OUTPUT and APPLY-TO. In total, the RANIS corpus contains 250 abstracts collected from ACL Anthology (230 abstracts in the development set and 20 abstracts in the test set) and 150 abstracts collected from ACM Digital Library. For training and testing our proposed model, we only use the 250 abstracts from ACL Anthology. From the ACL Anthology abstracts, we extract 11,520 examples from the development set of ACL Anthology and 1,142 examples from the test set of ACL Anthology. These instances are classified into one of the following semantic relations: ORIGIN, COMPARE, EQUIVALENCE, TARGET, OUTPUT, PEFORM, ATTRIBUTE, DESTINATION, RESULT, EVALUATE, APPLY-TO, INPUT, IN-OUT, SUBCONCEPT, POSS, CONDITION, SPLIT and OTHER. We choose the weighted F1 score as the evaluation metric.

Parameter Name	Value
Word Emb. size	200
Word Entity Type (or TSS) Emb. size	50
Word Position Emb. szie	100
Convolutional Units	1000
Context Window size	3
Learning Rate	0.01

Table 4: Hyperparameters for Relation Classification

6 Experiments

6.1 Setup

Since the most informative part of text to classify the relation type generally exists between and including target entity pair (Lee et al., 2017; Yin et al., 2018), we only utilize this part of the sentence and disregard the surrounding words for RC.

Previous works have shown that scientific papers specific pre-trained word embeddings can improve training for scientific RC models (Rotsztejn et al., 2018; Hettinger et al., 2018; Jin et al., 2018; Luan et al., 2018). Therefore, in this work, we trained the scientific papers specific word embeddings on the ACL Anthology Reference Corpus (Bird et al., 2008) by the skip-gram NN architecture made available by the Gensim word2vec tool. We initialized [6] the word embedding layer with the pre-trained domain-specific word embedding for RC. We randomly extract 10% training data as validation data and based on the performance on it to select all the hyperparameters. All experiments below use the hyperparameters as shown in Table 4.

6.2 Result and Discussion

In this paper, we hypothesize that TSS could be used to improve the performance of scientific RC. For testing this hypothesis, we compare the performance of TSS enhancement with the base model. In other words, we compare the performance before-and-after the automatic TSS tagging, which is mentioned in Section 3.

[6]In experiments on SemEval2018 Task 7 datasets, we didn't tune the word embedding layer, but on RANIS dataset, we tuned it while training.

Model	Precision	Recall	F-score
Base	79.61	64.73	71.40
Base + *SYSTEM or METHOD*	79.99	64.39	71.35
Base + *RESEARCH-PROCESS*	79.97	75.70	77.78
Base + *INPUT-PROCESS + OUTPUT-PROCESS*	80.05	62.81	70.39
Base + all	80.65	75.68	**78.09**

Table 5: Performance on SemEval-2018 Task 7.1.1

Model	Precision	Recall	F-score
Base	84.18	83.51	83.84
Base + *SYSTEM or METHOD*	84.92	89.04	**86.93**
Base + *RESEARCH-PROCESS*	80.09	82.19	81.12
Base + *INPUT-PROCESS + OUTPUT-PROCESS*	83.95	83.91	83.93
Base + all	82.58	88.58	85.48

Table 6: Performance on SemEval-2018 Task 7.1.2

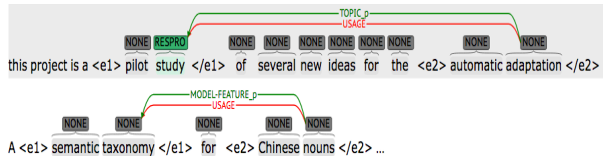

Figure 4: Comparison between **Base + all** and **Base** in SemEval-2018 Task 7.1.1, where red lines indicate the error from **Base**, while the green lines show the correctly identified relations (which end with "_p") from TSS enhanced model. <e1>, <e2>, </e1> and </e2> are entity boundary marks. RESPRO stands for *RESEARCH-PROCESS*.

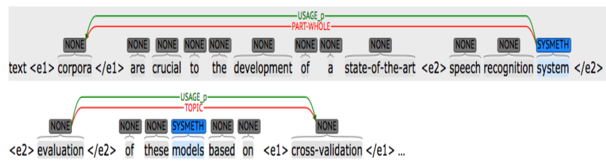

Figure 5: Comparison between **Base + *SYSTEM or METHOD*** and **Base** in SemEval-2018 Task 7.1.2.

Results for SemEval-2018 Task 7.1.1 are show in Table 5. Adding *RESEARCH-PROCESS* proves to be very beneficial compared to the base model alone, as we could improve macro-F1 by more than 5 points. This improvement can be explained by the interdependency between TSS and scientific relations as mentioned in Section 1. Thus, even if the number of training samples is small, depending on the corelation, a RC system could correctly classify some relations. While adding the TSS, *SYSTEM or METHOD*, could not enhance the performance on this subtask. This could be because given a specific RC task and its corresponding dataset, some TSS might be redundant when classifying relations. In other words, without the external information from TSS, only the internal information from the dataset itself (e.g., the hint word "*using*" in Example 9) could be enough to identify some relations (e.g., USAGE(X, Y) in Example 9).

(9) $\overset{entity}{\underline{predictor}}_X$ *pre-selects the phrase candidates* ***using*** $\overset{entity}{\underline{transition\ rules}}_Y$

Similar observation can be made for SemEval-2018 Task 7.1.2, as is indicated in Table 6. Identification of the TSS, *SYSTEM or METHOD*, could enhance the performance, while adding the *RESEARCH-PROCESS* could decrease the performance. This indicates that, given a specific RC task, different TSS could have different contribution to the overall performance. Therefore, it would be important to select proper TSS for a given RC task.

Figure 4 and Figure 5 compare some practical results between the TSS enhanced model and **Base** model in SemEval-2018 Task 7.1. Take the second line in Figure 4 as an example, although there is the preposition "*for*", which usually appears in relation USAGE (e.g., "$\underline{parsing\ algorithm}_X$ *for* $\underline{augmented\ context\text{-}free\ grammars}_Y$"), the TSS enhanced model correctly identify the relation as MODEL-FEATURE rather than USAGE, partially because there is no entity marked as *SYSTEM or METHOD*, which is usually associated with USAGE relation.

In Table 7 and Table 8, we provide our SemEval-2018 Task 7.1 performance in the context of the original task participants. In both subtasks, our model could rank among Top 3, especially in subtask 7.1.2, our system could outperform the second best system. This indicates that, firstly, our selected base model is comparatively strong, secondly, the proposed TSS could boost the performance of the strong base model, so that it could achieve the competitive result to these top ranking models. This again indicates the effectiveness of TSS on scientific RC.

Result on RANIS dataset are shown in Table 9. Adding TSS information outperforms the base model. This also proves the effectiveness of TSS on scientific RC. In addition, as mentioned in Section 5, RASNIS dataset has been manually annotated with

Rank	Participant	Macro-F1 Score
1	ETH-DS3Lab	81.7
2	UWNLP	78.9
3	SIRIUS-LTG-UiO	76.7
4	ClaiRE	74.9
5	Talla	74.2
	Our model	78.1
	Base model	71.4

Table 7: Performance comparison to Top 5 task participants (28 teams) for SemEval-2018 Task 7.1.1

Rank	Participant	Macro-F1 Score
1	ETH-DS3Lab	90.4
2	Talla	84.8
3	SIRIUS-LTG-UiO	83.2
4	MIT-MEDG	80.6
5	GU IRLAB	78.9
	Our model	86.9
	Base model	83.8

Table 8: Performance comparison to Top 5 task participants (20 teams) for SemEval-2018 Task 7.1.2

Model	Precision	Recall	F-score
Base	69.34	68.91	67.85
Base + *SYSTEM or METHOD*	70.41	69.70	68.62
Base + *RESEARCH-PROCESS*	69.52	68.83	67.91
Base + *INPUT-PROCESS* + *OUTPUT-PROCESS*	71.12	70.05	**69.34**
Base + all	70.92	69.44	68.71

Table 9: Performance on RANIS dataset

entity types such as PROCESS, PLAN and DATA-ITEM, which have been incorporated in the base model. The enhancement of performance with TSS identification indicates that TSS could be the extension of existing entity type information when classifying semantic relation. Figure 6 compares some practical results between **Base + *INPUT-PROCESS* + *OUTPUT-PROCESS*** and **Base** in RANIS dataset. It could be seen that, by adding TSS information, the RC system could correctly distinguish some relations such as INPUT and OUTPUT.

In Comparison with the improvement of performance in SemEval-2018 Task 7 dataset, the increase in RANIS dataset is smaller. This could be because, firstly, the types of target relations in RANIS dataset are more than the ones in SemEval-2018 Task 7 dataset. Secondly, in RANIS dataset, one entity tends to participate in multiple relations in a single sentence. For instance, in the annotation example shown in Figure 3, the second line, entity "*analyze*" participates in three different relation. Thus, only identifying the entity "*analyze*" as *INPUT-PRCOESS* might not be enough to distinguish them.

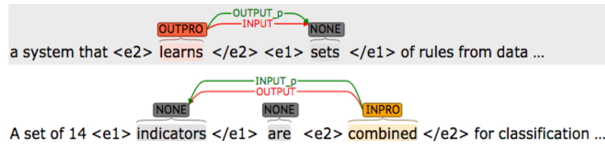

Figure 6: Comparison between **Base + *INPUT-PROCESS* + *OUTPUT-PROCESS*** and **Base** in RANIS dataset, where OUTPRO stands for *OUTPUT-PROCESS*.

7 Conclusion and Future Work

In this work, we address the task of relationship classification in scientific documents by leveraging TSS. We utilize a small number of seed TSS instances to train supersense embeddings and based on the embedding cosine similarity to identify TSS for given words. We extend one of state-of-the-art RC models by the proposed TSS information. Experimental results on three different datasets demonstrated that, firstly, TSS could be used as a feature to improve performance of scientific RC, secondly, the selection of TSS is essential for a given scientific RC task, thirdly, TSS could extend the exiting entity type information.

For the future work, since the effectiveness of TSS, we will explore more TSS which is helpful for scientific relation classification, such as the TSS that expresses NLP task (e.g., summarization, tagging and disambiguation). Due to the importance of TSS selection, we will investigate more about the criteria of TSS selection for a given RC task. In addition, we are considering an alternative way to collect TSS that captures TSS based on lexical syntactic patterns, rather than manually preparing TSS and seed words. For instance, we plan to use the lexical syntactic pattern like "X *is used for* Y" to collect arguments for slot X and Y. Then, based on their distributional information to find a representative word for X slot fillers (or Y slot fillers) as a TSS. In this way, we could avoid manually defining TSS and preparing TSS seed words, thereby increase the efficiency of TSS identification and scientific RC.

Acknowledgement

This work was supported by JST CREST Grant Number JPMJCR1513, Japan and KAKENHI Grant Number 16H06614.

References

Waleed Ammar, Matthew Peters, Chandra Bhagavatula, and Russell Power. 2017. The ai2 system at semeval-2017 task 10 (scienceie): semi-supervised end-to-end entity and relation extraction. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 592–596.

Gabor Angeli, Julie Tibshirani, Jean Wu, and Christopher D Manning. 2014. Combining distant and partial supervision for relation extraction. In *EMNLP*, pages 1556–1567.

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017. Semeval 2017 task 10: Scienceie-extracting keyphrases and relations from scientific publications. *arXiv preprint arXiv:1704.02853*.

Steven Bird, Robert Dale, Bonnie J Dorr, Bryan R Gibson, Mark Thomas Joseph, Min-Yen Kan, Dongwon Lee, Brett Powley, Dragomir R Radev, Yee Fan Tan, et al. 2008. The acl anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *LREC*.

Elizabeth Boschee, Ralph Weischedel, and Alex Zamanian. 2005. Automatic information extraction. In *Proceedings of the International Conference on Intelligence Analysis*, volume 71. Citeseer.

Yee Seng Chan and Dan Roth. 2010. Exploiting background knowledge for relation extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 152–160. Association for Computational Linguistics.

Kevin Bretonnel Cohen, Dina Demner-Fushman, Sophia Ananiadou, and Junichi Tsujii. 2017. Bionlp 2017. *BioNLP 2017*.

Christian Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Lucie Flekova and Iryna Gurevych. 2016. Supersense embeddings: A unified model for supersense interpretation, prediction, and utilization. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2029–2041.

Kata Gábor, Davide Buscaldi, Anne-Kathrin Schumann, Behrang QasemiZadeh, Haifa Zargayouna, and Thierry Charnois. 2018. Semeval-2018 task 7: Semantic relation extraction and classification in scientific papers. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 679–688.

Jinghang Gu, Fuqing Sun, Longhua Qian, and Guodong Zhou. 2017. Chemical-induced disease relation extraction via convolutional neural network. *Database*, 2017.

Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 427–434. Association for Computational Linguistics.

Gus Hahn-Powell, Dane Bell, Marco A Valenzuela-Escárcega, and Mihai Surdeanu. 2016. This before that: Causal precedence in the biomedical domain. *arXiv preprint arXiv:1606.08089*.

Lena Hettinger, Alexander Dallmann, Albin Zehe, Thomas Niebler, and Andreas Hotho. 2018. Claire at semeval-2018 task 7: Classification of relations using embeddings. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 836–841.

Di Jin, Franck Dernoncourt, Elena Sergeeva, Matthew McDermott, and Geeticka Chauhan. 2018. Mit-medg at semeval-2018 task 7: Semantic relation classification via convolution neural network. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 798–804.

Jin-Dong Kim, Tomoko Ohta, and Jun'ichi Tsujii. 2008. Corpus annotation for mining biomedical events from literature. *BMC bioinformatics*, 9(1):10.

Shantanu Kumar. 2017. A survey of deep learning methods for relation extraction. *arXiv preprint arXiv:1705.03645*.

Ji Young Lee, Franck Dernoncourt, and Peter Szolovits. 2017. Mit at semeval-2017 task 10: relation extraction with convolutional neural networks. *arXiv preprint arXiv:1704.01523*.

Yi Luan, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. The uwnlp system at semeval-2018 task 7: Neural relation extraction model with selectively incorporated concept embeddings. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 788–792.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. *arXiv preprint arXiv:1601.00770*.

Bhanu Pratap, Daniel Shank, Oladipo Ositelu, and Byron Galbraith. 2018. Talla at semeval-2018 task 7: Hybrid loss optimization for relation classification using convolutional neural networks. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 863–867.

Jonathan Rotsztejn, Nora Hollenstein, and Ce Zhang. 2018. Eth-ds3lab at semeval-2018 task 7: Effectively combining recurrent and convolutional neural networks for relation classification and extraction. *arXiv preprint arXiv:1804.02042*.

Cicero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. *arXiv preprint arXiv:1504.06580*.

Fabian M Suchanek, Georgiana Ifrim, and Gerhard Weikum. 2006. Combining linguistic and statistical analysis to extract relations from web documents. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 712–717. ACM.

Yuka Tateisi, Yo Shidahara, Yusuke Miyao, and Akiko Aizawa. 2014. Annotation of computer science papers for semantic relation extraction. In *LREC*, pages 1423–1429.

Simone Teufel et al. 2000. *Argumentative zoning: Information extraction from scientific text*. Ph.D. thesis, University of Edinburgh.

Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2015. Semantic relation classification via convolutional neural networks with simple negative sampling. *arXiv preprint arXiv:1506.07650*.

Zhongbo Yin, Zhunchen Luo, Luo Wei, Mao Bin, Tian Changhai, Ye Yuming, and Wu Shuai. 2018. Ircms at semeval-2018 task 7: Evaluating a basic cnn method and traditional pipeline method for relation classification. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 811–815.

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.

Dongxu Zhang and Dong Wang. 2015. Relation classification via recurrent neural network. *arXiv preprint arXiv:1508.01006*.

Deyu Zhou, Dayou Zhong, and Yulan He. 2014. Biomedical relation extraction: from binary to complex. *Computational and mathematical methods in medicine*, 2014.