

A Neural Edge-Editing Approach for Document-Level Relation Graph Extraction

Kohei Makino

Makoto Miwa

Yutaka Sasaki

Toyota Technological Institute

2-12-1 Hisakata, Tempaku-ku, Nagoya, 468-8511, Japan

{sd21505, makoto-miwa, yutaka.sasaki}@toyota-ti.ac.jp

Abstract

In this paper, we propose a novel edge-editing approach to extract relation information from a document. We treat the relations in a document as a relation graph among entities in this approach. The relation graph is iteratively constructed by editing edges of an initial graph, which might be a graph extracted by another system or an empty graph. The way to edit edges is to classify them in a close-first manner using the document and temporally-constructed graph information; each edge is represented with a document context information by a pretrained transformer model and a graph context information by a graph convolutional neural network model. We evaluate our approach on the task to extract material synthesis procedures from materials science texts. The experimental results show the effectiveness of our approach in editing the graphs initialized by our in-house rule-based system and empty graphs.¹

1 Introduction

Relation extraction (RE), the task to predict relations between pairs of given entities from literature, is an important task in natural language processing. While most existing work focused on sentence-level RE (Zeng et al., 2014), recent studies extended the extraction to the document level since many relations are expressed across sentences (Christopoulou et al., 2019; Nan et al., 2020).

In document-level RE, models need to deal with relations among multiple entities over a document. Several document-level RE methods construct a document-level graph, which is built on nodes of words or other linguistic units, to capture document-level interactions between entities (Christopoulou

¹The source code is available at <https://github.com/tti-coin/edge-editing>.

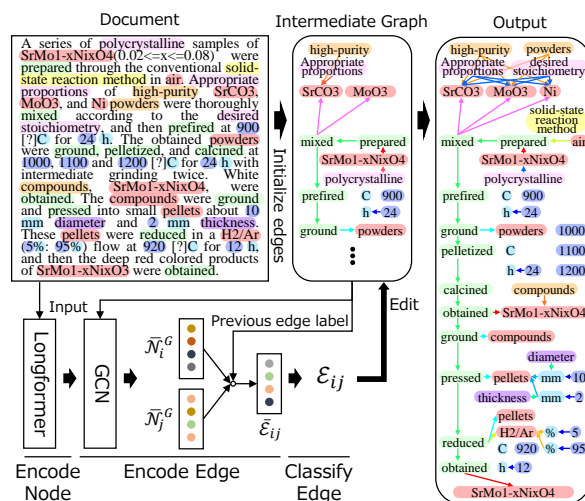


Figure 1: Overview of edge editing approach

et al., 2019; Nan et al., 2020). However, such methods do not directly consider interactions among relations in a document, while such relations are often dependent on each other, and other relations can be considered as important contexts for a relation.

We propose a novel, iterative, edge-editing approach to document-level RE. The overview of our approach and an example of the extraction results are illustrated in Figure 1. Our approach treats relations as a relation graph that is composed of entities as nodes and their relations as edges. The relation graph is first initialized using the edges predicted by an existing RE model if provided. Edges are then edited by a neural edge classifier that represents edges using the document information, pre-built graph information, and the current edge information. The document information is represented with pretrained Longformer models (Beltagy et al., 2020), while the graph information is represented with graph convolutional networks (Kipf and Welling, 2017). Edges are edited iteratively in a close-first manner so that the approach can utilize the information of edges between close entity pairs

in editing edges of distant entity pairs, which are often difficult to predict. We evaluate our approach on the task to extract synthesis procedures from text (Mysore et al., 2019) and show the effectiveness of our approach.

The contribution of this paper is three-fold. First, we propose a novel edge-editing approach for document-level RE that utilizes contexts in both relation graphs and documents. Second, we build a strong rule-based model and show that our approach can effectively utilize and enhance the output of the rule-based model. Third, we build and evaluate a neural model for extracting synthesis procedures from text for the first time.

2 Approach

Our approach extracts a relation graph on given entities from a document. We formulate the extraction task as an edge-editing task, where the approach iteratively edits edges with a neural edge classifier in a close-first manner (Miwa and Sasaki, 2014).

2.1 Iterative Edge Editing

We build a relation graph by editing the edges iteratively using the edge classifier in Section 2.2. The building finishes when all edges are edited. The edges are edited in a close-first manner (Miwa and Sasaki, 2014; Ma et al., 2019) that edits the close edges first and far edges later. The distance between the entity pair is defined based on the appearing order of entities in a document; if two entities in a pair appear m -th and $m + 3$ -th, the distance becomes 3. Note that each edge is edited only once throughout the entire editing process.

Algorithm 1 shows the method to build the graph by the iterative edge editing. To reduce the computational cost, the pairs with the same distance are edited simultaneously and the pairs with distances more than or equal to the maximum distance d_{max} are edited simultaneously. This reduces the number of edits from $|\mathcal{N}|^2$ to d_{max} .

2.2 Edge Classifier

An edge classifier predicts the class of the target edge $\hat{\mathcal{E}}_{ij}$ from inputs that are composed of a document information doc , a graph of nodes \mathcal{N} and edges \mathcal{E} , and the node pair $(\mathcal{N}_i, \mathcal{N}_j)$ of a target edge. The classifier composed of three modules:

EncodeNode that produces document-based node representations $\bar{\mathcal{N}}$ using the document doc and the entity information of the nodes \mathcal{N} .

Algorithm 1: Iterative Edge Editing

Distance(\mathcal{N}, d_1, d_2) returns pairs that have distance d ($d_1 \leq d < d_2$).

Input: doc : document, \mathcal{E} : initial edges

d_{max} : maximum distance

Output: \mathcal{E} : edited edges

$\bar{\mathcal{N}} \leftarrow \text{EncodeNode}(doc, \mathcal{N})$

while d in range($\max(|\mathcal{N}|, d_{max})$) **do**

$\bar{\mathcal{N}}^G \leftarrow \text{GCN}(\bar{\mathcal{N}}, \mathcal{E})$

$\bar{\mathcal{E}} \leftarrow \text{EncodeEdge}(\bar{\mathcal{N}}^G, \mathcal{E})$

if $d = d_{max}$ **then**

$\mathcal{P} \leftarrow \text{Distance}(\mathcal{N}, d_{max}, \infty)$

else

$\mathcal{P} \leftarrow \text{Distance}(\mathcal{N}, d, d + 1)$

end if

while (i, j) in \mathcal{P} **do**

$\mathcal{E}_{ij} \leftarrow \text{ClassifyEdge}(\bar{\mathcal{E}}_{ij})$

end while

end while

EncodeEdge that obtains the representation of edges $\bar{\mathcal{E}}$ that applies GCN on a prebuilt graph with the node representations $\bar{\mathcal{N}}$ and edges \mathcal{E} .

ClassifyEdge that predicts the class of the edge $\hat{\mathcal{E}}_{ij}$ using the edge representation $\bar{\mathcal{E}}_{ij}$ between the node pair $(\mathcal{N}_i, \mathcal{N}_j)$.

We explain the details of these modules in the remaining part of this section.

EncodeNode employs Longformer (Beltagy et al., 2020) to obtain the document-level representation. It aggregates subword representations within each entity by max-pooling Pool and concatenates the aggregated information with the entity’s class label representation \mathbf{v}^{lab} .

$$\begin{aligned} \bar{\mathcal{N}} &= \text{EncodeNode}(doc, \mathcal{N}) \\ &= [\text{Pool}(\text{Longformer}(doc)); \mathbf{v}^{lab}], \quad (1) \end{aligned}$$

where $[\cdot; \cdot]$ denotes concatenation.

To prepare the input to EncodeEdge, the obtained document-based node representation is enriched by GCN to introduce the context of each node in the prebuilt graph: $\bar{\mathcal{N}}^G = \text{GCN}(\bar{\mathcal{N}}, \mathcal{E})$. We add inverse directions to the graph and assign different weights to different classes in graph convolutional network (GCN) following Schlichtkrull et al. (2018). The produced node representation $\bar{\mathcal{N}}^G$ includes both document and prebuilt graph contexts.

EncodeEdge produces the edge representation $\bar{\mathcal{E}}$ from $\bar{\mathcal{N}}^G$. It individually calculates the repre-

sentation of the edge $\bar{\mathcal{E}}_{ij}$ for each pair of nodes $(\mathcal{N}_i, \mathcal{N}_j)$ by combining the representations of nodes similarly to Zhou et al. (2021) with the embedding of the distance of the entity pair \mathbf{b}_{ij} and the edge class e_{ij}^{old} before editing. The distance between the entity pairs is calculated in the same way as in Section 2.1. If the distance exceeds a predefined maximum distance, it will be treated as the maximum distance. We prepare fully connected (FC) layers, FC^H and FC^T , for the start point (head) and end point (tail) nodes and calculate the edge representation as follows:

$$\begin{aligned}\bar{\mathcal{E}}_{ij} &= \text{EncodeEdge}(\bar{\mathcal{N}}^G, \mathcal{E})_{ij} \\ &= [\text{FC}^H(\bar{\mathcal{N}}_i^G)^\top \mathbf{W} \text{FC}^T(\bar{\mathcal{N}}_j^G); \mathbf{b}_{ij}; e_{ij}^{old}],\end{aligned}\quad (2)$$

where \mathbf{W} denotes a trainable weight parameter.

ClassifyEdge classifies the target edge \mathcal{E}_{ij} into a relation class or no relation. It applies a dropout layer (Srivastava et al., 2014), a FC layer for output FC^{out} and softmax to the edge representation $\bar{\mathcal{E}}_{ij}$ to predict the class $\hat{\mathcal{E}}_{ij}$ with the highest probability.

$$\begin{aligned}\hat{\mathcal{E}}_{ij} &= \text{ClassifyEdge}(\bar{\mathcal{E}}_{ij}) = \arg \max \hat{\mathbf{p}}_{ij} \\ \hat{\mathbf{p}}_{ij} &= \text{Softmax}(\text{FC}^{out}(\text{Dropout}(\bar{\mathcal{E}}_{ij})))\end{aligned}\quad (3)$$

We maximize the log-likelihood in training the edge classifier.

3 Experiments

3.1 Experimental Settings

We evaluate our approach on the materials science procedural text corpus (Mysore et al., 2019). In the corpus, the synthesis procedures are annotated as a graph in a document, where 19 node types such as materials, operations, and conditions and 15 directed relation types are defined. The corpus consists of 200 documents for training, 15 for development, and 15 for test. The statistics of the corpus are shown in Appendix A. We chose this corpus since this corpus is publicly available, manually annotated, and it deals with a dense document-level relation graph.

We prepared a rule-based model (**RULE**) as a baseline and as an existing model to initialize the edges, which was adapted from the rule-based system in Kuniyoshi et al. (2020). The rules are summarized in Appendix B.

We employ the micro F-score for each relation class as the evaluation metric. We tune the hyper-parameters such as the number and dimensions of layers and dropout rate on the development set using the hyper-parameter optimization

	Dev	Test
EDIT	0.788	0.729
EDIT-IE	0.732	0.685
EDIT-GCN	0.744	0.703
RANDOM EDIT	0.751	0.690
RANDOM INIT	0.756	0.720

Table 1: Evaluation results in micro F-score without **RULE**

framework Optuna (Akiba et al., 2019) and the details are shown in Appendix C. We employ the Adam (Kingma and Ba, 2015) optimizer with the default parameters in PyTorch (Paszke et al., 2019) except for the learning rate. The training was performed without finetuning for the Longformer because the corpus is small to train a large transformer model.

We compare the following models on graphs initialized by the rule-based model (**with RULE**) and empty graphs (**without RULE**).

EDIT: Proposed model

EDIT-IE: EDIT without iterative edge editing, i.e., $d_{max} = 1$.

EDIT-GCN: EDIT without GCN by replacing $\bar{\mathcal{N}}^G$ with $\bar{\mathcal{N}}$ in Equation (2)

RANDOM EDIT: EDIT with random-order editing. Additionally, we evaluate the following model with randomly initialized graphs.

RANDOM INIT: EDIT with randomly connected edges, the number of which is same as that of the extraction results of **RULE**, with random classes

Note that although we did not provide the direct comparison with the existing models, our EDIT-GCN without **RULE** is similar to BRAN (Verga et al., 2018); the only differences are that we use Longformer (Beltagy et al., 2020) instead of transformers, and NER training is not included. Moreover, most of the models for the document-level RE require dataset annotating both entities and their mentions, so the existing models like AT-LOP (Zhou et al., 2021) cannot be directly applied to the current task.

3.2 Results without **RULE**

We show the results with empty initial graphs in Table 1. EDIT shows the highest scores and this indicates the effectiveness of our approach when the initial graphs are empty. When we compare EDIT, EDIT-IE, and RANDOM EDIT, we find that both iterative edge editing and close-first strategy

	Dev	Test
RULE	0.797	0.807
EDIT	0.878	0.851
EDIT-IE	0.863	0.863
EDIT-GCN	0.857	0.834
RANDOM EDIT	0.791	0.744

Table 2: Evaluation results in micro F-score with RULE

are effective. Since EDIT-GCN extracts from context without graph structure information, the better performance of EDIT over EDIT-GCN shows the effectiveness of the information in the graph structure. The low performance with RANDOM INIT shows that the edge information needs to be reliable.

3.3 Results with RULE

We summarize the results with RULE in Table 2. We show the detailed results for EDIT without RULE, RULE, and EDIT-IE with RULE in Appendix D.

When we compare the results with Table 1, the performance with RULE is better than the counterpart without RULE for all the settings. Furthermore, all the scores in Table 2 are better than those in Table 1, which shows the strength of RULE.

Surprisingly, the results with our approach are better than that of RULE even though RULE is better than our approach without RULE. This indicates our EDIT approach can make the prediction accurate. We can conclude that our EDIT approach can utilize the information from the rule-based model and the initialization of the edges by RULE is useful.

As for the performance of the models, most results are consistent with Table 1 except that EDIT-IE shows the highest score on the test set. This may be partly because the initial graph by RULE is already reliable and editing does not help to improve the context. Results with RANDOM EDIT support this since the performance degradation with RANDOM EDIT is large compared to Table 1 and RANDOM EDIT is harmful in this case. Moreover, the different behaviors on the development and test sets indicate an imbalance in the corpus split.

4 Case Study

We illustrated 6 graphs for an example document (Zhang et al., 2007) in the development data set shown in Figure 2: the result on the right side

of Figure 1 shows our best extraction result using EDIT-IE with RULE; Figure 3 shows the correct extraction; Figure 4 shows the extraction result using EDIT without RULE; Figure 5 shows the extraction result using RULE; and Figure 6 shows the extraction result using EDIT with RULE. Figure 3 shows the material synthesis starts from *mixed* with materials *SrCO3*, *MoO3* and *Ni* to *prefired* and so on, and the material *SrMo1-xNixO4* is synthesized. When we compare Figure 6 with Figure 5, the extraction results are similar to RULE. Although the overall performance is low, Figure 4, which does not depend on the rule, extracts relations that are not extracted by the other systems and this shows the models with RULE and without RULE capture different relations.

5 Related Work

RE has been widely studied to identify the relation between two entities in a sentence. In addition to traditional feature/kernel-based methods (Zelenko et al., 2003; Miwa and Sasaki, 2014), many neural RE methods have been proposed based on convolutional neural networks (CNNs) (Zeng et al., 2014), recurrent neural networks (RNNs) (Xu et al., 2015; Miwa and Bansal, 2016), graph convolutional networks (GCNs) (Zhang et al., 2018; Schlichtkrull et al., 2018), and transformers (Wang et al., 2019). However, sentence-level RE is not enough to cover the relations in a document, and document-level RE has increasingly received research attention in recent years.

Major approaches for document-level RE are graph-based methods and transformer-based methods. For graph-based methods, Quirk and Poon (2017) first proposed a document graph for document-level RE. Christopoulou et al. (2019) constructed a graph that included heterogeneous nodes such as entity mentions, entities, and sentences and represented edges between entities from the graph. Nan et al. (2020) proposed the automatic induction of a latent graph for relational reasoning across sentences. The document graphs in these methods are defined on nodes of linguistic units such as words and sentences, which are different from our relation graphs. Unlike our method, these methods do not directly deal with relation graphs among entities.

For transformer-based methods, Verga et al. (2018) introduced a method to encode a document with transformers to obtain entity embedding and

A series of polycrystalline samples of $SrMo_{1-x}Ni_xO_4$ ($0.02 \leq x \leq 0.08$) were prepared through the conventional solid-state reaction method in air. Appropriate proportions of high-purity $SrCO_3$, MoO_3 , and Ni powders were thoroughly mixed according to the desired stoichiometry, and then prefired at $900 [?]C$ for 24 h. The obtained powders were ground, pelletized, and calcined at 1000, 1100 and 1200 $[?]C$ for 24 h with intermediate grinding twice. White compounds, $SrMo_{1-x}Ni_xO_4$, were obtained. The compounds were ground and pressed into small pellets about 10 mm diameter and 2 mm thickness. These pellets were reduced in a H_2/Ar (5%: 95%) flow at 920 $[?]C$ for 12 h, and then the deep red colored products of $SrMo_{1-x}Ni_xO_3$ were obtained.

Figure 2: Example document

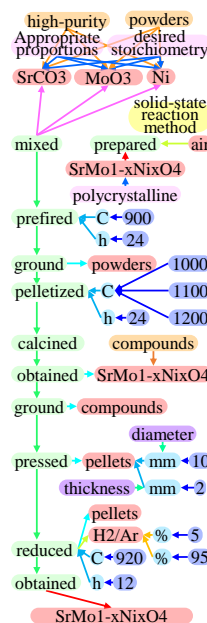


Figure 3: Gold graph for the document in Figure 2

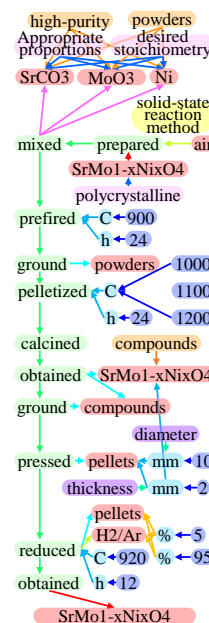


Figure 4: Example extraction results from the document in Figure 2 by EDIT without RULE

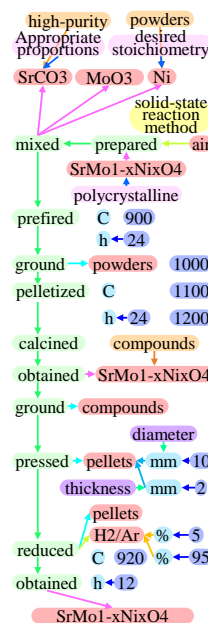


Figure 5: Example extraction results from the document in Figure 2 by EDIT with RULE

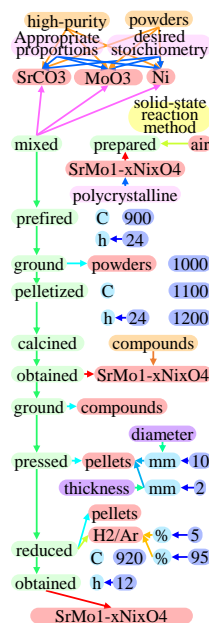


Figure 6: Example extraction results from the document in Figure 2 by our EDIT with RULE

classify the relations between entities using the embedding. Tang et al. (2020) proposed a Hierarchical Inference Network (HIN) for document-level RE, which aggregates information from entity level to document level. Zhou et al. (2021) tackled document-level RE with an Adaptive Thresholding and Localized cOntext Pooling (ATLOP) model that introduces a learnable entity-dependent threshold for classification and aggregated local mention-level contexts that are relevant to both entities.

Several studies focus on procedural texts such as cooking recipes (Bosselut et al., 2018), scientific processes (Dalvi et al., 2018) and open domain procedures (Tandon et al., 2020). They, however, do not directly treat relation graphs. Several efforts have been made to annotate procedural or action graphs in procedural text (Mori et al., 2014; Mysore et al., 2019; Kuniyoshi et al., 2020). Kuniyoshi et al. (2020) and Mehr et al. (2020) individually proposed rule-based systems to extract procedures from a document, but no neural methods have been

proposed for the extraction.

6 Conclusions

We proposed a novel edge editing approach for document-level relation extraction. This approach treats the task as the edge editing of relation graphs, given nodes. It edits edges considering contexts in the document and the relation graph. We evaluated the approach on the material synthesis procedure corpus, and the results showed the usefulness of initializing edges by the rule-based model, utilizing prebuilt graph information for editing and editing in a close-first manner. As a result, our model performed an F-score of 86.3% for edge prediction.

In future work, we plan to improve the approach to obtain more consistent and accurate relation graphs. We also would like to apply the approach to other data sets such as cooking recipes (Mori et al., 2014) and temporal graphs (Pustejovsky et al., 2003; Cassidy et al., 2014).

References

- Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Op-tuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
- Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.
- James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. [Algorithms for hyper-parameter optimization](#). In *Advances in Neural Information Processing Systems*, volume 24, pages 2546–2554. Curran Associates, Inc.
- Antoine Bosselut, Omer Levy, Ari Holtzman, Corin Ennis, Dieter Fox, and Yejin Choi. 2018. Simulating action dynamics with neural process networks. In *International Conference on Learning Representations*.
- Taylor Cassidy, Bill McDowell, Nathanael Chambers, and Steven Bethard. 2014. [An annotation framework for dense event ordering](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 501–506, Baltimore, Maryland. Association for Computational Linguistics.
- Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2019. [Connecting the dots: Document-level neural relation extraction with edge-oriented graphs](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4925–4936, Hong Kong, China. Association for Computational Linguistics.
- Bhavana Dalvi, Lifu Huang, Niket Tandon, Wen-tau Yih, and Peter Clark. 2018. [Tracking state changes in procedural text: a challenge dataset and models for process paragraph comprehension](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1595–1604, New Orleans, Louisiana. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.
- Fusataka Kuniyoshi, Kohei Makino, Jun Ozawa, and Makoto Miwa. 2020. [Annotating and extracting synthesis process of all-solid-state batteries from scientific literature](#). In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1941–1950, Marseille, France. European Language Resources Association.
- Liam Li, Kevin Jamieson, Afshin Rostamizadeh, Ekaterina Gonina, Jonathan Ben-tzur, Moritz Hardt, Benjamin Recht, and Ameet Talwalkar. 2020. [A system for massively parallel hyperparameter tuning](#). In *Proceedings of Machine Learning and Systems*, volume 2, pages 230–246.
- Shuai Ma, Gang Wang, Yansong Feng, and Jinpeng Huai. 2019. [Easy first relation extraction with information redundancy](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3851–3861, Hong Kong, China. Association for Computational Linguistics.
- S Hessam M Mehr, Matthew Craven, Artem I Leonov, Graham Keenan, and Leroy Cronin. 2020. A universal system for digitization and automatic execution of the chemical synthesis literature. *Science*, 370(6512):101–108.
- Makoto Miwa and Mohit Bansal. 2016. [End-to-end relation extraction using LSTMs on sequences and tree structures](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.
- Makoto Miwa and Yutaka Sasaki. 2014. [Modeling joint entity and relation extraction with table representation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1858–1869, Doha, Qatar. Association for Computational Linguistics.
- Shinsuke Mori, Hirokuni Maeta, Yoko Yamakata, and Tetsuro Sasada. 2014. Flow graph corpus from recipe texts. In *LREC*, pages 2370–2377.
- Sheshera Mysore, Zachary Jensen, Edward Kim, Kevin Huang, Haw-Shiuan Chang, Emma Strubell, Jeffrey Flanigan, Andrew McCallum, and Elsa Olivetti. 2019. [The materials science procedural text corpus: Annotating materials synthesis procedures with shallow semantic structures](#). In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 56–64, Florence, Italy. Association for Computational Linguistics.
- Guoshun Nan, Zhijiang Guo, Ivan Sekulic, and Wei Lu. 2020. [Reasoning with latent structure refinement for document-level relation extraction](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1546–1557, Online. Association for Computational Linguistics.

- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. [Pytorch: An imperative style, high-performance deep learning library](#). In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.
- James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, et al. 2003. The TimeBank corpus. In *Proceedings of Corpus Linguistics*, pages 647–656, Lancaster, UK.
- Chris Quirk and Hoifung Poon. 2017. [Distant supervision for relation extraction beyond the sentence boundary](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1171–1182, Valencia, Spain. Association for Computational Linguistics.
- Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The Semantic Web*, pages 593–607, Cham. Springer International Publishing.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.
- Niket Tandon, Keisuke Sakaguchi, Bhavana Dalvi, Dheeraj Rajagopal, Peter Clark, Michal Guerquin, Kyle Richardson, and Eduard Hovy. 2020. [A dataset for tracking entities in open domain procedural text](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6408–6417, Online. Association for Computational Linguistics.
- Hengzhu Tang, Yanan Cao, Zhenyu Zhang, Jiangxia Cao, Fang Fang, Shi Wang, and Pengfei Yin. 2020. HIN: Hierarchical inference network for document-level relation extraction. In *Advances in Knowledge Discovery and Data Mining*, pages 197–209, Cham. Springer International Publishing.
- Patrick Verga, Emma Strubell, and Andrew McCallum. 2018. [Simultaneously self-attending to all mentions for full-abstract biological relation extraction](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 872–884, New Orleans, Louisiana. Association for Computational Linguistics.
- Haoyu Wang, Ming Tan, Mo Yu, Shiyu Chang, Dakuo Wang, Kun Xu, Xiaoxiao Guo, and Saloni Potdar. 2019. [Extracting multiple-relations in one-pass with pre-trained transformers](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1371–1377, Florence, Italy. Association for Computational Linguistics.
- Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. [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. Association for Computational Linguistics.
- Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Journal of machine learning research*, 3(Feb):1083–1106.
- Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. [Relation classification via convolutional deep neural network](#). In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.
- S.B. Zhang, Y.P. Sun, B.C. Zhao, X.B. Zhu, and W.H. Song. 2007. Influence of Ni doping on the properties of perovskite molybdates $\text{SrMo}_{1-x}\text{Ni}_x\text{O}_3$ ($0.02 \leq x \leq 0.08$). *Solid State Communications*, 142(12):671 – 675.
- Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. [Graph convolution over pruned dependency trees improves relation extraction](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215, Brussels, Belgium. Association for Computational Linguistics.
- Wenxuan Zhou, Kevin Huang, Tengyu Ma, and Jing Huang. 2021. Document-level relation extraction with adaptive thresholding and localized context pooling. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

A Statistics of the Materials Science Procedural Text Corpus

We present the statistics of the materials science procedural text corpus² proposed by Mysore et al. (2019). Table 3 and Table 4 summarize the numbers of entities and relations, respectively.

²<https://github.com/olivettigroup/annotated-materials-syntheses>

B Rule-based Relation Extraction Model

We built a rule-based model by defining the rules to extract relations between entity pairs for the materials science procedural text corpus (Mysore et al., 2019). The rules were adapted from the rule-based model in (Kuniyoshi et al., 2020) for the target corpus. The rules depend on labels of the entities of an entity pair, distance, and the order of occurrence of the entities. According to the combination of labels of the entities, our rules are divided into three types: OPERATION–OPERATION, OPERATION–MATERIAL and other relations. In the following, the starting point of a relation is called head and the ending point is called tail, and an edge is denoted as HEAD–TAIL.

B.1 OPERATION–OPERATION

The relation OPERATION–OPERATION takes only a NEXT_OPERATION label, which means the progress of operation.

NEXT_OPERATION: Close OPERATION entities are linked with the relation from the beginning to the end in the document order, in which the entities of OPERATION appear.

B.2 OPERATION–MATERIAL

For the edges of OPERATION–MATERIAL, there are five relation labels: RECIPE_PRECURSOR indicates the input of a material; RECIPE_TARGET indicates the generation of a product; PARTICIPANT_MATERIAL indicates the generation of an intermediate product; SOLVENT_MATERIAL indicates the solvent material of an operation; and ATMOSPHERIC_MATERIAL indicates the atmosphere of an operation.

For SOLVENT_MATERIAL, ATMOSPHERIC_MATERIAL and PARTICIPANT_MATERIAL labels, a dictionary is prepared manually for each label. The relations are linked from the nearest OPERATION to a MATERIAL in the sentence if the MATERIAL match in the dictionary since these relations take specific MATERIAL entities. The dictionary is included in the source code.

RECIPE_PRECURSOR is linked from all MATERIAL that do not match the dictionary of SOLVENT_MATERIAL, ATMOSPHERIC_MATERIAL, and PARTICIPANT_MATERIAL to the nearest OPERATION. This rule-based model does not produce the relation RECIPE_TARGET. The reason for these

Entity class	Train	Dev	Test
MATERIAL	4,271	277	316
OPERATION	3,249	212	242
NUMBER	2,872	224	219
CONDITION-UNIT	1,363	101	87
MATERIAL-DESCRIPTOR	1,214	67	89
AMOUNT-UNIT	1,193	96	98
PROPERTY-MISC	481	25	16
CONDITION-MISC	468	32	20
SYNTHESIS-APPARATUS	433	20	34
NONRECIPE-MATERIAL	329	33	25
BRAND	291	30	27
APPARATUS-DESCRIPTOR	165	10	9
AMOUNT-MISC	149	14	7
META	128	12	13
PROPERTY-TYPE	124	10	4
CONDITION-TYPE	119	2	1
REFERENCE	106	10	11
PROPERTY-UNIT	92	7	8
APPARATUS-UNIT	89	6	16
CHARACTER.-APPARATUS	54	2	11
APPARATUS-PROPERTY-TYPE	26	0	6

Table 3: Entities in the materials science procedural text corpus

Relation class	train	dev	test
NEXT_OPERATION	2,898	184	202
RECIPE_PRECURSOR	876	67	89
RECIPE_TARGET	363	31	22
PARTICIPANT_MATERIAL	1,723	113	124
SOLVENT_MATERIAL	463	28	33
ATMOSPHERIC_MATERIAL	183	11	14
PROPERTY_OF	586	35	21
CONDITION_OF	1,810	132	107
NUMBER_OF	2,805	219	209
AMOUNT_OF	1,512	130	121
DESCRIPTOR_OF	1,495	91	102
BRAND_OF	423	42	41
TYPE_OF	164	7	13
APPARATUS_OF	455	20	36
APPARATUS_ATTR_OF	90	6	11
COREF_OF	267	12	14

Table 4: Relations in the materials science procedural text corpus

decisions is that it is difficult to classify these relations with simple rules.

B.3 Remaining Relations

The remaining 9 relation labels are defined between the other pairs of entity labels: PROPERTY_OF, which indicates a condition of a material; CONDITION_OF, which indicates a condition of an operation; NUMBER_OF, which indicates the relationship between a number and a unit; AMOUNT_OF, which indicates a condition of a quantity; TYPE_OF, which indicates a condition of a numerical condition; BRAND_OF, which indicates the brand of a material or equipment; APPARATUS_OF, which indicates equipment used in an

operation; APPARATUS_ATTR_OF, which indicates a numerical condition of on equipment; and DESCRIPTOR_OF, which indicates other conditions. For these labels, the rules are defined based only on the labels of head and tail entities and the distance between them. We explain the detailed rules in the remainder of this section.

PROPERTY_OF: The relation can take PROPERTY-UNIT or PROPERTY-MISC as the head and MATERIAL or NONRECIPE-MATERIAL as the tail. When PROPERTY-UNIT is a head, it is linked with the nearest MATERIAL in the sentence. When PROPERTY-MISC is a head, it is linked to the nearest MATERIAL or NONRECIPE-MATERIAL in the sentence.

CONDITION_OF: CONDITION-UNIT and CONDITION-MISC are linked to the nearest OPERATION with the relation in the sentence.

NUMBER_OF: NUMBER is linked to the nearest PROPERTY-UNIT, CONDITION-UNIT, or APPARATUS-UNIT that appear after the NUMBER in the sentence.

AMOUNT_OF: The relation is linked from AMOUNT-UNIT and AMOUNT-UNIT to the nearest MATERIAL or NONRECIPE-MATERIAL in the sentence.

DESCRIPTOR_OF: When MATERIAL-DESCRIPTOR is a head, it is linked to the nearest MATERIAL or NONRECIPE-MATERIAL in the sentence. When APPARATUS-DESCRIPTOR is a head, it is linked to the nearest SYNTHESIS-APPARATUS in the sentence.

APPARATUS_OF: The relation is linked from SYNTHESIS-APPARATUS and CHARACTERIZATION-APPARATUS to the nearest OPERATION with the priority given to the OPERATION that appear before the APPARATUS in the sentence.

TYPE_OF: PROPERTY-TYPE and APPARATUS-PROPERTY-TYPE are linked to the nearest PROPERTY-UNIT and APPARATUS-UNIT in the sentence with the relation, respectively. When CONDITION-TYPE is a head, it is linked to the nearest CONDITION-UNIT that appears before the CONDITION-TYPE in the sentence.

BRAND_OF: The relation is linked from BRAND to the nearest entities that may have brands (i.e., MATERIAL, NONRECIPE-MATERIAL, SYNTHESIS-APPARATUS, and CHARACTERIZATION-APPARATUS) in the sentence.

Parameter	Range	Value
Learning rate	[1e-5, 1e-2)	0.001
No. of GCN layers	[0, 4]	3
d_{max}	[1, 10]	4
Dimension of hidden layers	[32, 128]	85
No. of FC ^{out} layers	[1, 5]	4
No. of FC ^h and FC ^t layers	[1, 5]	1
Dropout rate	[0.0, 1.0)	0.46
Dimension of e_{ij}^{old}	[1, 32]	3
Maximum distance for b_{ij}	[1, 32]	3
Dimension of b_{ij}	[1, 100]	1
Use bidirectional GCN	True or False	True

Table 5: Search space for optimization of hyper-parameters and the selected values after optimization

Relation	Prec.	Recall	F-score
NEXT_OPERATION	0.622	0.693	0.656
RECIPE_PRECURSOR	0.632	0.539	0.582
RECIPE_TARGET	0.640	0.727	0.681
PARTICIPANT_MATERIAL	0.641	0.476	0.546
SOLVENT_MATERIAL	0.491	0.818	0.614
ATMOSPHERIC_MATERIAL	0.733	0.786	0.759
PROPERTY_OF	0.773	0.810	0.791
CONDITION_OF	0.798	0.850	0.824
NUMBER_OF	0.874	0.962	0.916
AMOUNT_OF	0.722	0.645	0.681
DESCRIPTOR_OF	0.761	0.814	0.787
BRAND_OF	0.567	0.415	0.479
TYPE_OF	0.900	0.692	0.783
APPARATUS_OF	0.657	0.639	0.648
APPARATUS_ATTR_OF	0.769	0.909	0.833
COREF_OF	0.875	0.500	0.636
Overall	0.717	0.722	0.720

Table 6: Detailed results using EDIT without RULE on the test set

APPARATUS_ATTR_OF: APPARATUS-UNIT is linked to the nearest SYNTHESIS-APPARATUS or CHARACTERIZATION-APPARATUS.

COREF_OF: The relation is not detected by the rules because it is difficult to describe rules.

C Tuning Details

We tuned our model using a hyper-parameter optimization framework Optuna (Akiba et al., 2019). We searched for the hyper-parameters that maximize micro-F scores within 600 trials on the development set. We employed the tree-structured Parzen estimator algorithm (Bergstra et al., 2011) for the sampler and the successive halving algorithm (Li et al., 2020) for the pruner with default options in Optuna. In each trial of the search, we trained our model for 100 epochs, which was confirmed by preliminary experiments to be sufficient for convergence. We searched hyper-parameters on 20 NVIDIA GPUs, which include Tesla V100, TITAN V, RTX 3090, and GTX TITAN Xp GPUs.

Relation	Prec.	Recall	F-score
NEXT_OPERATION	0.990	0.881	0.932
RECIPE_PRECURSOR	0.730	0.414	0.528
RECIPE_TARGET	0.000	0.000	0.000
PARTICIPANT_MATERIAL	0.419	0.800	0.550
SOLVENT_MATERIAL	0.697	0.418	0.522
ATMOSPHERIC_MATERIAL	1.000	0.378	0.549
PROPERTY_OF	0.905	1.000	0.950
CONDITION_OF	0.963	0.981	0.972
NUMBER_OF	0.943	0.961	0.952
AMOUNT_OF	0.744	0.865	0.800
DESCRIPTOR_OF	0.931	0.979	0.955
BRAND_OF	0.561	0.920	0.697
TYPE_OF	0.769	1.000	0.870
APPARATUS_OF	0.972	0.854	0.909
APPARATUS_ATTR_OF	0.909	0.769	0.833
COREF_OF	0.000	0.000	0.000
Overall	0.807	0.808	0.807

Table 7: Detailed results with RULE on the test set

Relation	Prec.	Recall	F-score
NEXT_OPERATION	0.905	0.990	0.946
RECIPE_PRECURSOR	0.810	0.573	0.671
RECIPE_TARGET	0.560	0.636	0.596
SOLVENT_MATERIAL	0.733	0.667	0.698
PARTICIPANT_MATERIAL	0.624	0.790	0.698
ATMOSPHERIC_MATERIAL	0.778	1.000	0.875
PROPERTY_OF	0.905	0.905	0.905
CONDITION_OF	0.953	0.944	0.948
NUMBER_OF	0.958	0.990	0.974
AMOUNT_OF	0.854	0.868	0.861
DESCRIPTOR_OF	0.941	0.931	0.936
BRAND_OF	0.880	0.537	0.667
TYPE_OF	1.000	0.692	0.818
APPARATUS_OF	0.833	0.972	0.897
APPARATUS_ATTR_OF	0.769	0.909	0.833
COREF_OF	0.750	0.429	0.545
Overall	0.856	0.870	0.863

Table 8: Detailed results using EDIT-IE with RULE on the test set

We defined the search space as shown in Table 5; the hyper-parameters for the search are composed of the learning rate for Adam, the number of GCN layers, the maximum edit distance d_{max} , the dimensions of all hidden layers, the number of FC^{out} layers, the number of FC^h and FC^t layers, the dropout rate, the dimension of e_{ij}^{old} , the maximum distance and the dimension for b_{ij} and whether to use bidirectional GCNs or uni-directional GCNs. In the table, the range column shows the range of values to search and the final value column shows the rounded selected values after the optimization.

D Detailed Evaluation Results

Our editing models for evaluation are trained with a TITAN V GPU for EDIT with RULE and a Tesla V100 GPU for the others. The training takes about 6 hours 30 minutes with EDIT-IE using RULE and

21 hours with EDIT not using RULE.

We show the detailed evaluation results with precision (Prec.), recall, and F-score on the test set in Table 6 for EDIT without RULE, Table 7 for RULE, and Table 8 for EDIT-IE without RULE. The results show the relations that are not covered by RULE, i.e., RECIPE_TARGET and COREF_OF, are extracted by our approach, and for these classes, EDIT without RULE show the better performance than the models with RULE. Some relations with high performance by RULE, including NEXT_OPERATION, CONDITION_OF, and DESCRIPTOR_OF, are extracted by EDIT-IE with RULE in high performance. This shows our approach can effectively utilize the outputs of RULE.