Electronic Supporting Information

DACRER: A Chemical Reaction Entity Recognition Method Based on Natural Language Data Augmentation Strategy

Xiaowen Zhang,†[a] Yang Li,†[b] Chaoyi Li,[a] Jingyuan Zhu,[a] Zhiqiang Gan[a],Lei Wang,*[c] Xiaofei Sun,*[c] Hendzhi You,*[ad]

[a] School of Science, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, Guangdong, China. E-mail: youhengzhi@hit.edu.cn

[b] School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230601, Anhui, China

[c] School of Information Science and Engineering, Zaozhuang University, Zaozhuang 277160, Shandong, China

[d] Green Pharmaceutical Engineering Research Center, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, Guangdong, China

† These authors contributed equally.

Supplementary Note 1: Data mining 3

1.1 Data acquire sources 3

1.2 Data annotations 3

1.3 Data augmentation 5

Supplementary Note 2: DACRER: Transformer-based extraction approach 5

Supplementary Note 3: Results and Discussion 7

3.1 Model validation on independent dataset 7

3.2 Performance analysis 8

References 10

Supplementary Note 1: Data mining

1.1 Data acquire sources

The dataset was derived from synthesis reactions catalyzed by transition metal. To acquire the data, we conducted a search using keywords such as transition metal, synthesis, and flow in the Reaxys database. We performed deduplication and discarded passages without product entities. Finally, we obtained an initial dataset comprising 200 passages for annotation. Each passages contains 2-4 sentences describing a synthesis process and encompasses chemical components such as reactants, catalysts, solvents, products, yields, temperature, time, reactions, and reagents.

1.2 Data annotations

For this work, we focused on collecting a corpus consisting of experimental sections. These sections typically provided detailed procedures for chemical reactions, which were aligned with our concept of reaction extraction. We manually selected a large number of synthesis passages from Reaxys and discarded passages without product entities. Then we obtained an initial dataset comprising 200 passages for annotation.

Firstly, tokenization was conducted using ChemDataExtractor[1] toolkit. The toolkit was specifically designed for the tokenization and segmentation of chemical texts, playing a significant role in preprocessing organic chemical passages. Following tokenization, chemical entities are split into separate tokens (e.g., a product entity 6-

(4-morpholinyl)-3-pyridinamine becomes two separate tokens, 6-(4-morpholinyl)-3- and pyridinamine). Next, we utilized the "BIO" tagging scheme for annotation. In this approach, there were "B-Type", "I-Type", and "O" tags to represent each token, where "B-Type" indicated the beginning of an entity and "I-Type" indicated the inside of it (e.g., token [6-(4-morpholinyl)-3-] was labelled as "B-Prod", token [pyridinamine] was labelled as "I-Prod"). We adopted nine major chemical labels, including Reactants, Catalyst, Solvent, Product, Yield, Temperature, Time, Reactions, and Reagents, to classify essential reaction entities. For tokens that belong to no reaction entity were labelled as "O". Finally, we obtained a corpus consisting of 200 annotated experimental passages. Each passages contained 2-4 sentences describing a synthesis process. The statistics for each entity are summarized in Table S1.

Table S1 Quantities statistics of each entity in annotated dataset.

Label	Count
O	13992
B-Reactants	427
I-Reactants	212
B-Catalyst_Reagents	154
I-Catalyst_Reagents	104
B-Workup_reagents	75
I-Workup_reagents	19
B-Reaction	6
I-Reaction	1
B-Solvent	236
I-Solvent	41
B-Yield	131
I-Yield	132
B-Temperature	141
I-Temperature	200
B-Time	180
I-Time	170

1.3 Data augmentation

This work focused on obtaining a larger-scale annotated corpus. We performed tenfold data augmentation by permutating all texts on training and validation sets. This method was developed based on a Python script without any trained model.

Supplementary Note 2: DACRER: Transformer-based extraction approach

For this work, the aim was to train a novel model in identifying chemical entities accurately. We trained a sequence tagging model to recognize chemical entities and classify them according to the provided labels, with 19 labels in total.

In this experiment, we employed the natural language processing model to convert complex chemical reaction concepts such as substances-conditions relationships into a computer-understandable high-dimensional vectors[2], thus characterizing the natural semantic information of chemical texts with word embeddings[3]. The input for sequence label task was packed token sequences. There were three types of embeddings that generated by encoder layer for each token: token embeddings, segment embeddings, and position embeddings, respectively. These embeddings were used to capture semantic information, and each token will be assigned a unique embedding different from that of others. As a result, complex chemical language was encoded into information-dense word embeddings, which were used to parse out chemical reaction semantic features and teach machine the correspondences between words and specific entity types. The word embeddings

were combined by three types of embeddings, which can be represented as follows :

$$\mathrm{x} = \mathrm{x}_{Token} \oplus \mathrm{x}_{Segment} \oplus \mathrm{x}_{Position} \quad (1)$$

After obtaining representative features, we employed DNN with two fully connected layers and activation function to obtain the probability scores of entity-label pairs[4], which can be used for sequence label. Each token was corresponded to a specific label. In addition, to achieve function optimization on the train process, the loss was adjusted using backpropagation algorithm. For each parameter θ, gradient updates iteratively can be computed by:

$$\frac{\partial L}{\partial \theta} = \sum_{i=1}^{n} \frac{\partial loss(y_i, y_v)}{\partial \theta_i} \quad (2)$$

where y_i *is the true label of the model,* y_v *indicates the prediction, and* $\frac{\partial loss(y_i, y_v)}{\partial \theta}$ *is a loss function to calculate the gradient through the following formula:*

$$L = \sum_{i=0}^{n} loss(y_i, y_v) \quad (3)$$

The model was trained according to maximum likelihood estimation. In this way, the probability of each token can be calculated. Assuming the input token sequence as x_1, x_2,…, x_n, the prediction probability as $\mathrm{P}(\mathrm{x}_i|\theta)$, each token label according to probability distribution is obtained by accumulating the argmax:

$$\mathrm{P}(X|\theta) = \mathrm{argmax}\,\theta \prod_{i=1}^{\mathrm{n}} \mathrm{P}(\mathrm{x}_i|\theta) \quad (4)$$

We added sentence representation such as “[CLS]” token and “[SEP]” token to differentiate the sentences and improve the identification[5].

Supplementary Note 3: Results and Discussion

3.1 Model validation on independent dataset

To further investigate the effectiveness of our model, we selected an extra annotated dataset, named Product Dataset, to examine the generalization performance of model. The independent dataset includes three labels to distinguish product entities from others, which were annotated in the form of token-label pairs by the same method. It was also noteworthy that Product Dataset containing new chemical texts were not presented in the original datasets. We used the same procedure to perform tenfold augmentation for the product dataset, then followed an 8:1:1 split, such that there were 2400, 300 and 30 passages in the training, validation, and test sets respectively.

The prediction results were shown in Table S2. Clearly, we can see that the performance of the model was improved on augmentation dataset. The accuracy of the model was consistent with the number of augmented chemical texts, which aligned with the observations above. When using tenfold data augmentation, the proposed model achieved the highest scores, with F1-score and accuracy about 91% and 99%. Even for the baseline dataset without any augmentation, the model achieved higher F1-score and accuracy than before, with 71% and 98% respectively. This was mainly due to a fewer number of labels in this product dataset, which was beneficial for the model to comprehend sematic associations of token-label pairs. The above results indicated that DACRER model has excellent generalization performance and can predict various of chemical labels effectively.

Table S2 Experimental results on Product Dataset. Precision(P), Recall(R), F1-score(F1), Accuracy(ACC).

Product Dataset	P(%)	R(%)	F1(%)	ACC(%)
baseline	60.7	85.0	70.8	98.5
5*fold data augmentation	80.0	80.0	80.0	98.7
10*fold augmentation	86.4	95.0	90.5	99.7

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

where TP and FP respectively indicate the true positives that the model correctly predicted and the instances that the model incorrectly predicted as positives. TN and FN respectively indicate the true negatives that the model correctly identified and the false negatives that the model failed to identify.

3.2 Performance analysis

Figure S1 visualized confusion matrix of 17 labels. Most predicted labels showed correct correspondence with truth labels. The few false positives appeared on labels with low frequency. , Model struggled to distinguish labels that only appeared few times.

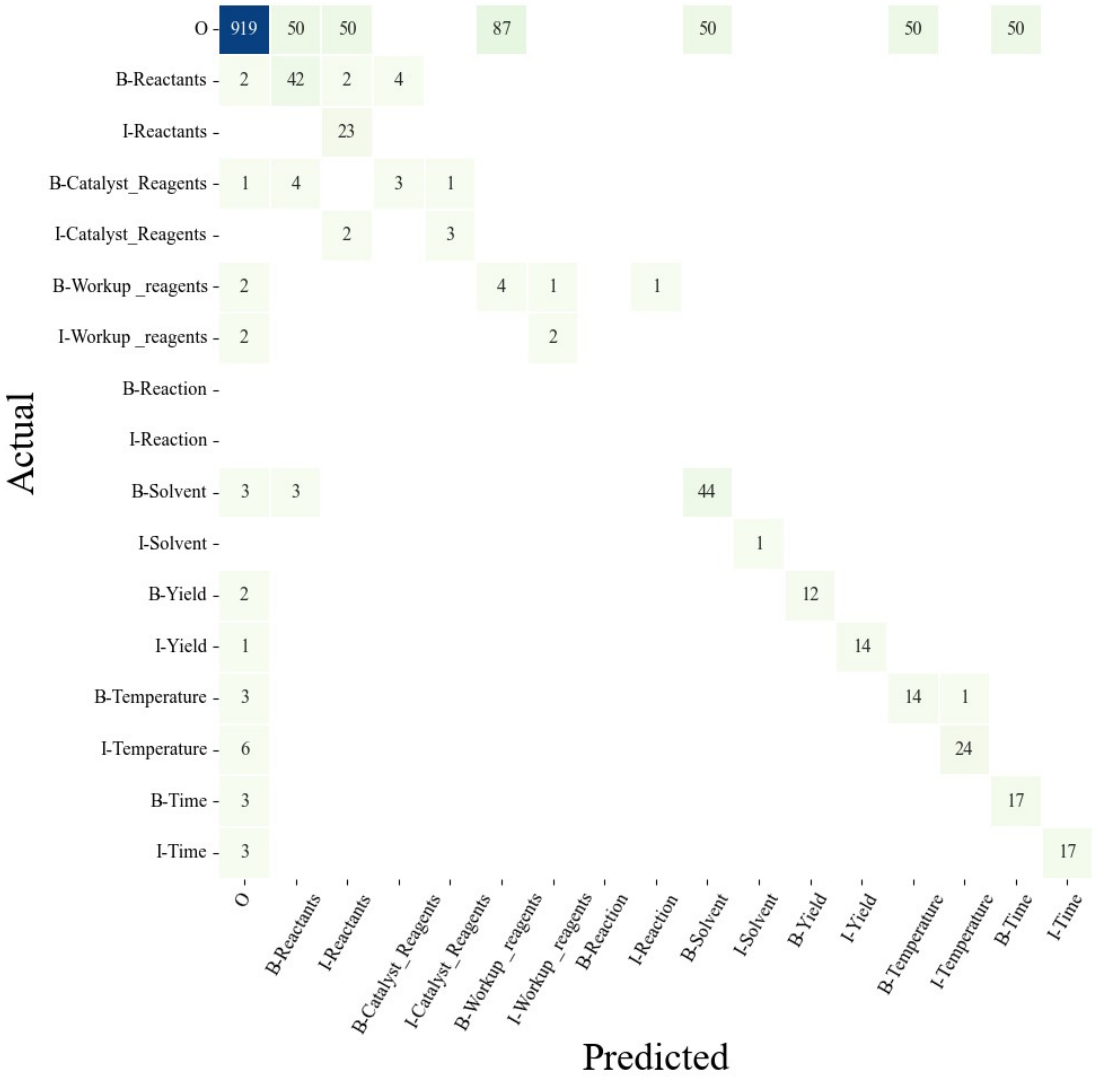

Figure S1 Confusion matrix of predicted labels and true labels. Rows represented predicted labels, while columns represented actual labels.

Figure S2 visualized two cases of entity extraction from the test dataset. The first row provided the true labels, and the second row represented the predicted label. To qualitatively understand the reaction extraction ability of our model, we illustrated correct predictions and mistake instances, respectively. Example (a) showed that the model successfully annotated reaction product as well as reaction conditions such as reactants, solvent, temperature, and time. In comparison, Example (b) showed a typical error made by model, where the Reactants label was predicted as Catalyst label mistakenly. We found that the classifier confuses Reactants and Catalyst labels several

times, this was mainly because reactants entities and catalyst entities have similar context patterns, including the same reactants and solvents. For example, as shown in Example (b), Reactant [PPh3] and Catalyst [TBBDA] shared the same contextual representation, thereby causing semantic ambiguity to the model. This kind of error was in line with the analysis above and also provided insights to improve our model.

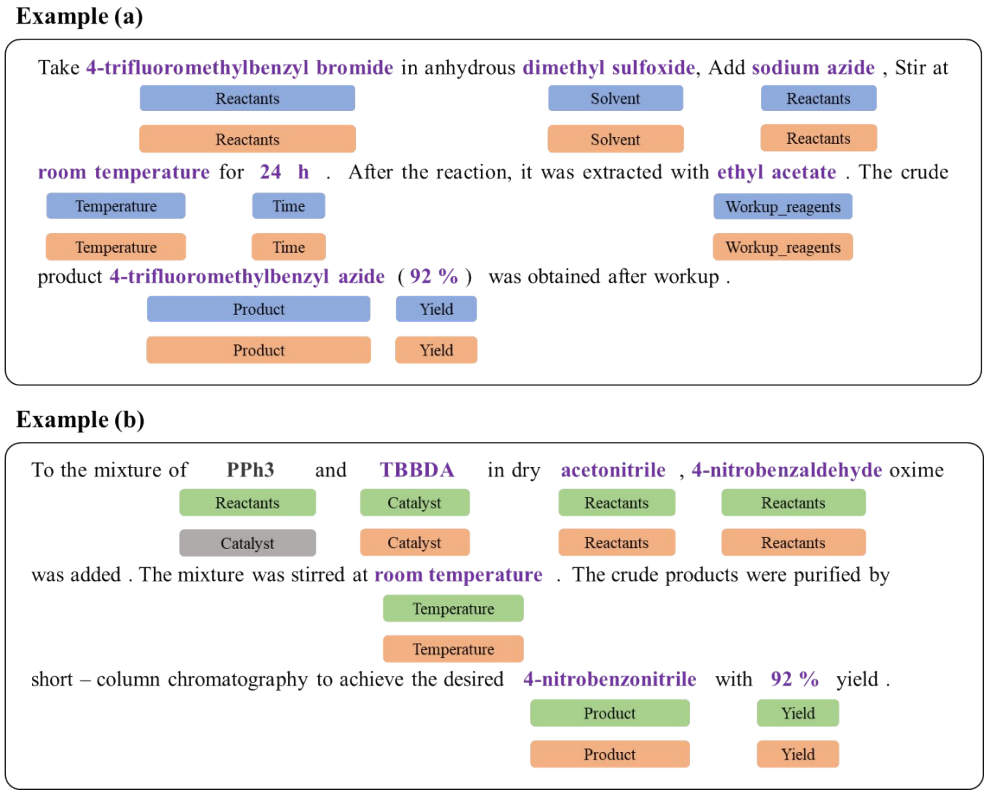

Figure S2 Examples of chemical reaction extraction. Example (a) showed a correct prediction instance, while Example (b) depicted an instance with an error.

To offer further clarification in the effect of data augmentation, we decrease the training data size to 100 passages and keep the data augmentation, the prediction results were given in Table below. The results illustrate that data augmentation is more pronounced at smaller datasets.

Table S3 Results on utilizing augmentation to 100 passages.

Data augmentation	P(%)	R(%)	F1(%)	ACC(%)
baseline	44.9	57.1	50.9	81.1
10*fold	65.9	82.8	73.0	95.3

References

1. M. C. Swain and J. M. Cole, *J. Chem. Inf. Model.*, 2016, **56**, 1894-1904.
2. K. Zheng, X.-L. Zhang, L. Wang, Z.-H. You, B.-Y. Ji, X. Liang and Z.-W. Li, *Brief. Bioinform.*, 2023, **24**, bbac498.
3. V. Tshitoyan, J. Dagdelen, L. Weston, A. Dunn, Z. Rong, O. Kononova, K. A. Persson, G. Ceder and A. Jain, *Nature*, 2019, **571**, 95-98.
4. Y. Li, X.-G. Hu, L. Wang, P.-P. Li and Z.-H. You, *Brief. Bioinform.*, 2022, **23**, bbac479.
5. J. Guo, A. S. Ibanez-Lopez, H. Gao, V. Quach, C. W. Coley, K. F. Jensen and R. Barzilay, *J. Chem. Inf. Model.*, 2021, **62**, 2035-2045.