Doctoral Dissertation

A Framework for Extraction of Structured Organic Synthesis Procedures from the Literature

Kojiro Machi

January, 2025

Graduate School of Information Science and Technology
Hokkaido University

Doctoral Dissertation

submitted to Graduate School of Information Science and Technology,

Hokkaido University

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy.

Kojiro Machi

Dissertation Committee: Professor Masaharu Yoshioka (Chief examinar)
Professor Hiroki Arimura
Professor Atsuyoshi Nakamura

A Framework for Extraction of Structured Organic Synthesis Procedures from the Literature

Kojiro Machi

Abstract

Organic synthesis procedures in the scientific literature are typically shared in prose (i.e., as unstructured data), which is not suitable for data-driven research applications. To represent such procedures, there is a well-structured language, named chemical description language (χDL). While automated conversion methods from text to χDL using either a rule-based approach or a generative large language model (GLLM) have been proposed, they sometimes produce errors. Therefore, human review following an automated conversion is essential to obtain an accurate χDL. The aim of this work is to visualize embedded information in the original text with a structured format to support the understanding of human reviewers. In this paper, we propose a novel framework for editing automatically converted χDLs from the literature with annotated text. As the annotation schema and the training data for the automated annotation system, we propose a corpus named Organic Synthesis Procedures with Argument Roles (OSPAR) that is annotated with rolesets to consider the semantic relation between the verb and argument. We also provide rolesets for chemical reactions, especially for organic synthesis, which represents the argument roles of actions in the corpus. For the automated conversion from text to χDL, we propose a rule-based conversion method. To improve the quality of automated conversions, a method of using two candidate χDLs with different characteristics was proposed: one generated by the proposed rule-based method and the other by an existing GLLM-based method. In an experiment involving six organic synthesis procedures, we confirmed that showing the outputs of both systems to the user improved recall compared with showing one output individually.

Contents

1 Introduction **1**

1.1 Background 1

1.2 Proposed approach 3

1.3 Overview of the dissertation 4

2 Related works **5**

2.1 ChEMU 5

2.1.1 Task 1: NER task 6

2.1.2 Task 2: EE task 6

2.2 Semantic roles and PropBank 7

2.3 Tools for chemical reaction information extraction 7

2.3.1 ChemicalTagger 7

2.3.2 ChemBERT 9

2.4 χDL 10

2.5 Tools for the automated extraction of χDL from the literature 12

2.5.1 SynthReader 12

2.5.2 CLAIRify 13

3 BERT-based method for ChEMU 2022 tasks **15**

3.1 Introduction 15

3.2 Methods 16

3.2.1 Mention detection 16

3.2.2 Relation classification 17

3.2.3 Experimental settings . . . 18
3.2.4 Evaluation metrics . . . 18
3.3 Results . . . 19
3.4 Discussion . . . 20
3.5 Comparison with other systems . . . 21
3.6 Summary . . . 23

4 Construction of the OSPAR corpus **25**
4.1 Introduction . . . 25
4.2 Expansion of PropBank and corpus construction . . . 27
4.2.1 Semantic roles and PropBank . . . 27
4.2.2 Task definition . . . 27
4.2.3 Rolesets for actions in chemistry . . . 28
4.2.4 Annotation schema . . . 29
4.2.5 Data curation and preprocessing . . . 30
4.2.6 Roleset expansion and corpus annotation . . . 31
4.2.7 Corpus statistics . . . 32
4.2.8 Roleset analysis . . . 33
4.3 Experiment . . . 35
4.3.1 System description . . . 36
4.3.2 Evaluation metrics . . . 37
4.3.3 Results and discussion . . . 37
4.3.4 Limitations and future works . . . 41
4.4 Summary . . . 42

5 A Framework for Extraction of χDL from the Literature **43**
5.1 Overview . . . 43
5.2 Methods . . . 43
5.2.1 The OSPAR format . . . 44
5.2.2 User interface . . . 44
5.2.3 Conversion from text to the OSPAR format. . . . 45

5.2.4 Conversion from the OSPAR format to χDL. 46
5.3 Experiment 50
5.3.1 Construction of evaluation data 50
5.3.2 Experimental settings 51
5.3.3 Results and discussion 53
5.4 Summary 56

6 Conclusions and future works **61**
6.1 Conclusions 61
6.2 Future works 61

Acknowledgements **63**

References **65**

A Article selection **77**

List of Figures

1.1 Overview of our approach . . . 3

2.1 An example of the output of ChemicalTagger . . . 8
2.2 An example of the prediction by BERT in an NER (sequential tagging) task 9
2.3 An example of the prediction by BERT in a sentiment classification (text classification) task . . . 10
2.4 Structure of χDL . . . 11

3.1 Overview of our pipeline method . . . 16
3.2 Confusion matrix for compounds in Task 1a . . . 22

4.1 Overview of OSPAR and rolesets (verbs with argument roles). . . . 27
4.2 An example of annotated text. . . . 28
4.3 Distribution of rolesets in the corpus (most frequent 15). . . . 35
4.4 Overview of the pipeline system . . . 36
4.5 Incorrect prediction due to the boundary of synthesis and work-up . . . 37
4.6 Prediction results for the unseen verbs (a) hold (b) mix . . . 39
4.7 Comparison between our method and χDL generated by SynthReader. (a) action sequence for both methods (b) gold standard and prediction by our system visualized by using brat . . . 40

5.1 Screenshot of the proposed user interface . . . 44
5.2 Screenshot of brat . . . 45
5.3 An example of conversion from the OSPAR format to χDL . . . 46

5.4 Rules for the converting ARGM into χDL action(s) 49
5.5 Dictionary to interpret words that represent parameters. 50
5.6 Examples of evaluating correct actions in exact recall and action recall . . . 52
5.7 An example for increased recall by systems other than CLAIRify when CLAIRify did not extract parameters 58
5.8 Effect of modifying the annotation 59

List of Tables

2.1 χDL actions that used in this work and their categories 12

3.1 Set of target labels for the two tasks involving mention detection and relation classification 16
3.2 Results for Task 1a on the private set 19
3.3 Detailed results for Task 1a on the private set, as predicted by the postprocessing version of the system 19
3.4 Results for Task 1b on the private set 20
3.5 Detailed results for Task 1b on the private set, as predicted by the corrected system. 20
3.6 Comparison of F-scores between development and private sets 21

4.1 Number of procedures and sentences in each data set 32
4.2 Number of NER labels in each data set 33
4.3 Number of RE labels in each data set 33
4.4 List of rolesets 34
4.5 Distribution of the types of rolesets 35
4.6 NER results on the test set 37
4.7 Relation classification results on the test set 38
4.8 Recalls of each verb (REACTION_STEP) on the test set 38

5.1 χDL actions that can be generated by our system, along with the corresponding part of the OSPAR rolesets 48
5.2 Result of explicit actions 53

5.3 Result of explicit actions for each category 54
5.4 Result of implicit actions 55

A.1 Document list of the OSPAR 78

Chapter 1

Introduction

1.1 Background

Organic synthesis procedures in the scientific literature are typically shared in prose (i.e., as unstructured data). Reproducing these procedures is sometimes challenging because these texts can be ambiguous and require interpretation by human experts. While recent machine learning techniques and laboratory automation can accelerate chemical research [1]. the absence of machine-readable procedures hinders the application of these efforts.

Because manual information extraction from the chemical literature is a labor-intensive task for domain experts, natural language processing (NLP) tools have been developed to support this work. From an early stage, rule-based methods have been developed for the extraction of information, such as compound names, reaction parameters, and actions [2, 3, 4, 5]. Because the extraction is done by rules, domain experts can see the alignment of raw text and the output if it is visualized. However, rule-based methods sometimes suffer from scalability and flexibility issues against a wide variety of texts. In the past decade, deep learning-based methods have also been developed [6, 7]. These methods are generally more flexible and robust to data variations and show higher performance than rule-based methods but require large amounts of task-specific data for the training of a model. To enable the training of deep learning models with smaller datasets, bidirectional encoder representations from transformers (BERT) [8] were proposed. BERT employed self-supervised pretraining to obtain general natural language patterns and supervised fine-tuning for solving a specific task. In the past few years, generative large language

models (GLLMs), which have shown high performance when trained only with zero or a few training examples, have constituted a trend in NLP tasks [9, 10, 11]. However, several challenges, including consistency of the output format and unclear text alignment, hinder the wide application of these models.

There are several schemes for representing organic synthesis procedures and these can be classified into two levels: (a) a general description that cannot be executable on the platforms [5, 12, 13, 14, 15, 16] and (b) a detailed description that can be executable on robotic platforms [17, 18]. At the detailed level, Mehr et al. proposed the chemical description language (χDL) [18]. χDL was designed as a universal chemical programming language that could be executed on any automated platform by translating into platform-specific low-level actions if the actions were feasible on the platforms. Their research group demonstrated the capability of χDL by executing chemical reactions on their robotic platform. Furthermore, several examples that use χDL to execute automated chemical reactions on other platforms have been reported [19, 20]. An integrated development environment for χDL named ChemIDE [21] and a rule-based natural language processing (NLP) tool named SynthReader for the conversion of organic synthesis procedures from text to χDL have also been proposed.

For the automated conversion from text to χDL, Yoshikawa et al. proposed a GLLM-based method named CLAIRify [19] and compared the performance of CLAIRify with a rule-based SynthReader. CLAIRify employed an iteration cycle of generation by a GLLM and the validation of generated code to obtain a syntactically correct output. The outputs of organic synthesis procedures generated by SynthReader and CLAIRify were compared by expert chemists, and the experts often preferred the outputs of CLAIRify over those from SynthReader. In addition, CLAIRify tends to obtain higher recall and lower precision than SynthReader.

While the focus of these studies was on improving the performance of automatic extraction, the aspect of manually correcting the automatically extracted results was not systematically investigated. However, a human review process is essential to make appropriate χDLs from the literature because these methods did not have 100% accuracy. In the review step, human reviewers need to read the original text.

1.2 Proposed approach

Along these lines, the aim of this work is to visualize embedded information in the original text with a structured format to support the understanding of human reviewers. We propose a novel framework for editing automatically converted χDLs from the literature with annotated text. Figure 1.1 shows the overview of our framework. Our framework

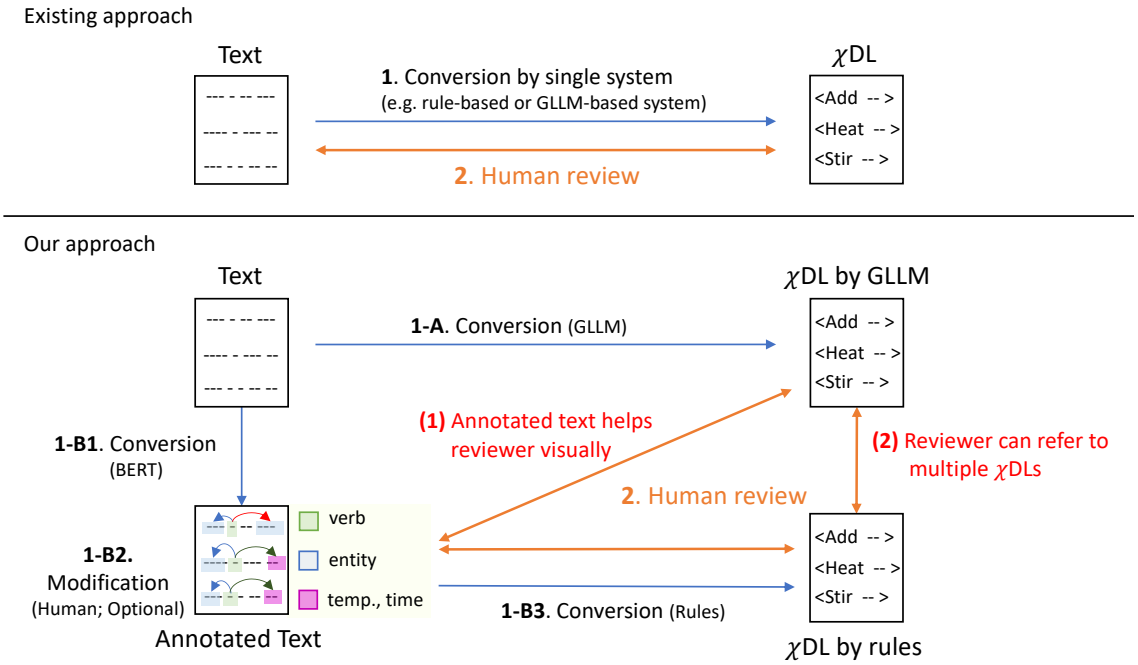


Figure 1.1: Overview of our approach

has two main points. First, to make actions described in plain text easier to understand visually, our framework provides reviewers with annotated text; it annotates action verbs with related entities and parameters. We also propose the organic synthesis procedures with argument roles (OSPAR) format [16] as the annotation format and constructed the OSPAR corpus to train a BERT-based system for the automated annotation. Second, to improve automated conversion quality, we propose a method to use two candidate χDLs with different characteristics. One is the GLLM-based method [19] and the other is a rule-based system that was developed in this study. Although CLAIRify achieved higher recall than the rule-based system, there are several cases where only the rule-based

systems can find appropriate information. Therefore, it is useful for the user to refer to both results to select appropriate parts from them. By using this framework, the user can recognize the action in the text with annotation and select appropriate action parts from the candidate χDLs. Even if the appropriate actions are not included in either of the converted χDLs, the user can easily identify the lack of action information in the χDLs by seeing the annotation.

1.3 Overview of the dissertation

Chapter 2 introduces related works on the automated extraction of chemical information from literature. Chapter 3 describes our BERT-based method developed to solve ChEMU 2022 tasks for the chemical information extraction tasks [22]. Chapter 4 describes the construction of the OSPAR corpus for the automated annotation system [16], which employs a similar method described in Chapter 3. Chapter 5 describes the framework for extraction of χDL from literature [23]. Chapter 6 summarizes the dissertation and discuss future works of this study.

Chapter 2

Related works

This chapter introduces related works on the automated extraction of chemical information from literature. The contents are the following:

- ChEMU tasks that inspired our annotation schema.
- Semantic roles for expanding ChEMU's annotation schema.
- χDL, a language for representing reproducible organic synthesis procedures.
- Methods related to automated information extraction.

2.1 ChEMU

The automated extraction of the chemical-reaction information in patents plays an important role in collecting chemical reaction information in reaction databases for use by synthetic chemists. Chemical patents contain important information about new chemical discoveries because any new chemical compounds are usually published via patents [24]. With the number of patents increasing rapidly, manually collecting the information written in patents not only takes time and cost but also requires expertise in the subject matter of the patents.

Since 2020, the ChEMU shared tasks have been held for extracting chemical reaction information from patent documents. In ChEMU 2020 [25], two tasks to identify reaction information at a level of chemical equation with core parameters and relations between the information and actions were proposed. In ChEMU 2021 [26], an anaphora resolution task

was proposed to identify the relation among chemical entities in a document. In ChEMU 2022 [27]), the above three tasks were rerun with new evaluation data. In this section, we describe two tasks from ChEMU 2020 that inspired our annotation schema.

ChEMU 2020 consists of two tasks [13]: named entity recognition (NER task[1]) and entity extraction (EE task). Here, focused on identifying and classifying entities mentioned in a text into predefined categories.

2.1.1 Task 1: NER task

Task 1 is a task to identify the boundary of entities in text and their label. The set of labels consists of the 10 labels. EXAMPLE_LABEL represents a label to specify a reaction. STARTING_MATERIAL, REAGENT_CATALYST, REACTION_PRODUCT, SOLVENT and OTHER_COMPOUND represent chemical substances with their role in a reaction. TIME and TEMPERATURE represent the parameters related to the reaction. YIELD_PERCENT and YIELD_OTHER represent information related to the yield of the product.

2.1.2 Task 2: EE task

Task 2 is a task that involves both mention detection and relation extraction. The mention detection task identifies event trigger words (e.g. "added", "heated") that have a relationship with entities in NER tasks and identifies relations between the event trigger words and the entities. Almost all event trigger words involve a label that is either REACTION_STEP or WORK_UP[2] and some event trigger words involve both labels. The relation extraction task identifies relations between the event trigger words and compounds, which are annotated with ARG1, and relations between the event trigger words and parameters or yields are annotated with ARGM. While they referred PropBank [28] to define relation labels, they did not consider detailed semantic relation that is described in

[1]The abbreviation "NER" is also used as the general meaning of named entity recognition, which is a task to identify and classify entities in text into predefined labels. Therefore, we use "NER task" or "ChEMU's NER task" to refer ChEMU's NER task in this dissertation.

[2]To perform chemical reaction, two steps are usually required: synthesis (represented by REACTION_STEP) to convert materials to a product, and work-up (represented by WORK_UP) to isolate and purify the product.

Section 2.2. This is because they are interested in the reaction information at the level of the reaction database. For example, their corpus did not distinguish the order of addition in the following phrase: "*A is added to B*" because they just needed a chemical reaction formula derived as A + B.

2.2 Semantic roles and PropBank

Semantic roles express the relation between a predicate (verb) and its arguments. PropBank [28] is one of the data sets annotated with semantic roles by using *The Wall Street Journal*. PropBank defines the semantic roles of verb senses individually and assigns a set of roles with numbers (Arg0, Arg1, Arg2, ...) to each verb sense (this is called roleset). ARG0 generally represents the agent of the predicate, and Arg1 represents the patient. Other arguments more than Arg2 do not have consistent usage.

This is an example of roleset.

- add.02
  - ARG0: adder
  - ARG1: thing being added
  - ARG2: thing being added to

If we apply this roleset to the sentence "*A is added to B*", A is ARG1 and B is ARG2. Because one verb can have multiple rolesets, the number after the verb shows the index number of the roleset.

2.3 Tools for chemical reaction information extraction

2.3.1 ChemicalTagger

ChemicalTagger is a rule-based text-mining tool for the extraction of chemical reaction information from the chemical literature [5]. ChemicalTagger extracts information such as compounds, their quantities, roles in a reaction, and units, as well as action phrases that represent the steps carried out during a reaction. Figure 2.1 shows an example of the output of ChemicalTagger. The compound information "acetonitrile (100 mL)"

Input text: *To the flask is added acetonitrile (100 mL).*

Tagged text

```
<?xml version="1.0" encoding="UTF-8"?>
<Document>
  <Sentence>
    <ActionPhrase type="Add">
      <PrepPhrase>
        <TO>To</TO>
        <NounPhrase>
          <DT-THE>the</DT-THE>
          <APPARATUS>
            <NN-APPARATUS>flask</NN-APPARATUS>
          </APPARATUS>
        </NounPhrase>
      </PrepPhrase>
      <VerbPhrase>
        <VBZ>is</VBZ>
        <VB-ADD>added</VB-ADD>
      </VerbPhrase>
      <NounPhrase>
        <MOLECULE>
          <OSCARCM>
            <OSCAR-CM>acetonitrile</OSCAR-CM>
          </OSCARCM>
          <QUANTITY>
            <_-LRB->(</_-LRB->
            <VOLUME>
              <CD>100</CD>
              <NN-VOL>mL</NN-VOL>
            </VOLUME>
            <_-RRB->)</_-RRB->
          </QUANTITY>
        </MOLECULE>
      </NounPhrase>
    </ActionPhrase>
    <STOP>.</STOP>
  </Sentence>
</Document>
```

Figure 2.1: An example of the output of ChemicalTagger

in the input text can be extracted by the <MOLECULE> tag. To see the child elements of <MOLECULE>, the compound name can be extracted by <OSCARCM>. Additionally, the quantity of the compound can be extracted by <QUANTITY> and the type of quantity is also extracted by <VOLUME>. ChemicalTagger can recognize other quantities by using tags such as <MASS> <AMOUNT> if they are written in input text.

2.3.2 ChemBERT

ChemBERT [29] is a BERT model for information tasks on chemistry. BERT employed self-supervised pretraining to obtain general natural language patterns and supervised fine-tuning for solving a specific task such as text classification tasks.

As an example of the fine-tuning task, sequential tagging task such as NER can be learned. Figure 2.2 shows an example of the prediction by BERT in an NER task. The

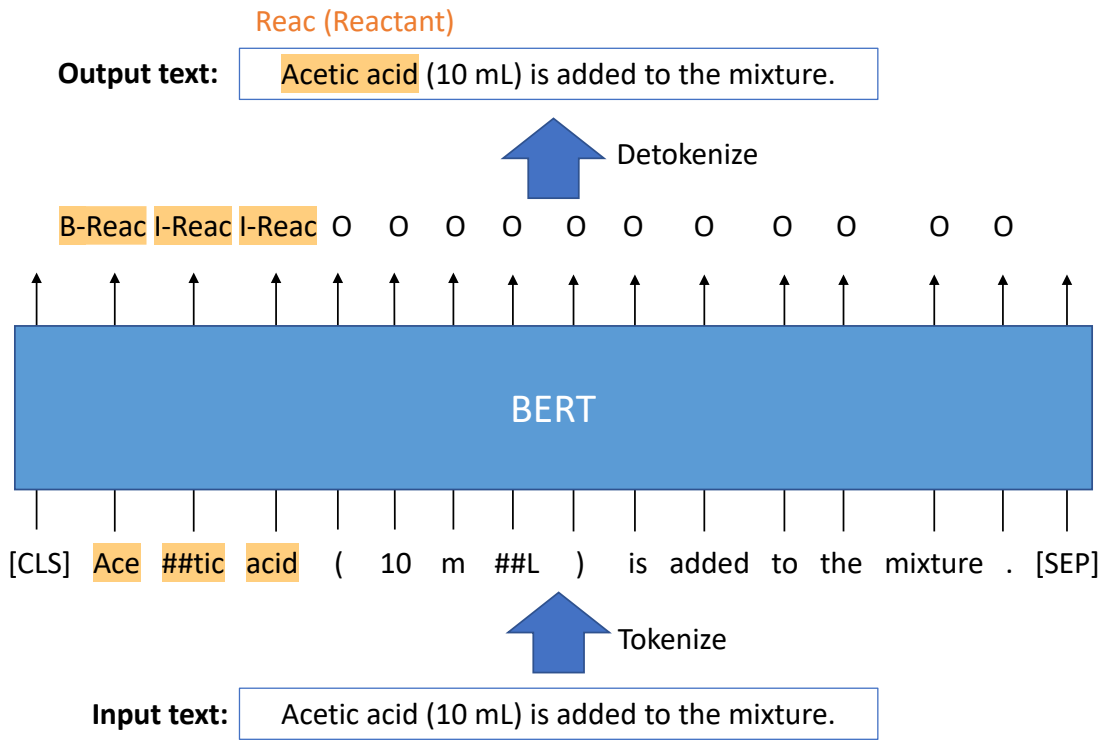


Figure 2.2: An example of the prediction by BERT in an NER (sequential tagging) task *Tokens that start with ## indicate separated tokens from a single word.

input text is tokenized into subwords [30] and special tokens such as [CLS] for classifica-

tion) are inserted. Then, BERT outputs the labels of subword tokens that indicate the boundary of entities by IOB2 labels. Finally, the annotated text is obtained by detokenize the subwords with their label.

Another example is text classification such as sentiment classification. Figure 2.3 shows an example of the prediction by BERT in a sentiment classification task. The input text

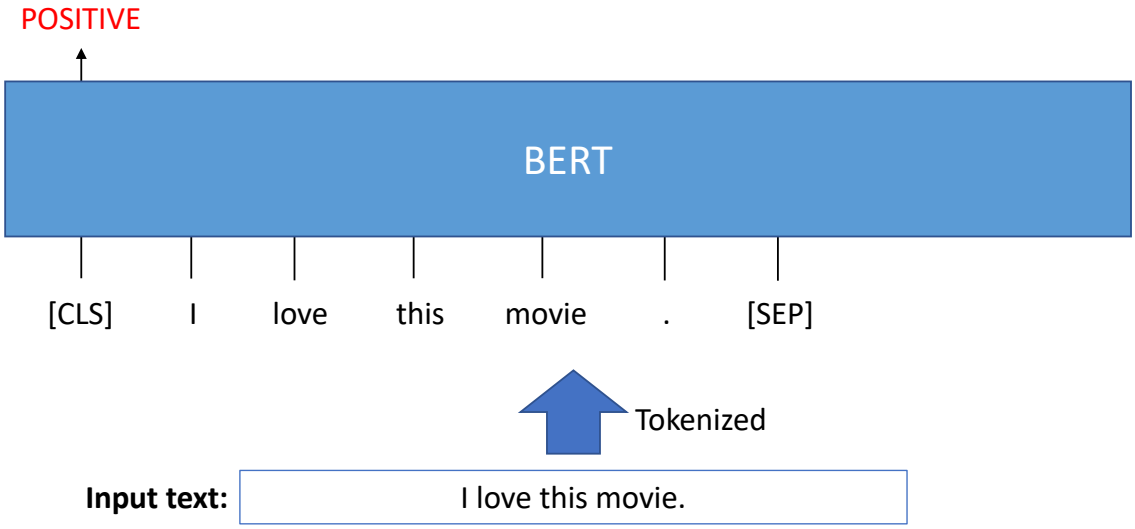


Figure 2.3: An example of the prediction by BERT in a sentiment classification (text classification) task

is tokenized as well as the example of NER. Then, the label of the text is predicted by using the output embedding of the `[CLS]` token.

To obtain higher performance in domain-specific text, several domain-specific BERTs have been introduced [31, 32, 33, 29, 34].

2.4 χDL

χDL is a universal programming language for representing chemical reaction procedures proposed by Mehr et al. [18]. Table 2.4 shows the structure of χDL. χDL uses the same syntax as extensible markup language (XML) and consists of three mandatory sections (`Hardware`, `Reagents` and `Procedure`) and two optional sections (`Metadata` and `Parameters`) which are enclosed in `Synthesis` section.

```
<?xdl version="2.0.0" ?>
<XDL>
  <Synthesis>

    <Hardware>
        <Component
            id="reactor"
            type="reactor" />
        ...
    </Hardware>

    <Reagents>
        <Reagent
            name="acetonitrile" />
        ...
    </Reagents>

    <Procedure>
        <Add
            vessel="reactor"
            reagent="acetonitrile"
            volume="100 ml" />
        ...
    </Procedure>

  </Synthesis>
</XDL>
```

Figure 2.4: Structure of χDL

Hardware The `Hardware` section consists of the `Component` tags that represent the equipment and devices used in the `Procedure` section. The `id` and `type` attributes are mandatory for `Component` tags. The `type` attributes define the type of the `Component` such as flask and reactor.

Reagents The `Reagents` section consists of the `Reagent` tags that represent the reagents used in the `Procedure` section. The `name` attribute is mandatory for `Reagent` tags. Attributes for representing the substance's information such as international chemical identifier (InChI) and simplified molecular input line entry system (SMILES) are also acceptable. To represent the role of the `Reagent` in a reaction, `role` attribute (optional) can be used.

Procedure The `Procedure` section consists of the tags that represent actions for performing a reaction. Table 5.1 shows the χDL actions that used in this work and their categories. Here, the actions enclosed in parentheses did not appear in the evaluation data in the experiment in Section 5.3. The `vessel` attribute is mandatory in the actions

Table 2.1: χDL actions that used in this work and their categories

Liquid handling	Stirring	Temperature control	Inert gas	Special
Add	StartStir	HeatChill	EvacuateAndRefill	Wait
Separate	Stir	HeatChillToTemp	Purge	(Repeat)
Transfer	StopStir	StartHeatChill	(StartPurge)	
		StopHeatChill	(StopPurge)	

except for the `Wait` action. The other attributes depend on each action. For example, `time` and `temperature`.

2.5 Tools for the automated extraction of χDL from the literature

2.5.1 SynthReader

In 2020, Mehr et al. proposed a rule-based method named SynthReader for the automated extraction of χDL from the literature [18]. While deep learning-based methods such as Transformer [35] and BERT had already exist as powerful information extraction models,

there was no labeled dataset for the conversion of χDL from text. Therefore, they employed a rule-based method for SynthReader. However, it is difficult to maintain a set of rules used in SynthReader against a wide variety of sentences.

2.5.2 CLAIRify

In 2023, Yoshikawa et al. proposed a GLLM-based method named CLAIRify [19] and compared the performance of CLAIRify with a rule-based SynthReader. They used generative pretrained transformer (GPT) [9] as a GLLM (GPT3.5). CLAIRify employed an iteration cycle of generation by GPT and the validation of generated code to obtain a syntactically correct output. The outputs of organic synthesis procedures generated by SynthReader and CLAIRify were compared by expert chemists, and the experts often preferred the outputs of CLAIRify over those from SynthReader. In addition, CLAIRify tends to obtain higher recall and lower precision than SynthReader. The experts mentioned that the effects of missing actions are more severe than ambiguous or wrong actions when they determine preferred outputs.

Chapter 3

BERT-based method for ChEMU 2022 tasks

This Chapter 3 describes our BERT-based method developed to solve ChEMU 2022 tasks for the chemical information extraction tasks [22].

3.1 Introduction

In recent years, deep learning has been recognized as a promising approach to information extraction from chemical literature. For example, pretrained language models such as BioBERT [32] and ChemBERT [29] have shown high performance in information extraction tasks. Moreover, the best systems in ChEMU 2020 and 2021 tasks all employed deep learning-based approaches, together with a small amount of rule-based postprocessing [25, 26]. In addition, both of these systems used a pipeline approach for relation extraction tasks [36, 37], such as the event extraction and anaphora resolution tasks.

This chapter describes our system and results for the two tasks at ChEMU 2022: Task 1a[1] and Task 1b[2]. We employed hybrid approaches that used deep learning models and a small set of postprocessing rules.

[1] Task 1 in ChEMU 2020 (Section 2.1.1)
[2] Task 2 in ChEMU 2020 (Section 2.1.2)

3.2 Methods

We developed systems for mention detection and relation classification tasks that used ChemBERT, a pretrained language model for chemistry-related documents. This approach was similar to those of the best systems adopted in previous tasks [36, 37]. Figure 3.1 shows our pipeline method. First, we split a snippet into sentences by using ChemDataExtractor [12]. Then, ChemBERT predicted the labels for mentions/relations. postprocessing methods were adopted for mention detection in Task 1a and relation detection in Task 1c, with the aim of addressing the document-level context.

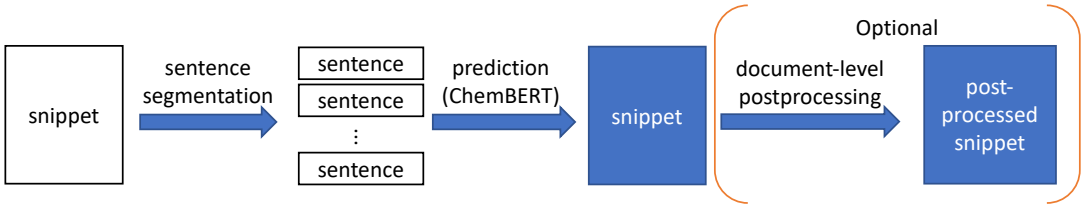


Figure 3.1: Overview of our pipeline method

We fine-tuned ChemBERT for Task 1a. In addition, we fine-tuned multiple ChemBERTs for Task 1b because the task was more complex than Task 1a. Table 3.1 shows the set of target labels for the two tasks involving mention detection and relation classification.

Table 3.1: Set of target labels for the two tasks involving mention detection and relation classification

	Mention detection	Relation classification
Task 1a	Named entity	-
Task 1b	Named entity (Task 1a)	ARG1
	Event trigger word	ARGM

3.2.1 Mention detection

For Task 1a, we trained a ChemBERT model and constructed a set of postprocessing rules. First, ChemBERT was fine-tuned on the training set, excluding snippets that had overlapping mentions. A snippet was split into sentences by using Chem-

DataExtractor [12] and the sentences were split by a simple regex rule that used a particular tool [38] by default. Then, IOB2 labels were assigned to the tokens. We used a linear classifier to predict the label of each tokens and the input for it was the output of the first sub-token [8]. Second, two postprocessing methods were applied. Because compounds in a heading section, which contains the product of the snippet (REACTION_PRODUCT) and/or the final product of the multistep reaction (OTHER_COMPOUND), cannot be distinguished without context, a postprocessing method is required. The rule adopted is that if EXAMPLE_LABEL exists in a snippet, then the REACTION_PRODUCT that appears before the last EXAMPLE_LABEL is annotated with OTHER_COMPOUND. Then, if OTHER_COMPOUND appears in a heading and the same string appears as a REACTION_PRODUCT after the last source compound (STARTING_MATERIAL, REAGENT_CATALYST, or SOLVENT), then the entity is labeled as a REACTION_PRODUCT.

For Task 1b, we used the method in Task 1a for named entity recognition and trained a ChemBERT model for the detection of event trigger words in the same manner as for Task 1a.

3.2.2 Relation classification

For Task 1b, we trained a one ChemBERT model for ARG1 relations and a second for ARGM relations. The input to the relation classification was a sentence with the candidate pair for a relation between an event trigger word enclosed by [E1] and [/E1] tokens and a target enclosed by [E2] and [/E2] tokens. The output was a binary classification result indicating whether the pair has a relation or not. All candidate pairs in a sentence were classified and positive relations were annotated by the system. For example, if two event trigger words and three targets are included in a sentence, the number of candidate pairs would be six. This approach is similar to the Melax Tech's system [36], which performed best in the ChEMU 2020 task [25]. This approach was also discussed in a general framework [39]. In the training stage, we used not only gold-standard entities but also predicted event trigger words that were generated by five systems trained on 80% of the training set, similarly to five-fold cross-validation.

In addition to the above methods, we used a postprocessing tool distributed by the task organizer[3], which generates a coreference between A and C when coreferences between A and B and between B and C already exist.

3.2.3 Experimental settings

We used ChemBERT v3.0 [29] for the mention detection and relation classification models. ChemBERT was implemented by using AllenNLP [40] and HuggingFace Transformers [41]. We used the AdamW optimizer [42] and cross entropy loss for optimization. The models were trained on the training set for the task and evaluated on the development set and private test sets. Hyperparameter values were set as follows: max sequence length=384 (covering all sequences contained in the training and development sets), batch size=16, learning rate=1e-5, and patience=7. Because the relation classification in Task 1c accepts a pair of sentences as its input, a maximum sequence length of 512 was used for this task. The validation metric for early stopping of mention detection in the development set was the F-score. For relation classification, it was the validation loss.

3.2.4 Evaluation metrics

For the evaluation of the system, Precision, Recall, and F-score were used.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F\text{-}score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

where TP is the number of true positives, which are correctly predicted by the system, FP is the number of false positives, which are incorrectly predicted by the system, and FN is the number of false negatives, which are not found by the system but exist in the corpus.

To evaluate close failures, relaxed match for the evaluation of NER in addition to an exact match were used. The relaxed match means that the span of the entity is partially matched TP and the label is correct.

[3] https://raw.githubusercontent.com/yuan-li/chemu2021/master/apply-transitive-closure.py

3.3 Results

We submitted a system with postprocessing for Task 1a before the deadline of the ChEMU 2022 competition and a system without postprocessing after the deadline. Table 3.2 shows our results for Task 1a on the private set. Our system obtained an exact match F-score of

Table 3.2: Results for Task 1a on the private set

Systems	Exact			Relaxed		
	P	R	F	P	R	F
ChemBERT	0.9327	0.9349	0.9338	0.9481	0.9503	0.9492
ChemBERT + PP	0.9401	0.9422	0.9412	0.9561	0.9583	0.9572

*P represents precision, R represents recall, and F represents F-score. The scores

0.9412 and a relaxed match F-score of 0.9572. Table 3.3 shows our results for the private set in detail.

Table 3.3: Detailed results for Task 1a on the private set, as predicted by the postprocessing version of the system

Entity	Exact			Relaxed		
	P	R	F	P	R	F
EXAMPLE_LABEL	0.9714	0.9913	0.9812	0.9714	0.9913	0.9812
OTHER_COMPOUND	0.9498	0.9486	0.9492	0.9637	0.9625	0.9631
REACTION_PRODUCT	0.9112	0.9034	0.9073	0.9433	0.9352	0.9392
REAGENT_CATALYST	0.8529	0.9050	0.8782	0.8721	0.9253	0.8979
SOLVENT	0.9284	0.9666	0.9471	0.9284	0.9666	0.9471
STARTING_MATERIAL	0.8997	0.8594	0.8791	0.9353	0.8934	0.9138
TEMPERATURE	0.9802	0.9770	0.9786	0.9901	0.9869	0.9885
TIME	0.9673	0.9741	0.9707	0.9883	0.9953	0.9918
YIELD_OTHER	0.9853	0.9711	0.9782	0.9902	0.9759	0.9830
YIELD_PERCENT	0.9750	0.9943	0.9846	0.9778	0.9972	0.9874
All	0.9401	0.9422	0.9412	0.9561	0.9583	0.9572

We submitted one system for Task 1b before the deadline of the competition. We also submitted a corrected version after the deadline, having found an error related to the sequence length of the input to the system. Table 3.4 shows our results for Task 1b on the private set. The corrected system obtained an exact match F-score of 0.8865 and a relaxed match F-score of 0.9027.

Table 3.4: Results for Task 1b on the private set

Relation	Exact			Relaxed		
	P	R	F	P	R	F
ChemBERT	0.9058	0.8685	0.8868	0.9222	0.8842	0.9028
ChemBERT (Corrected)	0.9054	0.8684	0.8865	0.9220	0.8842	0.9027

Table 3.5 shows our results for the private set in detail.

Table 3.5: Detailed results for Task 1b on the private set, as predicted by the corrected system.

Relation	Exact			Relaxed		
	P	R	F	P	R	F
ARG1|REACTION_STEP|OTHER_COMPOUND	0.4756	0.5342	0.5032	0.5000	0.5616	0.5290
ARG1|REACTION_STEP|REACTION_PRODUCT	0.8731	0.8561	0.8645	0.9179	0.9000	0.9089
ARG1|REACTION_STEP|REAGENT_CATALYST	0.8399	0.8744	0.8568	0.8590	0.8904	0.8744
ARG1|REACTION_STEP|SOLVENT	0.8929	0.8883	0.8906	0.8596	0.8950	0.8770
ARG1|REACTION_STEP|STARTING_MATERIAL	0.8832	0.8043	0.8419	0.9036	0.8228	0.8613
ARG1|WORKUP|OTHER_COMPOUND	0.9455	0.9052	0.9249	0.9627	0.9217	0.9418
*ARG1|WORKUP|REACTION_PRODUCT	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
*ARG1|WORKUP|REAGENT_CATALYST	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
*ARG1|WORKUP|SOLVENT	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
*ARG1|WORKUP|STARTING_MATERIAL	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
ARGM|REACTION_STEP|TEMPERATURE	0.9262	0.8750	0.8999	0.9328	0.8811	0.9062
ARGM|REACTION_STEP|TIME	0.8976	0.9024	0.9000	0.9213	0.9261	0.9237
ARGM|REACTION_STEP|YIELD_OTHER	0.9845	0.9315	0.9573	0.9871	0.9340	0.9598
ARGM|REACTION_STEP|YIELD_PERCENT	0.9671	0.9229	0.9444	0.9701	0.9257	0.9474
ARGM|WORKUP|TEMPERATURE	0.9063	0.6541	0.7598	0.9271	0.6692	0.7773
ARGM|WORKUP|TIME	0.7895	0.4054	0.5357	0.7895	0.4054	0.5357
*ARGM|WORKUP|YIELD_OTHER	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
*ARGM|WORKUP|YIELD_PERCENT	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
All	0.9054	0.8684	0.8865	0.9220	0.8842	0.9027

* represents relations that had fewer than five gold-standard examples on the private set.

3.4 Discussion

Selecting "the best model" of the various models in training was difficult. Table 3.6 shows the F-scores on the development and private sets. We used models that showed the best F-score on the development set for mention detection and the best validation loss for relation classification; therefore, it is not surprising that F-scores on the development sets were better than those for the private sets. Therefore, we must reconsider training methods when seeking a better model.

In Task 1b, all relations whose recalls were zero had fewer than five gold-standard

Table 3.6: Comparison of F-scores between development and private sets

Task	Exact		Relaxed	
	Development	Private	Development	Private
Task 1a	0.9548	0.9412	0.9677	0.9535
Task 1b	0.9179	0.8865	0.9294	0.9027

examples (Table 3.5). It is quite difficult for our machine learning framework to identify such relations with a small amount of examples.

We found errors caused by a lack of document-level information. Figure 3.2 shows a confusion matrix for compounds in Task 1a. Examples included errors among STARTING_MATERIAL, REAGENT_CATALYST, and SOLVENT and errors between REACTION_PRODUCT and OTHER_COMPOUND. In addition, errors between a solvent for a reaction (SOLVENT) and one for a work-up (OTHER_COMPOUND) were found because distinguishing between them from just one sentence was sometimes difficult. Errors between REACTION_STEP and WORK_UP for Task 1b were also found to be caused by the same difficulty.

With the aim of improving our systems, we tried to consider the relationships between the tasks in some preliminary experiments. For example, we tried to construct postprocessing rules for named entity mentions in Task 1a by on the results for Task 1b. However, these rules did not improve the results because it was difficult to determine which prediction (named entity or event) was correct.

Several errors were caused by failures in sentence splitting. For example, STARTING_MATERIAL "Ex. 18A" caused a split into two sentences by the ChemDataExtractor sentence splitter because of the "." in "Ex. 18A". A solution to this problem would involve applying a set of rules involving dependency parsing and trigger words.

3.5 Comparison with other systems

The baseline system employed a vanilla BERT-based system (not a domain specific model such as ChemBERT). The baseline for Task 1a obtained an exact match F-score of 0.9367, which is 0.0045 lower than ChemBERT with post-processing. The baseline for Task 1b

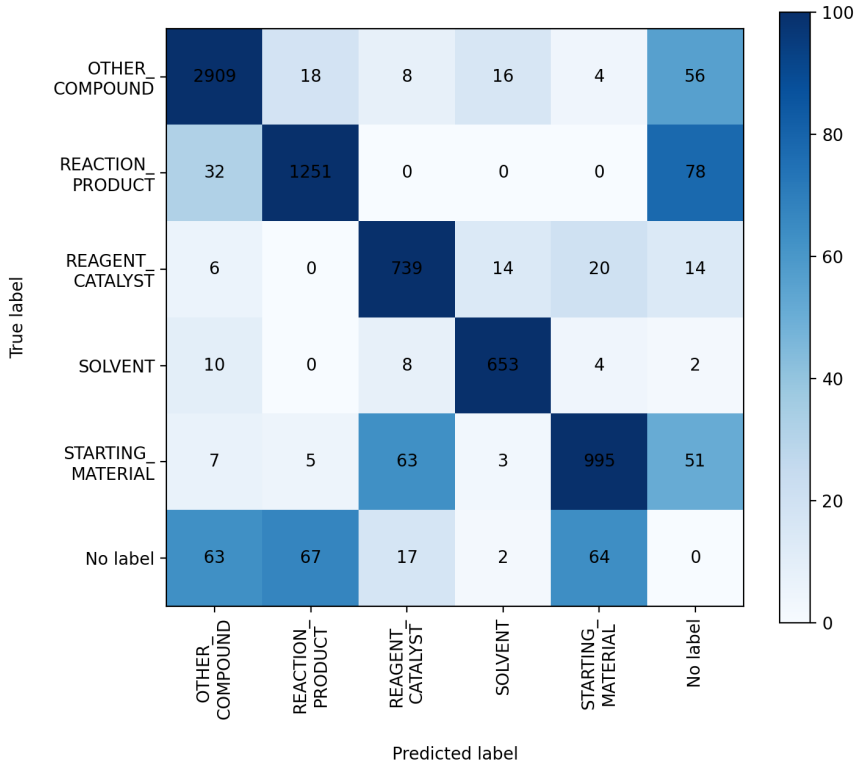


Figure 3.2: Confusion matrix for compounds in Task 1a

obtained an exact match F-score of 0.9088, which is 0.0223 higher than ChemBERT-based method.

There are three teams joined Task 1a and 1b: our team, LG AI research (LG) and Virginia Commonwealth University (VCU). LG achieved the best scores in both Task 1a and 1b, an exact match F-score of 0.9673 and an exact match F-score of 0.9199. They employed domain specific BERT-based system (BioLinkBERT [43]) with additional pre-training on various chemical corpora. In addition, they used ensemble approach and rule-based post-processing. The higher scores they achieved compared to ours were considered to result from these efforts.

To consider information extraction system for our research (information extraction tasks in Chapter 4 and Chapter 5), our system can serve as the baseline model. To compare our system with ChEMU's BERT-based baseline, there were not large difference in Task 1a and Task 1b. Although LG showed higher f-scores than our system, they

employed complicated methods and their model has not been made publicly available. Therefore, we decided to use ChemBERT as the baseline.

3.6 Summary

This chapter described our results for Task 1a and 1b at ChEMU 2022. We proposed hybrid methods that used ChemBERT and a small set of postprocessing rules for these tasks. We employed a pipeline approach for Task 1b that combined mention detection and relation classification. Because we used only one or two sentences as the input to ChemBERT, this lack of document-level information suppressed the performance of the system. Although we confirmed that adopting a set of postprocessing rules was effective in considering document-level information, we also confirmed that the set of rules we used was insufficient. In addition, although we tried to use relationships between the tasks to improve performance, it was difficult to construct rules that did achieve improvements.

Chapter 4

Construction of the OSPAR corpus

This chapter describes the construction of the organic synthesis procedure with argument roles (OSPAR) corpus and rolesets for the automated annotation.

4.1 Introduction

The number of chemical reactions reported in scientific documents has rapidly increased. New chemical reactions have been collected in chemical reaction databases, such as Reaxys [44] and Scifinder^{n} [45]; however, this has been done by human experts and taken much time as well as high costs. To reduce this burden, the development of automated information extraction methods has been studied [46, 47, 48, 49, 12, 50, 13, 51, 29, 52]. While most of them are focused on core information, such as chemical entities and parameters, a few works attempted to extract actions in a reaction from documents.

For example, as mentioned in Section 2.1, ChEMU campaign aimed to extract such chemical reaction data from patent documents. Because they are interested in the reaction information at the level of the reaction database, they did not consider detailed action information. For example, their corpus did not distinguish the semantic roles of A and B in the following phrase: "*A is added to B*" because they just needed a chemical reaction formula derived as A + B. However, the order of addition, namely, A to B or B to A, is usually regarded as important information that is required to reliably reproduce the chemical reaction.

There were two works in the literature for extracting more detailed procedures in organic chemistry. The first one is proposed by Mehr et al. [18] (described in Section 2.4 and 2.5.1).

Vaucher et al. [14] proposed a text-generation task that generated an action sequence from an experimental procedure. While their machine learning-based system trained on a large pseudo-corpus and a small human-annotated corpus was more flexible and robust than rule-based systems, the users of their system have to read an original text to validate the generated procedure from various perspectives, such as safety and completeness. Therefore, we emphasize that not only are the structured data important, but also the annotated text, which is aligned with the structured data.

To address these issues, we propose a corpus for procedure extraction named OSPAR that has more detailed information with annotated text than ChEMU's corpus. ChEMU used semantic role labels based on PropBank [28] that represented relations between verbs and related terms. Although ChEMU simplified the semantic labels, the PropBank and the simplified labels were not sufficient enough to describe the detailed procedure. Under this perspective, in this work, we systematically review the semantic role labels of the original PropBank and propose new semantic rolesets that are suitable for representing chemical reactions and especially organic synthesis. Our annotation schema using the augmented semantic role labels is more general than the one used in the specific platform, but it is easy to convert the information to the specific format. The overview of OSPAR and rolesets is shown in Figure 4.1 (the procedure text is from [53]). We annotated actions and related entities in the organic synthesis procedure, as well as the semantic roles of the entities using rolesets. The rolesets were selected from PropBank if there were appropriate ones for the examples in the corpus. Otherwise, we defined the new rolesets. Because the texts in our corpus are aligned to the output of the action sequence, the users can validate the action sequence easier than the text-generation approaches.

In addition, we train a deep learning-based system using the corpus to evaluate the practical use of the corpus and conduct more detailed analyses. To elaborate on the generality and application possibility of our framework, we compare the output sequence of our corpus with a χDL example.

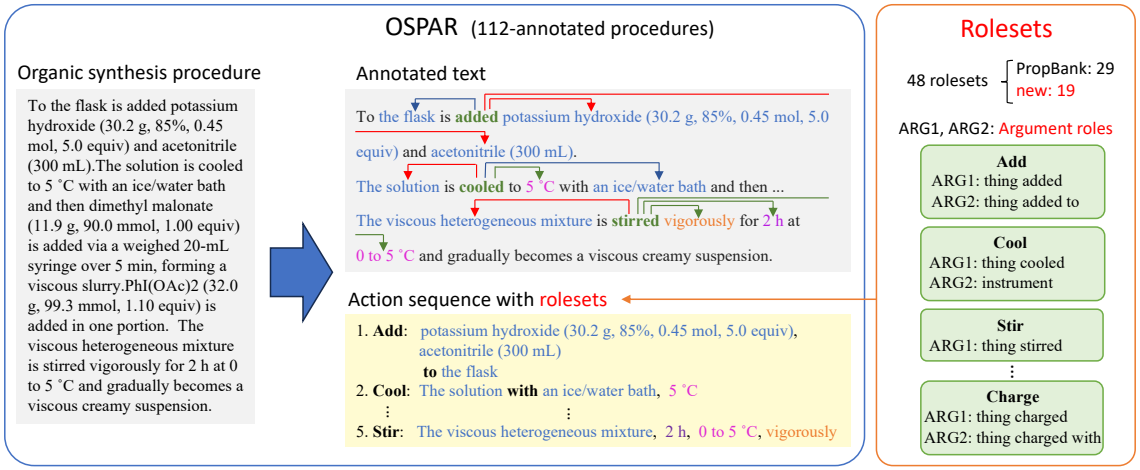


Figure 4.1: Overview of OSPAR and rolesets (verbs with argument roles).

4.2 Expansion of PropBank and corpus construction

4.2.1 Semantic roles and PropBank

As mentioned in Section 2.2, ARG0 generally represents the agent of the predicate, and Arg1 represents the patient. Other arguments more than Arg2 do not have consistent usage. While ARG0 shows the agent, we basically ignored this label in this work because the agent of action is obviously the experimenter. If rolesets are appropriately defined, the roles could be mapped to any schema for representing chemical actions similar to the arguments of functions in computer programming.

Because *The Wall Street Journal* articles take news related to business and economics, the rolesets in PropBank cannot cover actions in chemistry. Therefore, we need to add new rolesets for chemistry concurrently with constructing our corpus.

4.2.2 Task definition

In this work, we focus on chemical reaction procedures in organic synthesis at the level of rolesets. While the procedures usually consist of synthesis and work-up parts, we only used the synthesis part of the procedures. There were two reasons for this. (1) We focus on whether the product can be reproduced or not from the described procedure. Therefore, the method of work-up, which affects the yield of the products, is not important. (2)

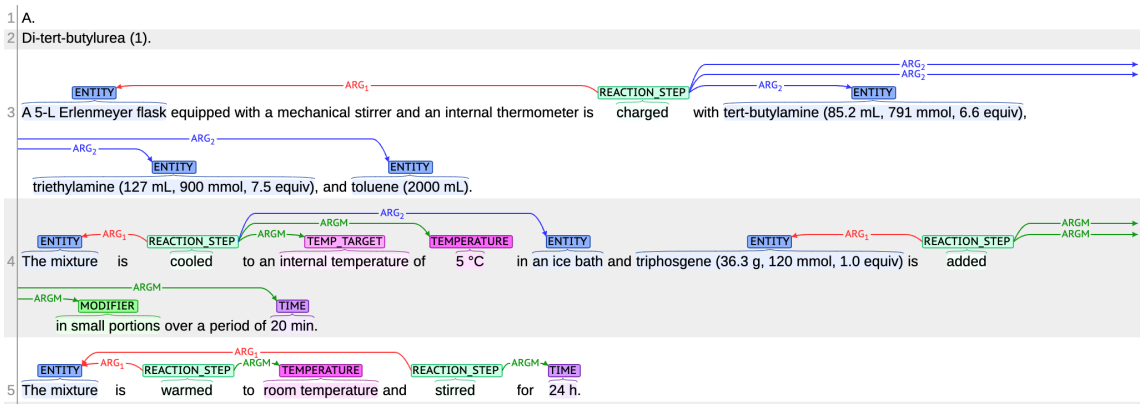


Figure 4.2: An example of annotated text.
*The procedure text is from [55].

Annotating work-up parts requires considerably more cost than the synthesis parts because the vocabulary in the work-up was much wider than synthesis. The preparation methods of instruments before starting a reaction (e.g., drying the instruments) are also out of scope in this work. This is because experimenters do not necessarily follow the methods as long as the state of the instruments is satisfied. Figure 4.2 shows an example of annotated text visualized by brat annotation tool [54]. The brat annotation tool is used for visualization of annotation after this. Our corpus consists of two tasks: named entity recognition (NER)[1] for finding spans of entities, actions, parameters, and relation extraction (RE) for finding relations between the action and entity/parameter.

4.2.3 Rolesets for actions in chemistry

As we discussed, rolesets defined in PropBank are not sufficient for describing the action in chemistry. Therefore, we propose new rolesets for actions in chemistry, based on PropBank. We checked the existence of appropriate entries in PropBank for all actions in the corpus. If we could find appropriate entries, these entries were included in our rolesets. However, there are several cases in the textual explanation about arguments that were not appropriate for describing chemical reactions. In such cases, we modified the text to reduce the ambiguity to interpret the meaning of the argument. While the rolesets in

[1]The term "NER" here does not refer to the ChEMU's NER task but rather originates from the general concept of NER as an information extraction task.

PropBank sometimes contained more than three n of ARGn, the rolesets in our corpus had up to two, which was enough to represent examples in the corpus. We ordered them to smaller numbers because labels with fewer examples harm the performance of machine learning-based systems. As a result, the rolesets in our corpus only contained ARG1 and ARG2 as ARGn. When there were no appropriate rolesets, we defined new rolesets.

4.2.4 Annotation schema

We used six labels for NER and four labels for RE. While there were four NER labels (REACTION_STEP, ENTITY, TIME, TEMPERATURE), which had the same name as labels in ChEMU tasks [13, 56], the definitions and target entities of the four labels were different from ChEMU's labels.

Annotation labels for NER are the following:

- REACTION_STEP: actions to transform the reactant to a product except for the actions for the preparation of instruments before starting a reaction, and work-up. When there were parallel actions, such as "**added** with **stirring**", **added** was annotated as REACTION_STEP but **stirring** was annotated as MODIFIER because the relation between actions is out of scope in this corpus for simplification.

- ENTITY: entities that have a semantic relation with a REACTION_STEP. The entities include chemical substances, gas, flasks, and other instruments.

- TIME: time that is related to REACTION_STEP. This contains the exact time and duration, such as **14-15h**. Expression in words, such as **overnight** is also annotated.

- TEMPERATURE: the temperature that is related to REACTION_STEP. This contains not only exact temperature but also range, such as **between 0°C and 5°C**. Expression in words, such as **room temperature** is also annotated. In addition, if a specific temperature is not described, but there is a word that indirectly represents temperature, such as **ice/water**, the word is annotated with this label.

- TEMP_TARGET: place to maintain TEMPERATURE. For example, **internal temperature** (inside the reactor) and **bath temperature**.

- MODIFIER: other information that is important to perform actions including atmosphere, way to add compounds, stirring rate, parallel actions, and others. For example, under **nitrogen**, added **in one portion**, added **dropwise**.

In addition to each label's definition, there are two policies for the annotation. The first one is if the same parameter was mentioned multiple times, only the first appeared parameter was annotated because the repetition was confusing in conversion to the action sequence. For example, the second "**12h**" in "... was stirred **12h**. After **12h**, ..." was not annotated. The second is the same entities and parameters, such as coreference and the amount of entities are annotated as one span. For example, "**12-aminododecanolactam** (**1**) (9.00 g, 45.6 mmol, 1.00 equiv)", (**1** is coreference of **12-aminododecanolactam**) and "room temperature (23°C)".

Annotation labels for RE are the following:

- ARG1, ARG2: semantic relation between REACTION_STEP and ENTITY. The semantic relation depends on each roleset but these labels are shared among the rolesets.
- ARGM: semantic relation between REACTION_STEP and parameters (TIME, TEMPERATURE, TEMP_TARGET, MODIFIER). We annotated the relation as ARGM even if the parameters were defined as ARGn in PropBank.
- (ARG0): semantic relation between REACTION_STEP and ENTITY. This represents the agent of an action. We basically did not use this label because most of the actions' agents are experimenters. However, we used only one case, ENTITY A of "A **contain** B", because ARG0 should be required to represent the container of the action.

4.2.5 Data curation and preprocessing

Organic syntheses reported in scientific journals should be reproducible from the description of the experimental procedure. If the description does not provide enough information to perform the reaction, experimenters cannot obtain the same result. *Organic Syntheses* [57] is one of the most reliable journals in organic synthesis because each procedure is

checked carefully for reproducibility in the laboratory of a member of the Board of Editors. The editors or their colleagues check carefully the reproducibility of each reaction by performing the reaction in their laboratory. Therefore, we decided to construct our corpus using procedures in this journal.

We collected articles, which had a "Procedure" section that described organic synthesis procedures, from annual volumes 81 (2005) to 97 (2020). While some articles contain multiple reactions to obtain the final product, we used only the procedure of the first reaction of each article for the corpus. This is because coreference resolutions were required if we handled procedures after the first reaction. In addition, we excluded procedures that have multiple boundaries of synthesis and work-up. Finally, we obtained 112 procedures (Table A.1). See Appendix A for article selection details.

We applied one preprocessing for the texts that were distinctive in *Organic Syntheses*. Here is an example phrase: *"anhydrous CH_2Cl_2 (150 mL) (Note 11) (Figure 1)"*. As shown in this example, additional information, such as *Notes* and *Figures*, was written down in raw texts. This is not common in other journals; thus, we removed these strings by using regular expressions.

4.2.6 Roleset expansion and corpus annotation

We constructed OSPAR and developed rolesets against actions in the corpus. The corpus was annotated by using the brat annotation tool [54]. This was done by three authors, two organic chemists (associate and assistant professor), and one information scientist (Ph.D. student).

First, we randomly selected 30 procedures to stabilize the annotation guideline. To reduce annotation cost, we used our system for ChEMU 2022 tasks as pre-annotation because several entities of our corpus had similar characteristics to ChEMU tasks. Then, we repeatedly annotated them and revised the guidelines. Next, we trained our system mentioned in the below section by using 30 procedures and deployed the system for the remaining 82 procedures as pre-annotation. Finally, we annotated the remaining procedures with the stabilized guideline.

The rolesets were added or modified for REACTION_STEP verbs concurrently with

Table 4.1: Number of procedures and sentences in each data set

Data set	Procedure	Sentence
Train	90	664
Development	11	86
Test	11	68
Total	112	818

the annotation process. The modification included simple textual modification and edition of the roles, which were affected by the change of the definition of arguments in our corpus from PropBank. For example, “ingredient one” (ARG1) and “ingredient two” (ARG2) from “mix.01” in PropBank were aggregated as “ingredient” (ARG1) in our corpus because the difference of ingredients in PropBank came from a syntactic perspective, but it was not important in terms of chemistry. When we searched rolesets in PropBank, we used the natural language toolkit (NLTK) [58]. To distinguish verbs regardless of the tense, we used Wordnet lemmatizer [59] via NLTK. However, several REACTION_STEP could not be lemmatized when the verbs were not registered in Wordnet lemmatizer or written in noun form, such as **addition**. Therefore, we constructed a dictionary for these words. Finally, all REACTION_STEP in the corpus was given any roleset. While verbs in PropBank can have multiple rolesets, there was no verb that has multiple roleset in the corpus. All rolesets used in the corpus were given independent suffix numbers from PropBank and numbered as 01.

To separate the synthesis and the work-up parts, we defined the boundary as the end of the chemical transformation from the reactant to the product. In general, operations after the chemical transformation can be regarded as work-up operations.

4.2.7 Corpus statistics

The annotated procedures were split into training set, development set, and test set at the ratio of 8:1:1. Table 4.1 shows the number of procedures and sentences in each data set. Here, the number of sentences included only the synthesis part of a procedure.

Table 4.2 shows the number of NER labels in each data set. While REACTION_STEP and ENTITY had many examples, the parameters had fewer examples, especially

Table 4.2: Number of NER labels in each data set

Label	Train	Development	Test
REACTION_STEP	590	66	59
ENTITY	900	108	100
TEMPERATURE	225	27	21
TEMP_TARGET	48	10	4
TIME	218	31	24
MODIFIER	161	17	15
Total	2142	259	223

Table 4.3: Number of RE labels in each data set

Label	Train	Development	test
ARG1	611	68	61
ARG2	361	48	44
ARGM	648	81	63
ARG0	1	0	0
Total	1621	197	168

TEMP_TARGET.

Table 4.3 shows the number of RE labels in each data set. ARG2 had fewer examples than the others because some rolesets did not have ARG2, and ARG2 sometimes was not used by REACTION_STEP even if ARG2 was included in the roleset. ARG0 was only used for **contain.01**.

4.2.8 Roleset analysis

We obtained 48 rolesets from the corpus (Table 4.4). To evaluate the effectiveness of roleset expansion, we classified them into four types:

(A). Same as original PropBank or slightly modified string surface, or reduced and ordered arguments of the existing roleset in PropBank.

(B). Affected by the change in the definition of arguments from PropBank.

(C). New roleset but the verb existed in PropBank.

(D). New roleset and the verb did not exist in PropBank.

Table 4.4: List of rolesets

Roleset	ARG1	ARG2	ARG0	Type	PropBank
activate	thing activated			A	activate.01
add	thing added	thing added to		A	add.02
contain	contents		container	A	contain.01
dissolve	thing dissolved	liquid		A	dissolve.01
distill	thing distilled			A	distill.01
fill	thing filled	thing filled with		A	fill.01
heat	thing heated	instrument		A	heat.01
hold	thing held			A	hold.11
immerse	thing immersed	thing immersed in		A	immerse.01
inject	thing injected	place injected into		A	inject.01
introduce	thing introduced	place introduced into		A	introduce.01
keep	thing kept			A	keep.01
maintain	thing maintained			A	maintain.01
melt	thing melted			A	melt.01
open	thing opened			A	open.01
place	thing put	where put		A	place.01
pour	thing poured	thing poured into		A	pour.01
protect	thing protected	thing protected from		A	protect.01
remove	thing removed	thing removed from		A	remove.01
replace	old thing	new thing		A	replace.01
stir	thing stirred			A	stir.01
transfer	thing transferred	thing transferred to		A	transfer.01
use	thing used			A	use.01
wash	thing washed	liquid		A	wash.01
wrap	thing wrapped	thing wrapped in		A	wrap.01
combine	thing combined			B	combine.01
increase	thing increased			B	increase.01
mix	thing mixed			B	mix.01
raise	thing raised			B	raise.01
change	thing changed	thing changed to		C	
charge	thing charged	thing charged with		C	
chill	thing chilled	instrument		C	
collect	thing collected	thing collected in		C	
cool	thing cooled	instrument		C	
dilute	thing diluted	thing diluted with		C	
flush	target	liquid or gas		C	
load	thing loaded	thing loaded with		C	
prepare	thing prepared	thing prepared in		C	
purge	thing purged	thing purged with		C	
warm	thing warmed	instrument		C	
backfill	thing backfilled	thing backfilled with		D	
degas	thing degassed			D	
pour off	thing poured off			D	
reflux	thing refluxed			D	
rinse	thing rinsed	thing rinsed with		D	
swirl	thing swirled			D	
re-cool	thing recooled	instrument		D	
re-charge	thing recharged	thing recharged with		D	

Table 4.5: Distribution of the types of rolesets

	(A)	(B)	(C)	(D)
Top 15	7	0	5	3
All (48)	25	4	11	8

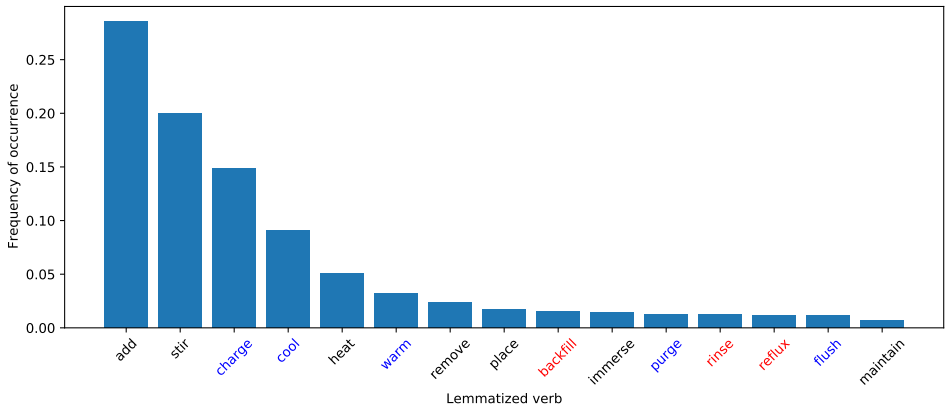


Figure 4.3: Distribution of rolesets in the corpus (most frequent 15).

Table 4.5 shows the distribution of types for the top 15 frequent rolesets and all rolesets. There were 52 % of (A) rolesets and 40 % of newly defined (C + D) rolesets. Therefore, the vocabulary of PropBank was insufficient for chemistry, and we confirmed the importance of expanding rolesets for chemistry.

Figure 4.3 shows the distribution of rolesets in the corpus. Here, the verbs in black indicate Type A, blue indicate Type C and red indicate Type D. We confirmed that the distribution of verbs was unbalanced as shown in related works [18, 14]. This is because the actions in organic synthesis are limited. It is worth noting that **charge** and **cool**, which were the third and fourth frequent rolesets, were categorized into Type C.

4.3 Experiment

We constructed an information extraction system by using the OSPAR to verify the practical use of the corpus and conduct a more detailed analysis.

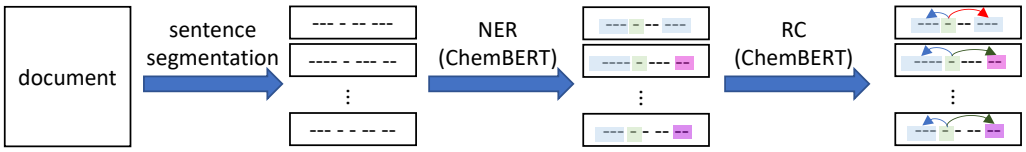


Figure 4.4: Overview of the pipeline system

4.3.1 System description

Recent deep learning models with pretraining have shown remarkable performances in natural language processing tasks [35, 8, 60]. We developed a pipeline system for NER and RE tasks that used ChemBERT v3.0 [29], a pretrained language model for chemistry, similar to our work for ChEMU 2022 task (Chapter 3). Figure 4.4 shows the pipeline method. First, we split a document into sentences by using ChemDataExtractor [12]. Then, we used ChemBERT models for NER and another ChemBERT model for RE. We used AllenNLP [40] and Hugging Face Transformers [41] for implementing ChemBERT.

For NER, we trained three ChemBERT models for: ENTITY, REACTION_STEP, and parameters. This is because each label can be overlapped and a simple ChemBERT model cannot take the overlapped entities. The input was tokens in a sentence that was split by simple regex rules [38]. The output was IOB2 labels with a class of each token. The labels were predicted in the same method as the original BERT [8]. For the optimization in training, AdamW optimizer [42] and cross–entropy loss were used. The hyperparameter values for the training are the same as our previous work [22]: max sequence length=384 (this value is enough to cover all sequences contained in the training and development sets), batch size=16, learning rate=1e-5. We applied early stopping at the training with patience=7 and used the best F-score model on the development set for evaluation.

For RE, we trained one ChemBERT model. The model solved the RE task as a classification problem of the relation of two entities [61]. ChemBERT predicted the labels of all candidate pairs in a sentence. The input of ChemBERT was a sentence with special tokens; a REACTION_STEP was [E1] and [/E1] tokens and an ENTITY or a parameter enclosed by [E2] and [/E2] tokens. The output was a relation label or the label to show there was no relation. In the training stage, we used gold-standard entities of the corpus.

Table 4.6: NER results on the test set

Label	Exact match			Relaxed match			Examples
	Precision	Recall	F-score	Precision	Recall	F-score	
REACTION_STEP	0.9474	0.9153	0.9310	0.9474	0.9153	0.9310	59
ENTITY	0.8447	0.8700	0.8571	0.8932	0.9200	0.9064	100
TEMPERATURE	0.6786	0.9048	0.7755	0.7500	1.0000	0.8571	21
TEMP_TARGET	0.5000	0.7500	0.6000	0.6667	1.0000	0.8000	4
TIME	0.9231	1.0000	0.9600	0.9231	1.0000	0.9600	24
MODIFIER	0.8571	0.8000	0.8276	0.9286	0.8667	0.8966	15
all	0.8504	0.8924	0.8709	0.8889	0.9327	0.9103	223

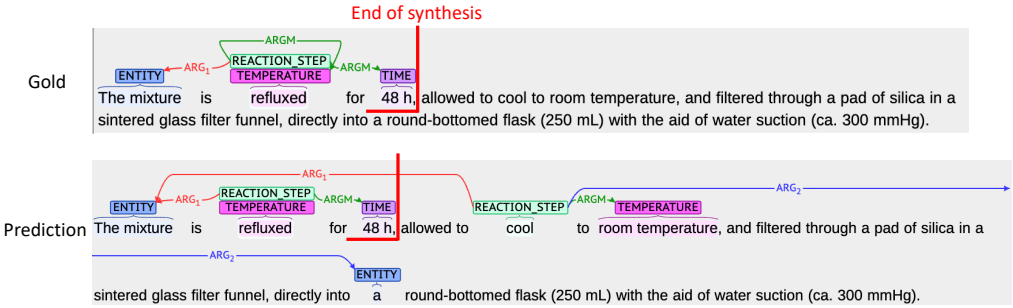


Figure 4.5: Incorrect prediction due to the boundary of synthesis and work-up

Other settings were the same as NER.

4.3.2 Evaluation metrics

We used Precision, Recall, and F-score described in Section 3.2.4 for the evaluation of the system. To evaluate close failures, we used a relaxed match in addition to an exact match.

4.3.3 Results and discussion

Table 4.6 shows NER results on the test set. While our corpus was not large, the NER model obtained an F-score of 0.8709 in an exact match. The model obtained an F-score of 0.9103 in a relaxed match, which was 0.04 greater than the exact match. This means the existence of some errors in the detection of entities' spans.

The precision was relatively lower than the recall. One of the reasons was that there were a few sentences that contained both synthesis and work-up actions. In such cases, the model incorrectly predicts work-up actions and related entities, even though the actions for the work-up are not annotated in the corpus (Figure 4.5).

Table 4.7: Relation classification results on the test set

Entity	Precision	Recall	F-score
Predicted	0.8393	0.8393	0.8393
Gold standard	0.9408	0.9464	0.9436

Table 4.8: Recalls of each verb (REACTION_STEP) on the test set

Verb	Recall	#test	#train	#development
add	1.0	18	168	18
stir	1.0	12	119	12
charge	1.0	8	86	12
cool	1.0	4	57	4
heat	1.0	3	31	2
remove	1.0	1	16	0
backfill	1.0	1	9	1
place	1.0	2	9	1
reflux	1.0	1	4	3
transfer	0.0	1	3	0
dissolve	0.0	1	3	0
fill	1.0	1	1	0
wrap	1.0	1	1	0
wash	0.0	1	1	0
hold	**1.0**	**1**	**0**	**0**
mix	**1.0**	**1**	**0**	**0**
open	0.0	1	0	0
activate	0.0	1	0	0

*#test, #train and #development are the number of occurrence in each data set

Table 4.7 shows RE results on the test set. The model with predicted entities obtained an F-score of 0.8393 in an exact match. On the other hand, the model with gold-standard entities obtained an F-score of 0.9436 in an exact match. We recognized the importance of NER performance from this difference (approximately 0.1) was not small. On the other hand, the results of NER and RE suggest the consistency of the annotation of the corpus and the usefulness of the corpus.

As expected, our system tends to fail in extracting entities that are seen zero or a few times during training. On the other hand, we found interesting cases in that our system succeeded in finding such entities. Table 4.8 shows recalls of each verb (REACTION_STEP) on the test. All verbs that existed more than three times on the train set

were found in the test set. In fewer verbs, “fill” and “wrap” were found even though the system saw them only once in the training phase. Although “hold” and “mix” were not seen in the training phase, they were found by the system. Figure 4.6 shows the prediction results for the unseen verbs. “hold” was exactly matched in both NER and RE. This could

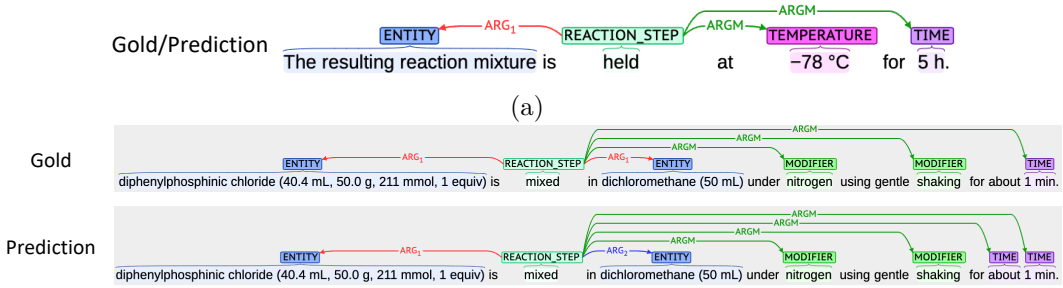


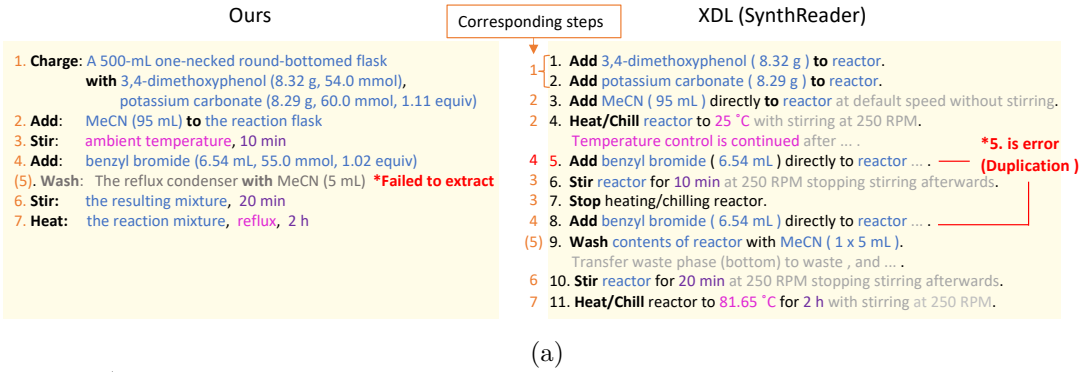
Figure 4.6: Prediction results for the unseen verbs (a) hold (b) mix

be explained because similar cases were included in the train set, for example, “keep” and “maintain”. Although the roleset “mix.01” consists of only ARG1 (“thing mixed”), ARG2 was predicted for “dichloromethane (50 mL)” (we did not distinguish ingredients one and two in the roleset “mix.01” because the differences were not important in the context of conducting the synthesis procedure). This may be caused by similar examples, such as “A was heated in B” and “A was cooled in B”. In these cases, A is ARG1 (thing heated/thing cooled) and B is ARG2 (instrument). Regardless of the predicted labels, the prediction can be corrected by referring to the arguments of the roleset.

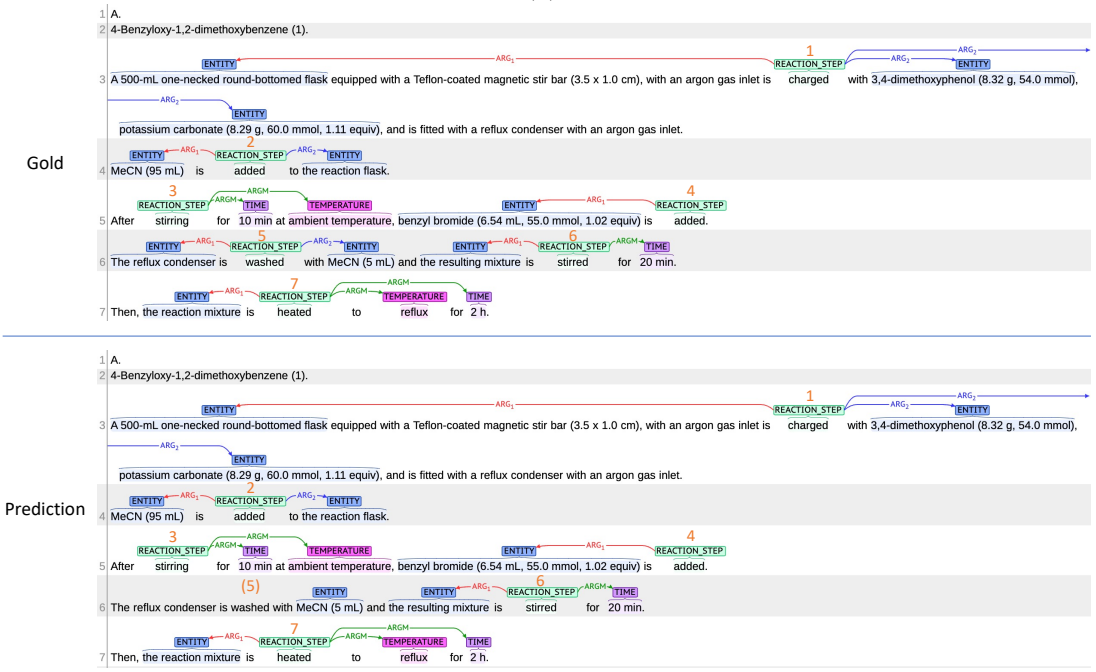
Comparison with χDL

To investigate the completeness of our schema, we selected one example from the extracted sequence by our system and compared it with χDL generated by using SynthReader via ChemIDE [21]. SynthReader is an information extraction system for χDL and showed high performance for articles in *Organic Syntheses* [57]. Figure 4.7 shows a comparison between our method and χDL generated by SynthReader.[2] Here, the numbers over the REACTION_STEP were manually drawn by us, and some detailed explanation for robotic

[2]The procedure text is from [55].



(a)



(b)

Figure 4.7: Comparison between our method and χDL generated by SynthReader. (a) action sequence for both methods (b) gold standard and prediction by our system visualized by using brat

synthesis in χDL was omitted for simplicity of comparison. We aligned the corresponding steps from ours and χDL (Figure 4.7a). Here, our system failed to extract the fifth action "Wash", and SynthReader wrongly generated duplicated actions (the 5th and 8th action "Add". the 5th one was incorrect). By assuming that both systems could correct the action sequence, ours can be basically mapped to the actions in the χDL sequence. This implies that our schema with argument roles has the potential to represent general and enough information for organic synthesis.

It is important to align the annotation with the text. Figure 4.7b shows the visualized texts of gold standard and prediction. If we verify the predicted action sequence, we can easily understand actions, related entities, and failure of extraction. χDL by SynthReader contained more detailed information than ours, for example, the temperature of the final action "reflux" was interpreted as "81.65°C" by using chemical knowledge. However, there are some mistakes in the χDL. As mentioned above, the 5th action, "Add", was not required. In addition, although the 9th action, "Wash", should be rinsing for the reflux condenser, it was interpreted as a separation action for compounds in the reactor. That is usually performed as work-up.

4.3.4 Limitations and future works

We have discussed the rolesets in the context of organic synthesis so far. We consider that it can be used in materials science [62, 63, 64, 65, 66, 67] because there are common actions in organic synthesis and materials science. On the other hand, we found that the variety of the rolesets is insufficient to represent actions in materials science. In such a case, we should expand rolesets for them. To enable this, we hope that not only our group but also other research groups will use this framework and enrich the rolesets and annotated corpus.

Finally, we discuss the limitations of this work. Because the relations among ENTITY in the corpus were unclear, anaphora resolution [68] (the problem of resolving what a pronoun or a noun phrase refers to) should be required to generate structured data for performing the reaction. Most of such relations are relations between a chemical compound and a mixture. One of the solutions is applying the task that is similar to ChEMU's

anaphora resolution task [56]. Another challenge is interpreting ENTITY for structured data. For example, distinguishing instruments and chemical compounds. In addition, the amount of chemical compounds should be parsed. To enable this, ChemicalTagger [5], a rule-based natural language processing parser for chemistry, can be used. Although ChemicalTagger is less flexible than deep learning-based systems, it can extract detailed information, such as amount and instruments, which follows notions included in its rules. Therefore, it can be useful for interpreting information in ENTITY. Next, variation and amount of examples in the corpus could be insufficient. We need to study the actions in different documents and compare them with the existing rolesets. In addition, we should increase the number of examples for rare rolesets.

4.4 Summary

To understand the relation between actions in chemical reaction procedure and the argument roles, we propose the rolesets in chemistry that can be used as a general procedure extraction task in organic/inorganic chemistry. We have constructed the corpus named OSPAR and evaluated it by training the system on the corpus. We confirmed the roleset was promising for expressing actions in reproducing a reaction by comparing it with χDL.

Chapter 5

A Framework for Extraction of χDL from the Literature

In Chapter 4, we have developed the OSPAR corpus and the system for the automated annotation of text. This chapter describes the proposed framework for the extraction of χDL from literature.

5.1 Overview

In Section 5.2, we introduce the user interface of our framework followed by the proposed rule-based conversion method. In Section 5.3, we describe an experiment we conducted to discuss the comparative advantages of our framework by comparing the conversion result of χDLs by CLAIRify and the proposed system. As a result, we confirmed that the proposed system could find action information, which was not feasible by CLAIRify. We also compare the proposed system with SynthReader and discuss the advantages of the proposed system against the existing rule-based system. We found that the proposed method performed better, in terms of finding explicit actions, which were important for the information extraction task, while SynthReader was better at finding implicit actions.

5.2 Methods

In this section, we first describe an existing schema for visualization. Then, we propose a user interface for human review and an automated conversion method from text to χDL.

5.2.1 The OSPAR format

We used the OSPAR format (described in Section 4.2) for visualization and as an intermediate representation to convert text to χDL. An example of the visualized annotation is shown in Figure 4.2.

5.2.2 User interface

We propose a user interface that consists of three main functions: (a) conversion from text to the OSPAR format by BERT, (b) conversion from the OSPAR format to χDL by rules, and (c) conversion from text to χDL by CLAIRify. Figure 5.1 shows the user interface. This editorial process starts by entering an organic synthesis procedure into the text box

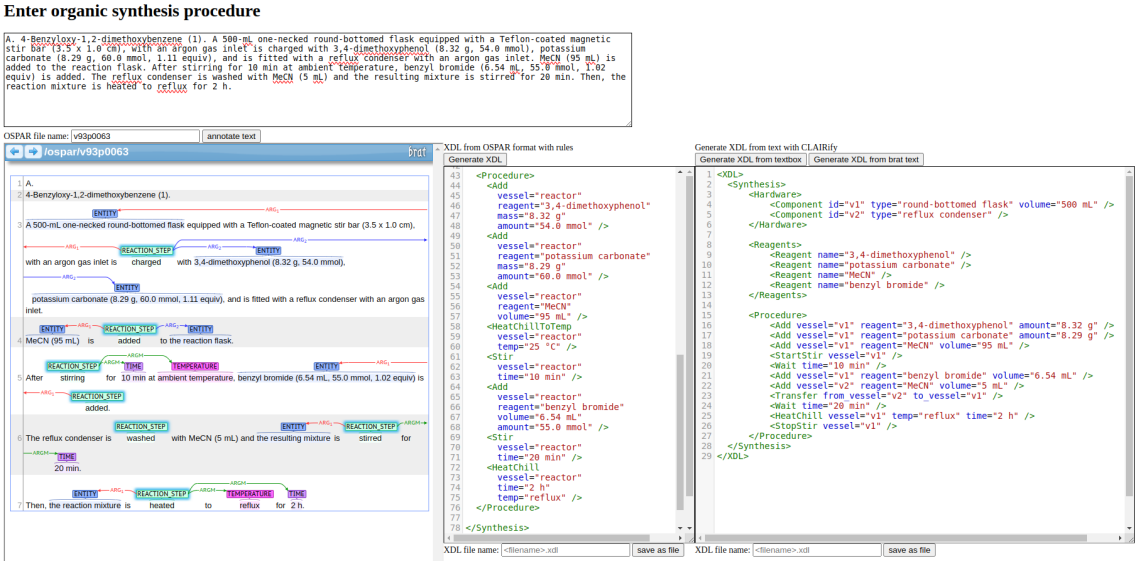


Figure 5.1: Screenshot of the proposed user interface
*The procedure text is based on Okaya et al. [69], with revisions made through preprocessing in the OSPAR corpus.

at the top of the screen.

After the user clicks the "annotate text" button, the text is converted to the OSPAR format and the result is visualized by brat [54], a web-based annotation tool. Then, the user can see the annotated text in the OSPAR format. If the user is not satisfied with the automatic annotation results, he/she can modify the annotation results by using brat as an annotation tool. This modification has the potential to improve the automated conversion

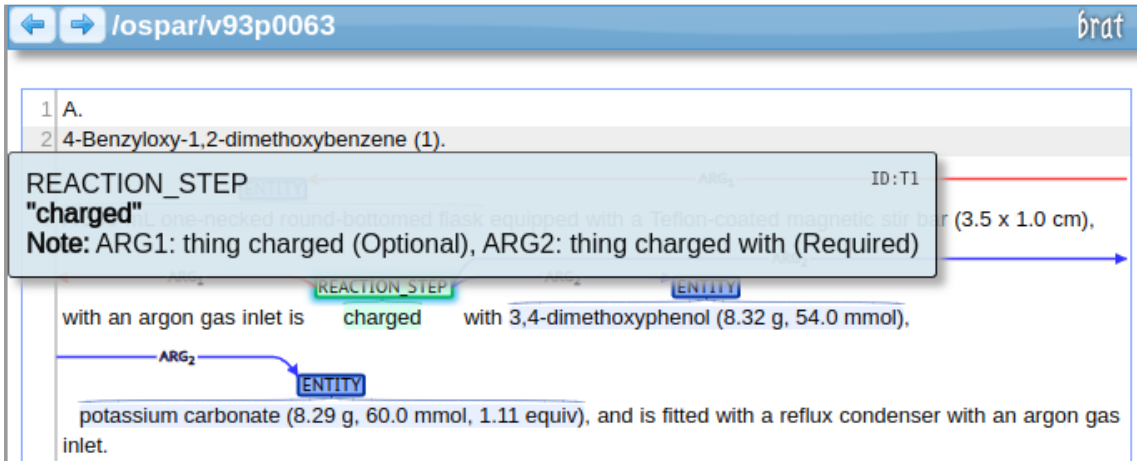


Figure 5.2: Screenshot of brat

method from the OSPAR format to χDL. By moving a cursor over REACTION_STEP, the user can view a roleset for the action to refer to the modification (Figure 5.2).

Then, the user clicks the "Generate χDL" button above the middle text editor to generate χDL from the OSPAR annotation which is displayed in the brat. The generated χDL can be edited and saved by clicking the "save as file" button as a filename beside the button. The text editor on the right is used for conversion from the text in the top textbox or text shown in brat to χDL by CLAIRify. The buttons around this editor have the same functions as the middle text editor. The user can compare both conversion results and select a better χDL as a base for the reviewing process. When the user finds some mistakes in the base χDL, the user can refer to the other χDL to check the existence of correct action for revising the base χDL.

The proposed user interface was implemented using Flask [70], a Python-based web framework. We used brat for visualization and annotation. While the original version was implemented with Python 2, we used the Python 3 version available at `https://github.com/nlplab/brat.git`. The text editor is implemented using CodeMirror [71].

5.2.3 Conversion from text to the OSPAR format.

We used a system that was similar to one described in Section 4.3.1. We trained ChemBERT models [29] for NER and another ChemBERT model for RE by using the training

set and development set of the corpus. The ChemBERT models were implemented using HuggingFace Transformers [41].

5.2.4 Conversion from the OSPAR format to χDL.

We converted the OSPAR format to χDL by mapping the arguments of roleset to χDL actions. We used χDL version 2.0.0 [72]. Figure 5.3 shows an example of the conversion. First, we defined candidate χDL actions for each roleset to map the rolesets to the χDL

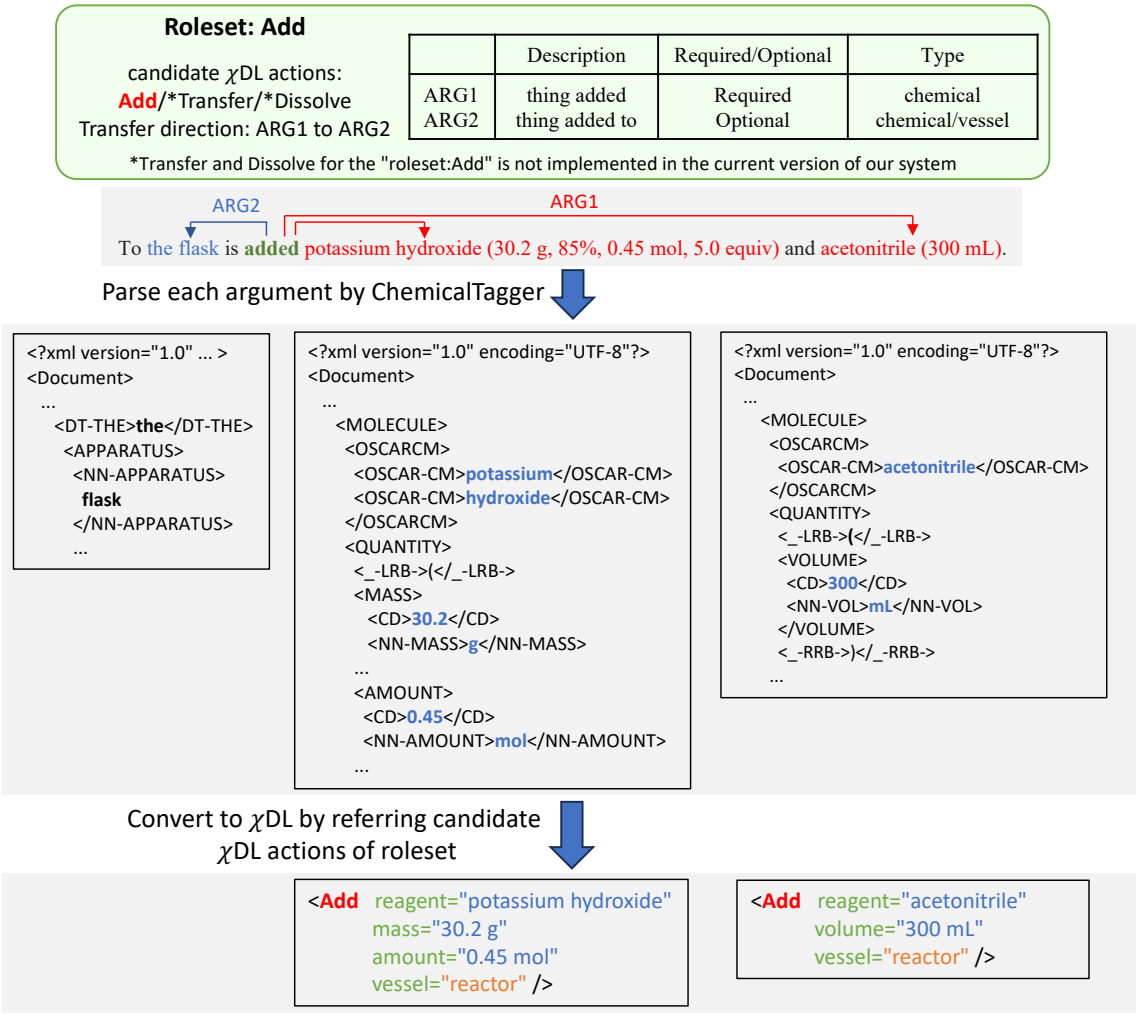


Figure 5.3: An example of conversion from the OSPAR format to χDL

actions. Then, to align the roleset arguments with χDL arguments, we specified the type and whether each roleset argument was "required" by referring to the target χDL

arguments. For liquid and solid handling actions of χDL such as `Add` and `Transfer`, the corresponding roleset has "transfer direction" information to determine the order of the χDL actions. We used ChemicalTagger [5] to obtain detailed information about the arguments, such as compounds, masses, and other parameters. Then, χDL actions were generated by referring to the roleset and the outputs of ChemicalTagger. When ARGM arguments such as TEMPERATURE or TIME were present, these parameters were either used to generate new actions (e.g., `HeatChillToTemp`) or to define specific χDL arguments (e.g., `time="1 h"`). In cases where an argument was a mixture, such as `a solution of sodium iodide (15.0 g, 100 mmol) in acetonitrile (100 mL)` (see Supplementary Information for details), a single argument could be converted into multiple χDL actions. The χDL actions were generated as instances of the χDL library, which automatically validated the χDL arguments. Finally, these instances were converted into XML format.

We used text normalization tools. To normalize the verbs, we used WordNet lemmatizer [59]. To interpret numbers in word form, we used text2num [73] to convert numbers in word form to numerical form. Additionally, we defined several constants to accurately interpret the parameters expressed in words and convert them into precise values (Figure 5.5).

Because it was difficult to capture multiple flasks (e.g., compounds A and B were mixed in a flask X and compounds C + D were mixed in a flask Y), we fixed vessel to reactor other than the case that OSPAR argument was a mixture and multiple compounds were written in the argument.

Alignment of the rolesets and χDL actions

There are two types of χDL actions that can be generated by OSPAR2χDL: corresponding to each roleset and the arguments of rolesets. Table 5.1 shows χDL actions that can be generated by OSPAR2χDL, along with the corresponding part of the OSPAR rolesets.

Conversion rules for mixtures in ARG1 and ARG2

In cases where an argument ARG1 or ARG2 is a mixture, a single argument may be converted into multiple χDL actions. When multiple chemical names are identified by

Table 5.1: χDL actions that can be generated by our system, along with the corresponding part of the OSPAR rolesets

χDL action	generated from
Add	roleset, ARG1, ARG2
Transfer	ARG1, ARG2
HeatChill	roleset
HeatChillToTemp	roleset, ARGM (TEMPERATURE)
Stir	roleset, ARGM (MODIFIER)
StartStir	roleset, ARGM (MODIFIER)
StopStir	ARGM (MODIFIER)
EvacuateAndRefill	rolset, ARGM (MODIFIER)
Purge	roleset

ChemicalTagger, the argument is considered as a mixture. For example, in the argument `a solution of sodium iodide (15.0 g, 100 mmol) in acetonitrile (100 mL)`, `sodium iodide` is identified as a reagent, and `acetonitrile` is identified as a solvent by ChemicalTagger. In this case, `Add` actions for each component into a temporary vessel for mixing are created. Then, when the argument is added to a reactor, a `Transfer` action from the temporary vessel to the reactor is created.

Details of the conversion for ARGM

ARGM arguments such as TEMPERATURE, TIME and MODIFIER are generally used as the parameter of χDL actions. However, ARGM arguments labeled with TEMPERATURE and MODIFIER may create χDL actions other than main χDL actions that created by a roleset. Figure 5.4 shows the rules for converting ARGM into χDL action(s). If ARGM is TEMPERETURE and the rolset does not have a `HeatChill` action among its candidate χDL actions, the ARGM generates a `HeatChillToTemp` action before the χDL actions by the roleset. If a MODIFIER is included in the predefined gas-related words, `EvaluateAndRefill` is created before the main χDL actions related to a roleset. If a MODIFIER is included in the predefined stirring-related words, the ARGM generates a `StartStir` action before the main χDL actions and a `StopStir` action after the main actions.

We created a dictionary to interpret words that represent parameters for TEMPERA-

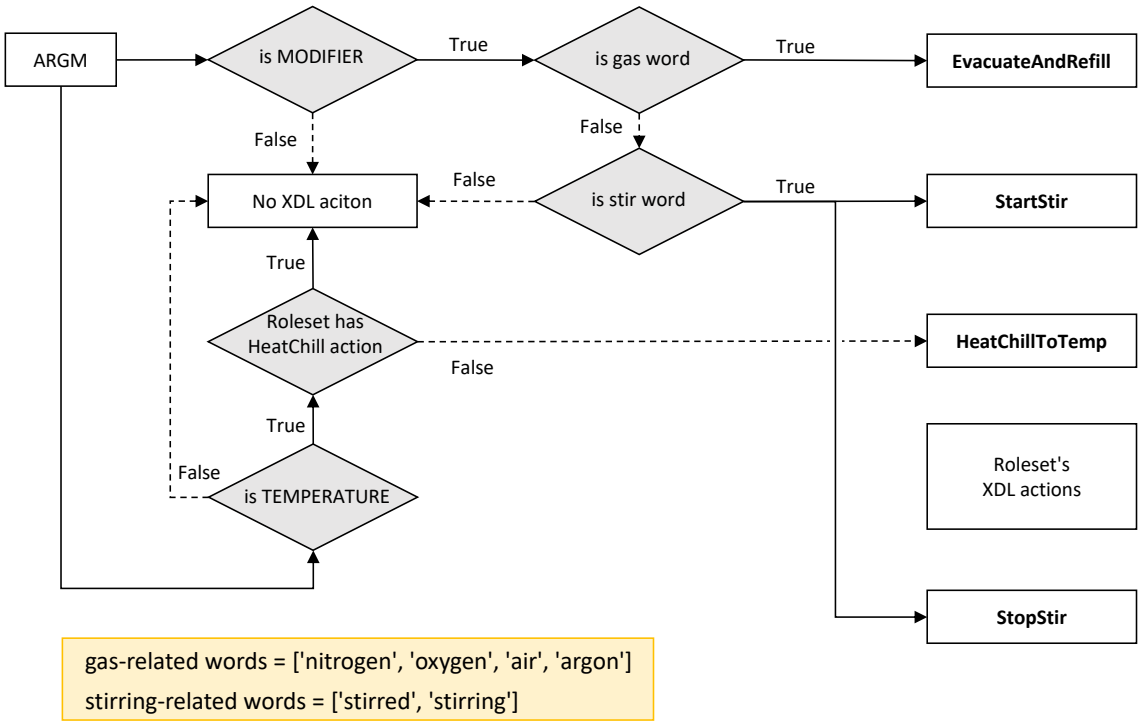


Figure 5.4: Rules for the converting ARGM into χDL action(s)

TURE, TIME, and stirring rates in MODIFIER (Figure 5.5).

Rules for detecting amounts, masses and volumes

To perform a reaction, one of amount, masses and volume is required at least. In general, we aim to extract as much information as possible from the text. Therefore, in the current version of our system, parameters in the text are detected by ChemicalTagger, and all detected parameters are mapped to the parameters of χDL actions by classifying them into χDL's parameters by the following rules:

- Molecular amounts such as `mol` and `mmol` are mapped to "amount" in χDL actions by using a `<NN-AMOUNT>` tag in ChemicalTagger.
- Masses such as `g` and `mg` are mapped to "mass" in χDL actions by using a `<NN-MASS>` tag in ChemicalTagger.
- Volumes such as `mL` are mapped to "volume" in χDL actions by using a `<NN-VOLUME>`

```
TEMPERATURE = {
    'room temperature': '25°C',
    'rt': '25° C',
    'ambient temperature': 25° C'
}

TIME = {
    'overnight': '16 h'
}

MODIFIER_STIR_RATE = {
    'vigorous': '500 rpm',
    'vigorously': '500 rpm'
}
```

Figure 5.5: Dictionary to interpret words that represent parameters.

tag in ChemicalTagger.

5.3 Experiment

As we mentioned in Section 1, recall is important for human review. To discuss the advantages of showing both the outputs of the proposed method and CLAIRify, we constructed six χDL examples and evaluated the recall of the methods on the examples. In addition, we compared the proposed method with SynthReader [18] as an existing rule-based method.

5.3.1 Construction of evaluation data

We selected organic synthesis procedures that can be performed by an automated robot (Chemspeed platform), from *Organic Syntheses* [57] for the examples. The χDL examples were constructed by three authors: two organic chemists (associate and assistant professors) and one information scientist (PhD student). To evaluate the effect of the automated annotation quality on the OSPAR format, we required annotated texts that were checked by humans. Because there were only four examples in the test set of the OSPAR corpus, we additionally annotated two examples other than the corpus in the same manner as the OSPAR corpus including text preprocessing.

To evaluate each method by considering only actions that were explicit in the text, we labeled implicit actions. We labeled each action into an explicit or implicit action to distinguish them in the evaluation phase because errors from these types of actions had different meanings in an information extraction task. There are two types of implicit actions: (a) initiating stirring after addition of reagents and solvents (`StartStir`) and (b) stirring for a certain period of time (`Stir`). We did not consider creating a mixture in a noun phrase such as `a solution of sodium iodide (15.0 g, 100 mmol) in acetonitrile (100 mL)` because the actions were embedded in the text, unlike the abovementioned actions. When creating the correct χDL, if stirring continues after a Stir, it is treated as an implicit action. Therefore, instead of setting `continue_stirring=True` as an argument for `Stir`, it was represented as `Stir` and `StartStir`, with `StartStir` being treated as an implicit action. While we annotated the stopping stirring (`StopStir`) or heating (`StopHeatChill`) when the target vessels were not used in subsequent steps, we excluded these actions from the evaluation as they were not critical to reproduce the procedure.

5.3.2 Experimental settings

We compared four methods in this experiment: SynthReader, the proposed method from text to χDL (Pipeline), the proposed method from human-annotated OSPAR format to χDL (OSPAR2χDL), and CLAIRify. To compare the proposed method with/without human intervention, we used Pipeline and OSPAR2χDL. We used SynthReader via a web interface called ChemIDE [21]. We used CLAIRify downloaded from github [74]. While the original CLAIRify used GPT-3.5, we used GPT-4o (gpt-4o-2024-05-13) as an LLM because later models were considered to be better than older models. Because the original implementation did not work on the current version of OpenAI API, we made a minor revision of the source code.

In addition to evaluating each system individually, the combination of CLAIRify with other systems for a practical situation by human review was also examined. The combined recall was calculated by verifying whether the correct answer was present among the χDL actions produced by independently running the two systems. To evaluate close failures, we defined action recall in addition to exact recall. The definitions were the following:

- Exact recall: the proportion of correct actions with only correct parameters among the actions in gold data.
- Action recall: the proportion of correct actions with correct parameters and correct actions with missing/wrong parameters among the actions in gold data.

Examples of evaluating correct actions in exact recall and action recall are shown in Figure 5.6. While missing and/or wrong parameters are not allowed in exact recall, they

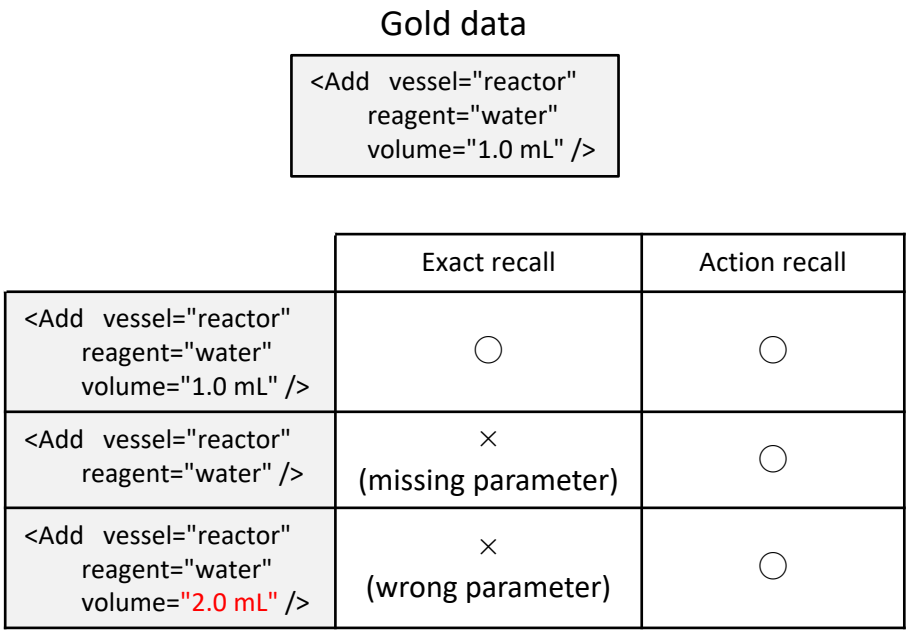


Figure 5.6: Examples of evaluating correct actions in exact recall and action recall

are allowed in action recall.

Other evaluation criteria were the following:

- Liquid/solid handling actions: It was considered as correct action if either mass or volume was specified in an action even if both mass/volume and amount were mentioned in the text. It was acceptable if `dropwise=True` was not specified for the `Add` action. It was also acceptable to create the initial mixture in the reactor.
- Stirring actions: When mass or volume were mentioned multiple times in the text, it was considered correct if either mass or volume was specified in an action.

Table 5.2: Result of explicit actions

	SR	Pipe	O2X	CLAIR	SR+CLAIR	Pipe+CLAIR	O2X+CLAIR
exact recall	22/65	28/65	31/65	38/65	45/65	48/65	49/65
action recall	34/65	41/65	50/65	60/65	60/65	60/65	60/65

*The numbers indicate (#found action)/(#all actions). SR is SynthReader, Pipe is Pipeline, O2X is OSPAR2χDL and CLAIR is CLAIRify.

- Temperature control actions: For HeatChill, cases like "between –15°C and –5°C" were considered correct if the temperature was set within that range.

5.3.3 Results and discussion

Evaluation for explicit actions.

Table 5.2 shows the result of explicit actions.

To see individual systems, CLAIRify showed the best performance in both exact and action recall. While CLAIRify demonstrated a high action recall (60/65), its exact recall was relatively modest (38/65). This was because that CLAIRify sometimes failed to extract the units of the parameters or even the parameters themselves. We found that when CLAIRify failed to extract the parameters, CLAIRify consistently did not extract the parameters in the example procedure (Figure 5.7). Such characteristics were observed in two of the six procedures.

The proposed Pipeline and OSPAR2χDL showed higher recall than SynthReader in both exact and action recalls. While the evaluation data were small, we confirmed that the OSPAR format could represent enough information and the proposed rule-based conversion was better than SynthReader. A major reason why Pipeline was better than SynthReader was because Pipeline employed BERT-based NER and RE in contrast to rule-based SynthReader, which could not extract entities and relations absent from its templates. As a result, we observed differences in the recall of liquid handling actions such as `Add` and `Transfer`, which require recognizing compound names (Table 5.3).

We confirmed that the modification of the OSPAR annotation could improve the conversion quality because actions, which were not extracted by Pipeline, were sometimes found by OSPAR2χDL, in terms of both exact and action recall. The main reason for the difference is that ChemBERT sometimes failed to extract compounds. In addition, errors

Table 5.3: Result of explicit actions for each category

Liquid handling							
	SR	Pipe	O2X	CLAIR	SR+CLAIR	Pipe+CLAIR	O2X+CLAIR
exact recall	14/42	18/42	21/42	26/42	31/42	33/42	33/42
action recall	23/42	28/42	36/42	40/42	40/42	40/42	40/42
Stirring							
	SR	Pipe	O2X	CLAIR	SR+CLAIR	Pipe+CLAIR	O2X+CLAIR
exact recall	3/7	5/7	5/7	4/7	4/7	5/7	5/7
action recall	4/7	5/7	5/7	6/7	6/7	6/7	6/7
Temperature control							
	SR	Pipe	O2X	CLAIR	SR+CLAIR	Pipe+CLAIR	O2X+CLAIR
exact recall	4/8	5/8	5/8	2/8	4/8	5/8	5/8
action recall	6/8	6/8	6/8	8/8	8/8	8/8	8/8
Inert gas							
	SR	Pipe	O2X	CLAIR	SR+CLAIR	Pipe+CLAIR	O2X+CLAIR
exact recall	0/7	0/7	0/7	5/7	5/7	5/7	5/7
action recall	0/7	2/7	3/7	5/7	5/7	5/7	5/7
Special							
	SR	Pipe	O2X	CLAIR	SR+CLAIR	Pipe+CLAIR	O2X+CLAIR
exact recall	1/1	0/1	0/1	1/1	1/1	1/1	1/1
action recall	1/1	0/1	0/1	1/1	1/1	1/1	1/1

*The numbers indicate (#found action)/(#all actions).

in identifying entity boundaries led to missing parameters. As a result, the recall of liquid handling actions of Pipeline was lower than that of OSPAR2χDL (Table 5.3). To improve the information extraction system from the text to the OSPAR format, for example, increasing training data and improving a deep learning-based model were required. If the user annotates procedures for χDL conversion, the annotated procedure can be used as training data of the models for text to the OSPAR format.

There were two common errors that were difficult for rule-based methods when converting the OSPAR format to χDL. The first was an incorrect target vessel for actions because the proposed rules could not consider multiple vessels, as described in Section 5.2.4. The second was the missing quantity or mass, as ChemicalTagger failed to determine which parameters belonged to which molecule. For example, `o-Tolylboronic acid, 10.0 g (73.6 mmol)` was not correctly parsed due to the comma after `o-Tolylboronic acid`. While the proposed method was sensitive to the notations of the OSPAR arguments in the text, CLAIRify was robust to these notations thanks to the flexibility of a GLLM.

Table 5.4: Result of implicit actions

	SR	Pipe	O2X	CLAIR	SR+CLAIR	Pipe+CLAIR	O2X+CLAIR
exact recall	2/12	0/12	0/12	6/12	7/12	6/12	6/12
action recall	2/12	0/12	0/12	6/12	7/12	6/12	6/12

*The numbers indicate (#found action)/(#all actions). SR is SynthReader, Pipe is Pipeline, O2X is OSPAR2χDL and CLAIR is CLAIRify

We also confirmed that combining results of rule-based methods and CLAIRify was effective in increasing exact recalls. To compare the exact recalls of the CLAIRify combined with other systems, Pipeline and OSPAR2χDL showed better results than SynthReader (SynthReader found 7 new actions, Pipeline found 10 actions and OSPAR2χDL found 11 new actions). On the other hand, action recalls of the combined results did not increased by three methods. This result indicates the improved capability of CLAIRify for finding actions. Figure 5.7 shows an example of how other systems can improve CLAIRify's recall. In this example, the amount or volume of reagents was not specified by CLAIRify. In other cases, CLAIRify failed to extract the time of addition or the stirring rate, even though these parameters were clearly stated in the text.

Effect of modifying the annotation

Figure 5.8 shows the effect of modifying the annotation. In the annotation by ChemBERT, `PPh3` and `N,N-dimethylformamide` were not annotated. This led to a lack of corresponding χDL actions. Another error occurred due to an incorrect boundary in `Pd(OAc)2 (180 mg, 0.80 mmol, 0.01 equiv)`. Because the closing parenthesis at the end of `Pd(OAc)2 (180 mg, 0.80 mmol, 0.01 equiv)` was missed by ChemBERT, the proposed rules failed to extract `mass` and `amount`. As a consequence, these parameters were missed in a χDL action by the pipeline system. Both types of errors may be addressed by increasing the training data in future.

Evaluation for implicit actions.

Table 5.4 shows the result of implicit actions. As we expected, the proposed Pipeline and OSPAR2χDL could not find implicit actions because the OSPAR format did not consider actions that were not written in text and we did not make rules to complement such

actions when converting the OSPAR format to χDL. SynthReader could find implicit actions because rules to complement such actions were included in SynthReader. We confirmed that CLAIRify could find implicit actions by the capability of a GLLM.

To enable finding implicit actions by the proposed methods, we need to construct rules to capture these actions in future work. For example, inserting `StartStir` following multiple `Add` actions would be considered. Another example is inserting `StopStir` when the reactor or flask was not used after the previous action.

What should be considered in the human review process?

As we discussed, we found that CLAIRify was generally promising for generating a "base χDL" because it demonstrated higher recalls than other systems. However, the output of the proposed system may be preferred as a base χDL when CLAIRify consistently fails to extract the parameters.

We found that human reviewers need to be careful with `vessel` parameters when multiple vessels are used in a procedure. When combining multiple candidate χDLs, reviewers also need to pay attention to the parameters. For example, the vessel names should be standardized to match the notation used in one of the χDLs, and the associated `Component` should be declared in the `Hardware` section. Similarly, the `Reagents` section should be updated when a `reagent` that used in χDL actions is not declared.

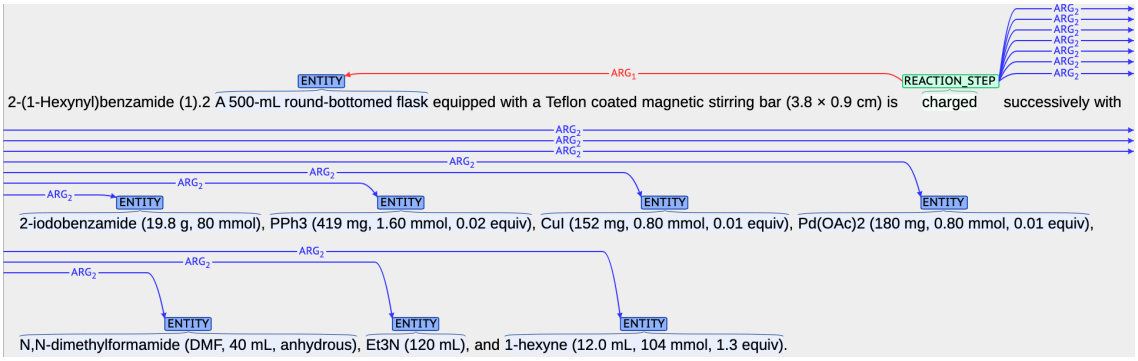
To supplement implicit actions, expertise in chemistry is required because it is necessary to determine when stirring is required and how long the stirring should continue. For example, it is difficult for non-experts to supplement implicit stirring actions after mixing compounds because the stirring time depends on the specific compounds involved.

5.4 Summary

To curate organic synthesis procedures as structured data, we focused on the limitations of the automated information extraction from the literature and proposed a framework for reviewing the results of the extraction. The proposed user interface can visually support human reviewers by highlighting original texts with the OSPAR format. In addition, to improve the quality of the automated conversion, a method to show the generated χDL by

our rule-based system and the GLLM-based CLAIRify was developed. In the experiment, we confirmed the comparative advantage of our approach by showing the outputs of both systems to the user. We also confirmed that our system obtained higher recall than SynthReader.

Text with the OSPAR annotation



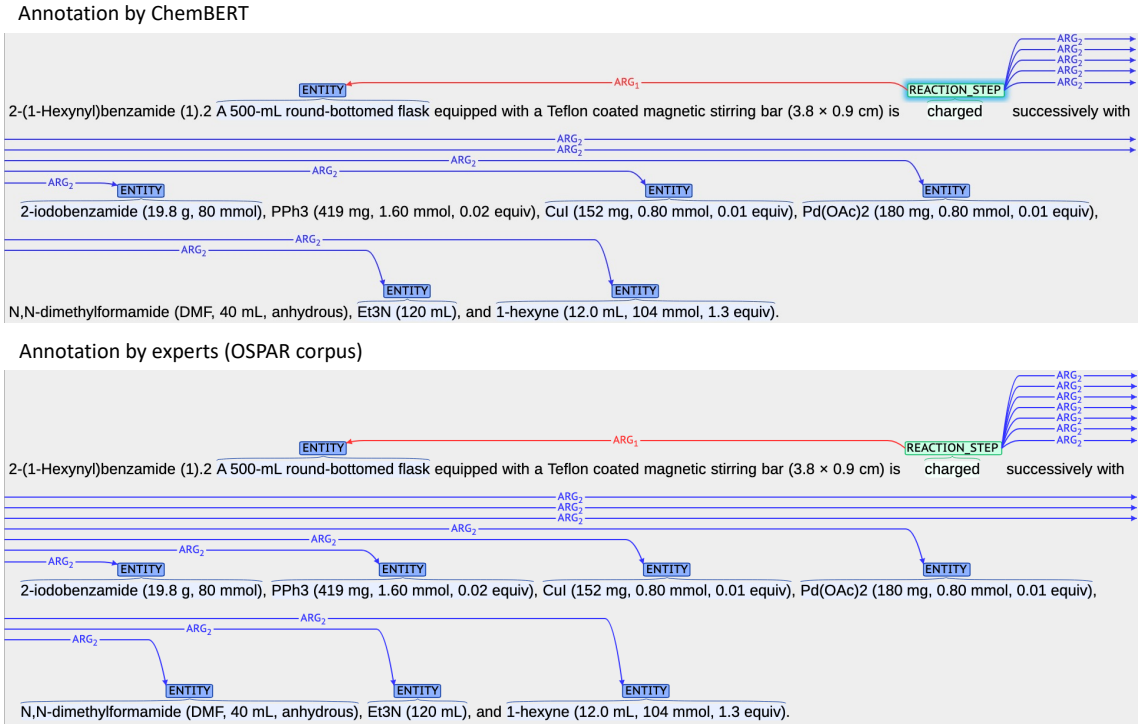
χDL by OSPAR2XDL

```
<Procedure>
 <Add  vessel="reactor"
       reagent="2-iodobenzamide"
       mass="19.8 g"
       amount="80 mmol" />
 <Add  vessel="reactor"
       reagent="PPh3"
       mass="0.419 g"
       amount="1.60 mmol" />
 <Add  vessel="reactor"
       reagent="CuI"
       mass="0.152 g"
       amount="0.80 mmol" />
 <Add  vessel="reactor"
       reagent="Pd(OAc)2"
       mass="0.18 g"
       amount="0.80 mmol" />
 <Add  vessel="reactor"
       reagent="N,N-dimethylformamide DMF"
       volume="40 mL" />
 <Add  vessel="reactor"
       reagent="Et3N"
       volume="120 mL" />
 <Add  vessel="reactor"
       reagent="1-hexyne"
       volume="12 mL"
       amount="104 mmol" />
 ...
</Procedure>
```

χDL by CLAIRify

```
<Procedure>
 <Add  vessel="round_bottom_flask_500mL"
       reagent="2-iodobenzamide"/>
 <Add  vessel="round_bottom_flask_500mL"
       reagent="PPh3"/>
 <Add  vessel="round_bottom_flask_500mL"
       reagent="CuI"/>
 <Add  vessel="round_bottom_flask_500mL"
       reagent="Pd(OAc)2"/>
 <Add  vessel="round_bottom_flask_500mL"
       reagent="DMF"/>
 <Add  vessel="round_bottom_flask_500mL"
       reagent="Et3N"/>
 <Add  vessel="round_bottom_flask_500mL"
       reagent="1-hexyne"/>
 ...
</Procedure>
```

Figure 5.7: An example for increased recall by systems other than CLAIRify when CLAIRify did not extract parameters
*The procedure text is based on Okaya et al. [69], with revisions made through preprocessing in the OSPAR corpus.



χDL from the annotation by experts

```
<Add  vessel="reactor"
    reagent="2-iodobenzamide"
    mass="19.8 g"
    amount="80 mmol" />
<Add  vessel="reactor"
    reagent="PPh3"
    mass="0.419 g"
    amount="1.60 mmol" />
<Add  vessel="reactor"
    reagent="CuI"
    mass="0.152 g"
    amount="0.80 mmol" />
<Add  vessel="reactor"
    reagent="Pd(OAc)2"
    mass="0.18 g"
    amount="0.80 mmol" />
<Add  vessel="reactor"
    reagent="N,N-dimethylformamide DMF"
    volume="40 mL" />
<Add  vessel="reactor"
    reagent="Et3N"
    volume="120 mL" />
<Add  vessel="reactor"
    reagent="1-hexyne"
    volume="12 mL"
    amount="104 mmol" />
```

Figure 5.8: Effect of modifying the annotation
*Red-colored text in χDL were not generated from the annotation by ChemBERT. The procedure text is based on Okaya et al.,[69] with revisions made through preprocessing in the OSPAR corpus.

Chapter 6

Conclusions and future works

6.1 Conclusions

To curate organic synthesis procedures as structured data, we focused on the limitations of the automated information extraction from the literature and proposed a framework for reviewing the results of the extraction. We proposed the OSPAR format as the annotation format and constructed the OSPAR corpus to train ChemBERT models for the automated annotation. The proposed user interface can visually support human reviewers by highlighting original texts with the OSPAR format. In addition, to improve the quality of the automated conversion, a method to show the generated χDL by our rule-based system and the GLLM-based CLAIRify was developed. In the experiment, we confirmed the comparative advantage of our approach by showing the outputs of both systems to the user.

6.2 Future works

Expansion of the OSPAR corpus As discussed in Section 4.3.3 and Section 5.3.3, the small size of the corpus posed several challenges for the annotation by ChemBERT. This issue can be resolved if the users of our framework annotate the procedures in the OSPAR format, as this will increase the amount of training data. Concurrently, the vocabulary of the rolesets should be extended.

Development of rule-based conversion Our system was unable to handle cases involving multiple vessels. Implementing anaphora resolution for compounds and vessels could be a promising solution to address this issue. Another challenge is handling implicit actions. We need to construct rules to capture implicit actions in future work. For example, inserting StartStir after multiple Add actions would be considered.

Prompt for CLAIRify

To improve the consistency of the output of CLAIRify, the description of χDL has room for improvement. For example, `mass`, `amount` or `volume` should be required to perform organic synthesis, however, the current version of the description of χDL did not mention it.

Acknowledgements

First and foremost, I would like to thank my supervisor, Prof. Masaharu Yoshioka for his guidance and advice. I am grateful for the opportunity to learn from him not only in terms of specific research skills but also regarding the essential qualities of a researcher. I would like to thank my collaborators, Assoc. Prof. Yuuya Nagata and Asst. Prof. Seiji Akiyama for their contributions and insightful discussions. I would also like to thank Prof. Hiroki Arimura and Prof. Atsuyoshi Nakamura for their comments on this dissertation as referees. I would also like to thank the past and present members of Knowledge base laboratory for their insightful discussions and the enjoyable environment.

I would like thank for the financial support by JST SPRING, Grant Number JPMJSP2119.

Finally, I would like to thank my family and friends for their kind support.

References

[1] Gary Tom, Stefan P Schmid, Sterling G Baird, Yang Cao, Kourosh Darvish, Han Hao, Stanley Lo, Sergio Pablo-García, Ella M Rajaonson, Marta Skreta, et al. Self-driving laboratories for chemistry and materials science. *Chemical Reviews*, 124(16):9633–9732, 2024.

[2] Elena M. Zamora and Paul E. Blower. Extraction of chemical reaction information from primary journal text using computational linguistics techniques. 1. Lexical and syntactic phases. *Journal of Chemical Information and Computer Sciences*, 24(3):176–181, 1984.

[3] Elena M. Zamora and Paul E. Blower. Extraction of chemical reaction information from primary journal text using computational linguistics techniques. 2. Semantic phase. *Journal of Chemical Information and Computer Sciences*, 24(3):181–188, 1984.

[4] Nick Kemp and Michael Lynch. Extraction of information from the text of chemical patents. 1. identification of specific chemical names. *Journal of Chemical Information and Computer Sciences*, 38(4):544–551, 1998.

[5] Lezan Hawizy, David M Jessop, Nico Adams, and Peter Murray-Rust. ChemicalTagger: A tool for semantic text-mining in chemistry. *Journal of Cheminformatics*, 3:1–13, 2011.

[6] Ling Luo, Zhihao Yang, Pei Yang, Yin Zhang, Lei Wang, Hongfei Lin, and Jian Wang. An attention-based BiLSTM-CRF approach to document-level chemical named entity recognition. *Bioinformatics*, 34(8):1381–1388, 2018.

[7] Peter Corbett and John Boyle. Chemlistem: chemical named entity recognition using recurrent neural networks. *Journal of cheminformatics*, 10(1):59, 2018.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[9] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[10] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[11] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

[12] Matthew C Swain and Jacqueline M Cole. ChemDataExtractor: A Toolkit for Automated Extraction of Chemical Information from the Scientific Literature. *Journal of Chemical Information and Modeling*, 56:1894–1904, 10 2016.

[13] Dat Quoc Nguyen, Zenan Zhai, Hiyori Yoshikawa, Biaoyan Fang, Christian Druckenbrodt, Camilo Thorne, Ralph Hoessel, Saber A. Akhondi, Trevor Cohn, Timothy Baldwin, and Karin Verspoor. ChEMU: Named Entity Recognition and Event Extraction of Chemical Reactions from Patents. In Joemon M. Jose, Emine Yilmaz, João Magalhães, Pablo Castells, Nicola Ferro, Mário J. Silva, and Flávio Martins, editors, *Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14-17, 2020, Proceedings, Part II*, volume 12036 of *Lecture Notes in Computer Science*, pages 572–579. Springer, 2020.

[14] Alain C Vaucher, Federico Zipoli, Joppe Geluykens, Vishnu H Nair, Philippe Schwaller, and Teodoro Laino. Automated extraction of chemical synthesis actions from experimental procedures. *Nature Communications*, 11:3601, 2020.

[15] Steven M Kearnes, Michael R Maser, Michael Wleklinski, Anton Kast, Abigail G Doyle, Spencer D Dreher, Joel M Hawkins, Klavs F Jensen, and Connor W Coley. The open reaction database. *Journal of the American Chemical Society*, 143(45):18820–18826, 2021.

[16] Kojiro Machi, Seiji Akiyama, Yuuya Nagata, and Masaharu Yoshioka. OSPAR: A Corpus for Extraction of Organic Synthesis Procedures with Argument Roles. *Journal of Chemical Information and Modeling*, 63(21):6619–6628, 2023.

[17] Connor W Coley, Dale A Thomas III, Justin AM Lummiss, Jonathan N Jaworski, Christopher P Breen, Victor Schultz, Travis Hart, Joshua S Fishman, Luke Rogers, Hanyu Gao, et al. A robotic platform for flow synthesis of organic compounds informed by ai planning. *Science*, 365(6453):eaax1566, 2019.

[18] S Hessam M Mehr, Matthew Craven, Artem I Leonov, Graham Keenan, and Leroy Cronin. A universal system for digitization and automatic execution of the chemical synthesis literature. *Science*, 370:101–108, 2020.

[19] Naruki Yoshikawa, Marta Skreta, Kourosh Darvish, Sebastian Arellano-Rubach, Zhi Ji, Lasse Bjørn Kristensen, Andrew Zou Li, Yuchi Zhao, Haoping Xu, Artur Ku-

ramshin, Alán Aspuru-Guzik, Florian Shkurti, and Animesh Garg. Large language models for chemistry robotics. *Auton. Robots*, 47(8):1057–1086, 2023.

[20] Kristine Laws, Marcus Tze-Kiat Ng, Abhishek Sharma, Yibin Jiang, Alexander JS Hammer, and Leroy Cronin. An Autonomous Electrochemical Discovery Robot that Utilises Probabilistic Algorithms: Probing the Redox Behaviour of Inorganic Materials. *ChemElectroChem*, 11(1):e202300532, 2024.

[21] *ChemIDE*. https://croningroup.gitlab.io/chemputer/xdlapp. (accessed September 23, 2024).

[22] Kojiro Machi and Masaharu Yoshioka. HUKB at chemu 2022 task 1: Expression-level information extraction. In Guglielmo Faggioli, Nicola Ferro, Allan Hanbury, and Martin Potthast, editors, *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, volume 3180 of *CEUR Workshop Proceedings*, pages 797–807. CEUR-WS.org, 2022.

[23] Kojiro Machi, Seiji Akiyama, Yuuya Nagata, and Masaharu Yoshioka. A framework for reviewing the results of automated conversion of structured organic synthesis procedures from the literature. *Digital Discovery*, 4:172–180, 2025.

[24] Mervyn Bregonje. Patents: A unique source for scientific technical information in chemistry related industry? *World Patent Information*, 27(4):309–315, 2005.

[25] Jiayuan He, Dat Quoc Nguyen, Saber A. Akhondi, Christian Druckenbrodt, Camilo Thorne, Ralph Hoessel, Zubair Afzal, Zenan Zhai, Biaoyan Fang, Hiyori Yoshikawa, Ameer Albahem, Jingqi Wang, Yuankai Ren, Zhi Zhang, Yaoyun Zhang, Mai Hoang Dao, Pedro Ruas, Andre Lamurias, Francisco M. Couto, Jenny Copara, Nona Naderi, Julien Knafou, Patrick Ruch, Douglas Teodoro, Daniel M. Lowe, John Mayfield, Abdullatif Köksal, Hilal Dönmez, Elif Özkirimli, Arzucan Özgür, Darshini Mahendran, Gabrielle Gurdin, Nastassja Lewinski, Christina Tang, Bridget T. McInnes, C. S. Malarkodi, Pattabhi Rk Rao, Sobha Lalitha Devi, Lawrence Cavedon, Trevor Cohn,

Timothy Baldwin, and Karin Verspoor. An Extended Overview of the CLEF 2020 ChEMU Lab: Information Extraction of Chemical Reactions from Patents. In Linda Cappellato, Carsten Eickhoff, Nicola Ferro, and Aurélie Névéol, editors, *Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25, 2020*, volume 2696 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2020.

[26] Yuan Li, Biaoyan Fang, Jiayuan He, Hiyori Yoshikawa, Saber A. Akhondi, Christian Druckenbrodt, Camilo Thorne, Zubair Afzal, Zenan Zhai, Timothy Baldwin, and Karin Verspoor. Extended Overview of ChEMU 2021: Reaction Reference Resolution and Anaphora Resolution in Chemical Patents. In Guglielmo Faggioli, Nicola Ferro, Alexis Joly, Maria Maistro, and Florina Piroi, editors, *Proceedings of the Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, Bucharest, Romania, September 21st - to - 24th, 2021*, volume 2936 of *CEUR Workshop Proceedings*, pages 693–709. CEUR-WS.org, 2021.

[27] Yuan Li, Biaoyan Fang, Jiayuan He, Hiyori Yoshikawa, Saber A. Akhondi, Christian Druckenbrodt, Camilo Thorne, Zubair Afzal, Zenan Zhai, Kojiro Machi, Masaharu Yoshioka, Youngrok Jang, Hosung Song, Junho Lee, Gyeonghun Kim, Yireun Kim, Stanley Jungkyu Choi, Honglak Lee, Kyunghoon Bae, Darshini Mahendran, Christina Tang, Bridget T. McInnes, Timothy Baldwin, and Karin Verspoor. Extended overview of chemu 2022 evaluation campaign: Information extraction in chemical patents. In Guglielmo Faggioli, Nicola Ferro, Allan Hanbury, and Martin Potthast, editors, *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, volume 3180 of *CEUR Workshop Proceedings*, pages 758–781. CEUR-WS.org, 2022.

[28] Martha Palmer, Daniel Gildea, and Paul Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106, 2005.

[29] Jiang Guo, A Santiago Ibanez-Lopez, Hanyu Gao, Victor Quach, Connor W Coley, Klavs F Jensen, and Regina Barzilay. Automated Chemical Reaction Extraction from

Scientific Literature. *Journal of Chemical Information and Modeling*, 62:2035–2045, 2021.

[30] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *CoRR*, abs/1609.08144, 2016.

[31] Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A pretrained language model for scientific text. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *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 3615–3620, Hong Kong, China, November 2019. Association for Computational Linguistics.

[32] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

[33] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23, 2021.

[34] Tanishq Gupta, Mohd Zaki, NM Anoop Krishnan, and Mausam. MatSciBERT: A materials domain language model for text mining and information extraction. *npj Computational Materials*, 8(1):102, 2022.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon,

U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[36] Jingqi Wang, Yuankai Ren, Zhi Zhang, and Yaoyun Zhang. Melaxtech: A report for CLEF 2020 - chemu task of chemical reaction extraction from patent. In Linda Cappellato, Carsten Eickhoff, Nicola Ferro, and Aurélie Névéol, editors, *Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25, 2020*, volume 2696 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2020.

[37] Ritam Dutt, Sopan Khosla, and Carolyn P. Rosé. A pipelined approach to Anaphora Resolution in Chemical Patents. In Guglielmo Faggioli, Nicola Ferro, Alexis Joly, Maria Maistro, and Florina Piroi, editors, *Proceedings of the Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, Bucharest, Romania, September 21st - to - 24th, 2021*, volume 2936 of *CEUR Workshop Proceedings*, pages 710–719. CEUR-WS.org, 2021.

[38] https://github.com/spyysalo/standoff2conll. (accessed May 27, 2020).

[39] Zexuan Zhong and Danqi Chen. A Frustratingly Easy Approach for Entity and Relation Extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 50–61, Online, June 2021. Association for Computational Linguistics.

[40] Eric Wallace, Jens Tuyls, Junlin Wang, Sanjay Subramanian, Matt Gardner, and Sameer Singh. AllenNLP Interpret: A Framework for Explaining Predictions of NLP models. In Sebastian Padó and Ruihong Huang, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019 - System Demonstrations*, pages 7–12. Association for Computational Linguistics, 2019.

[41] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[42] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[43] Michihiro Yasunaga, Jure Leskovec, and Percy Liang. LinkBERT: Pretraining language models with document links. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8003–8016, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[44] Reaxys. https://www.reaxys.com. (accessed June 20, 2023).

[45] Scifinder^{n}. https://scifinder-n.cas.org. (accessed June 20, 2023).

[46] Jin-Dong Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. GENIA corpus - a semantically annotated corpus for bio-textmining. In *Proceedings of the Eleventh International Conference on Intelligent Systems for Molecular Biology, June 29 - July 3, 2003, Brisbane, Australia*, pages 180–182, 2003.

[47] Peter Corbett, Colin Batchelor, and Simone Teufel. Annotation of Chemical Named Entities. In *Biological, translational, and clinical language processing*, pages 57–64, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[48] Saber A. Akhondi, Alexander G. Klenner, Christian Tyrchan, Anil K. Manchala, Kiran Boppana, Daniel Lowe, Marc Zimmermann, Sarma A.R.P. Jagarlapudi, Roger

Sayle, Jan A. Kors, and Sorel Muresan. Annotated chemical patent corpus: A gold standard for text mining. *PLoS ONE*, 9:1–8, 2014.

[49] Martin Krallinger, Obdulia Rabal, Florian Leitner, Miguel Vazquez, David Salgado, Zhiyong Lu, Robert Leaman, Yanan Lu, Donghong Ji, Daniel M. Lowe, Roger A. Sayle, Riza Theresa Batista-Navarro, Rafal Rak, Torsten Huber, Tim Rocktäschel, Sérgio Matos, David Campos, Buzhou Tang, Hua Xu, Tsendsuren Munkhdalai, Keun Ho Ryu, S. V. Ramanan, Senthil Nathan, Slavko Žitnik, Marko Bajec, Lutz Weber, Matthias Irmer, Saber A. Akhondi, Jan A. Kors, Shuo Xu, Xin An, Utpal Kumar Sikdar, Asif Ekbal, Masaharu Yoshioka, Thaer M. Dieb, Miji Choi, Karin Verspoor, Madian Khabsa, C. Lee Giles, Hongfang Liu, Komandur Elayavilli Ravikumar, Andre Lamurias, Francisco M. Couto, Hong-Jie Dai, Richard Tzong-Han Tsai, Caglar Ata, Tolga Can, Anabel Usié, Rui Alves, Isabel Segura-Bedmar, Paloma Martínez, Julen Oyarzabal, and Alfonso Valencia. The CHEMDNER corpus of chemicals and drugs and its annotation principles. *Journal of Cheminformatics*, 7(1):S2, Jan 2015.

[50] Saber A Akhondi, Hinnerk Rey, Markus Schwörer, Michael Maier, John P Toomey, Heike Nau, Gabriele Ilchmann, Mark Sheehan, Matthias Irmer, Claudia Bobach, Marius A Doornenbal, Michelle Gregory, and Jan A Kors. Automatic identification of relevant chemical compounds from patents. *Database J. Biol. Databases Curation*, 2019:baz001, 2019.

[51] Rezarta Islamaj, Robert Leaman, Sun Kim, Dongseop Kwon, Chih-Hsuan Wei, Donald C. Comeau, Yifan Peng, David Cissel, Cathleen Coss, Carol Fisher, Rob Guzman, Preeti Gokal Kochar, Stella Koppel, Dorothy Trinh, Keiko Sekiya, Janice Ward, Deborah Whitman, Susan Schmidt, and Zhiyong Lu. NLM-Chem, a new resource for chemical entity recognition in PubMed full text literature. *Scientific Data*, 8(1):91, Mar 2021.

[52] Yue Zhang, Cong Wang, Mya Soukaseum, Dionisios G. Vlachos, and Hui Fang. Unleashing the Power of Knowledge Extraction from Scientific Literature in Cataly-

sis. *Journal of Chemical Information and Modeling*, 62(14):3316–3330, 2022. PMID: 35772028.

[53] Sébastien R Goudreau, David Marcoux, and André B Charette. Synthesis of dimethyl 2-phenylcyclopropane-1, 1-dicarboxylate using an iodonium ylide derived from dimethyl malonate. *Organic Syntheses*, 87:115–125, 2003.

[54] Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France, April 2012. Association for Computational Linguistics.

[55] Shun Okaya, Keiichiro Okuyama, Kentaro Okano, and Hidetoshi Tokuyama. Trichloroboron-promoted deprotection of phenolic benzyl ether using pentamethylbenzene as a non lewis-basic cation scavenger. *Organic Syntheses*, 93:63–74, 2016.

[56] Biaoyan Fang, Christian Druckenbrodt, Saber A Akhondi, Jiayuan He, Timothy Baldwin, and Karin Verspoor. ChEMU-Ref: A Corpus for Modeling Anaphora Resolution in the Chemical Domain. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1362–1375. Association for Computational Linguistics, 4 2021.

[57] *Organic Syntheses*. http://www.orgsyn.org. (accessed October 14, 2021).

[58] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[59] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[60] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child,

Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[61] Liang Yao, Chengsheng Mao, and Yuan Luo. Kg-bert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*, 2019.

[62] Sheshera Mysore, Edward Kim, Emma Strubell, Ao Liu, Haw-Shiuan Chang, Srikrishna Kompella, Kevin Huang, Andrew McCallum, and Elsa Olivetti. Automatically Extracting Action Graphs from Materials Science Synthesis Procedures. *arXiv*, 11 2017.

[63] Sheshera Mysore, Zach Jensen, Edward Kim, Kevin Huang, Haw-Shiuan Chang, Emma Strubell, Jeffrey Flanigan, Andrew Mccallum, and Elsa Olivetti. 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, 2019.

[64] Fusataka Kuniyoshi, Kohei Makino, Jun Ozawa, and Makoto Miwa. 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, 2020.

[65] Tim O'Gorman, Zach Jensen, Sheshera Mysore, Rubayyat Mahbub, Kevin Huang, Elsa Olivetti, and Andrew McCallum. MS-MENTIONS: Consistently Annotating Entity Mentions in Materials Science Procedural Text. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1337–1352, 2021.

[66] Zheren Wang, Kevin Cruse, Yuxing Fei, Ann Chia, Yan Zeng, Haoyan Huo, Tanjin He, Bowen Deng, Olga Kononova, and Gerbrand Ceder. ULSA: unified language

of synthesis actions for the representation of inorganic synthesis protocols. *Digital Discovery*, 1:313–324, 2022.

[67] Kohei Makino, Fusataka Kuniyoshi, Jun Ozawa, and Makoto Miwa. Extracting and analyzing inorganic material synthesis procedures in the literature. *IEEE Access*, 10:31524–31537, 2022.

[68] Daniel Jurafsky and James H. Martin. *Speech and Language Processing (2Nd Edition)*, chapter 21 Computational Discourse, pages 693–736. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2009.

[69] Shun Okaya, Keiichiro Okuyama, Kentaro Okano, and Hidetoshi Tokuyama*. Trichloroboron-promoted Deprotection of Phenolic Benzyl Ether Using Pentamethylbenzene as a Non Lewis-Basic Cation Scavenger. *Organic Syntheses*, 93:63–74, 2003.

[70] *Flask*. https://flask.palletsprojects.com/en/3.0.x/. (accessed June 21, 2024).

[71] *CodeMirror 5*. https://codemirror.net/5. (accessed June 21, 2024).

[72] *XDL*. https://gitlab.com/croningroup/chemputer/xdl. (accessed May 9, 2024).

[73] *text2num*. https://github.com/allo-media/text2num. (accessed September 23, 2024).

[74] CLAIRify. https://github.com/ac-rad/xdl-generation. (accessed March 18, 2024).

[75] Cir. https://cactus.nci.nih.gov/chemical/structure. (accessed October 14, 2021).

[76] Cirpy. https://github.com/mcs07/CIRpy. (accessed October 14, 2021).

Appendix A

Article selection

This section describes the details of article selection in Section "Data curation and preprocessing." First, we collected articles from annual volumes of *Organic Syntheses* [57] 81 (2005) to 97 (2020). Next, we selected articles that had a "Procedure" section, and we used only the procedure of the first reaction of each article for the corpus. In addition, we excluded procedures that have multiple boundaries of synthesis and work-up.

As we internally constructed a corpus for the extraction of chemical entities in the past, we used the same articles in this work. Most articles in *Organic Syntheses* [57] have the annotation of chemical names with CAS (Chemical Abstructs Service) number. In the previous work, we limited the documents whose all chemical names were aligned with the entries in Reaxys [44] for the preannotation of the corpus. We used Chemical Identifier Resolver (CIR) [75] via CIRpy [76] to identify the CAS number of chemical names if there was no CAS number in the article of *Organic Syntheses*.

Finally, we obtained 112 procedures as shown in Table A.1.

Org. Synth. **2005**, 81, 1	*Org. Synth.* **2009**, 86, 262	*Org. Synth.* **2016**, 93, 263
Org. Synth. **2005**, 81, 14	*Org. Synth.* **2009**, 86, 298	*Org. Synth.* **2016**, 93, 272
Org. Synth. **2005**, 81, 26	*Org. Synth.* **2009**, 86, 308	*Org. Synth.* **2016**, 93, 413
Org. Synth. **2005**, 81, 63	*Org. Synth.* **2009**, 86, 315	*Org. Synth.* **2017**, 94, 16
Org. Synth. **2005**, 81, 244	*Org. Synth.* **2009**, 86, 325	*Org. Synth.* **2018**, 95, 97
Org. Synth. **2005**, 82, 1	*Org. Synth.* **2010**, 87, 16	*Org. Synth.* **2018**, 95, 112
Org. Synth. **2005**, 82, 30	*Org. Synth.* **2010**, 87, 126	*Org. Synth.* **2018**, 95, 276
Org. Synth. **2005**, 82, 64	*Org. Synth.* **2010**, 87, 192	*Org. Synth.* **2018**, 95, 310
Org. Synth. **2005**, 82, 93	*Org. Synth.* **2010**, 87, 218	*Org. Synth.* **2018**, 95, 357
Org. Synth. **2005**, 82, 99	*Org. Synth.* **2010**, 87, 310	*Org. Synth.* **2018**, 95, 425
Org. Synth. **2005**, 82, 179	*Org. Synth.* **2011**, 88, 398	*Org. Synth.* **2019**, 96, 36
Org. Synth. **2006**, 83, 18	*Org. Synth.* **2013**, 90, 62	*Org. Synth.* **2019**, 96, 418
Org. Synth. **2006**, 83, 31	*Org. Synth.* **2013**, 90, 164	*Org. Synth.* **2019**, 96, 436
Org. Synth. **2006**, 83, 49	*Org. Synth.* **2013**, 90, 229	*Org. Synth.* **2019**, 96, 455
Org. Synth. **2006**, 83, 61	*Org. Synth.* **2013**, 90, 240	*Org. Synth.* **2020**, 97, 12
Org. Synth. **2006**, 83, 70	*Org. Synth.* **2013**, 90, 287	*Org. Synth.* **2005**, 81, 157
Org. Synth. **2006**, 83, 97	*Org. Synth.* **2013**, 90, 306	*Org. Synth.* **2005**, 82, 134
Org. Synth. **2006**, 83, 103	*Org. Synth.* **2013**, 90, 327	*Org. Synth.* **2008**, 85, 138
Org. Synth. **2006**, 83, 111	*Org. Synth.* **2013**, 90, 338	*Org. Synth.* **2008**, 85, 189
Org. Synth. **2006**, 83, 155	*Org. Synth.* **2014**, 91, 39	*Org. Synth.* **2009**, 86, 92
Org. Synth. **2006**, 83, 193	*Org. Synth.* **2014**, 91, 93	*Org. Synth.* **2009**, 86, 344
Org. Synth. **2007**, 84, 22	*Org. Synth.* **2014**, 91, 106	*Org. Synth.* **2012**, 89, 230
Org. Synth. **2007**, 84, 32	*Org. Synth.* **2014**, 91, 116	*Org. Synth.* **2016**, 93, 163
Org. Synth. **2007**, 84, 43	*Org. Synth.* **2015**, 92, 1	*Org. Synth.* **2016**, 93, 352
Org. Synth. **2007**, 84, 68	*Org. Synth.* **2015**, 92, 13	*Org. Synth.* **2018**, 95, 486
Org. Synth. **2007**, 84, 88	*Org. Synth.* **2015**, 92, 117	*Org. Synth.* **2020**, 97, 139
Org. Synth. **2007**, 84, 262	*Org. Synth.* **2015**, 92, 182	*Org. Synth.* **2005**, 82, 157
Org. Synth. **2007**, 84, 272	*Org. Synth.* **2015**, 92, 195	*Org. Synth.* **2005**, 82, 170
Org. Synth. **2007**, 84, 295	*Org. Synth.* **2015**, 92, 247	*Org. Synth.* **2010**, 87, 115
Org. Synth. **2007**, 84, 325	*Org. Synth.* **2015**, 92, 267	*Org. Synth.* **2014**, 91, 27
Org. Synth. **2007**, 84, 347	*Org. Synth.* **2015**, 92, 296	*Org. Synth.* **2014**, 91, 60
Org. Synth. **2007**, 84, 359	*Org. Synth.* **2015**, 92, 356	*Org. Synth.* **2015**, 92, 171
Org. Synth. **2008**, 85, 34	*Org. Synth.* **2016**, 93, 75	*Org. Synth.* **2015**, 92, 227
Org. Synth. **2008**, 85, 172	*Org. Synth.* **2016**, 93, 100	*Org. Synth.* **2015**, 92, 328
Org. Synth. **2009**, 86, 11	*Org. Synth.* **2016**, 93, 115	*Org. Synth.* **2016**, 93, 63
Org. Synth. **2009**, 86, 18	*Org. Synth.* **2016**, 93, 178	*Org. Synth.* **2018**, 95, 127
Org. Synth. **2009**, 86, 36	*Org. Synth.* **2016**, 93, 245	*Org. Synth.* **2020**, 97, 54
Org. Synth. **2009**, 86, 252		

Table A.1: Document list of the OSPAR