Supplementary Information for "A Framework for Reviewing the Results of Automated Conversion of Structured Organic Synthesis Procedures from the Literature"

Kojiro Machi[1], Seiji Akiyama[2], Yuuya Nagata[2], and Masaharu Yoshioka[1,2,3]

[1]Graduate School of Information Science and Technology, Hokkaido University, Kita 14 Nishi 9, Kita-ku, Sapporo, Hokkaido, 060-0814, Japan
[2]Institute for Chemical Reaction Design and Discovery (WPI-ICReDD), Hokkaido University, Kita 21 Nishi 10, Kita-ku, Sapporo, Hokkaido 001-0021, Japan
[3]Faculty of Information Science and Technology, Hokkaido University, Kita 14 Nishi 9, Kita-ku, Sapporo, Hokkaido, 060-0814, Japan

1 How to run our framework

Our framework was validated in an environment with Python 3.8 on Ubuntu 24.04.

1.1 Installation

Our framework is available at https://github.com/mlmachi/OSPAR_XDL with a detailed installation guide.

1.2 Configuration and execution

Before starting the user interface, the user need to set `config.json`. The description of the arguments are shown in Table S1.

Table S1: Description of the arguments in `config.json`

argument	description
brat_dir	Where to save generated OSPAR annotation from text.
brat_url	URL of brat server.
brat_working_dir	URL of brat server with the name of working directory
chembert_config_file	Configs of ChemBERT models.
use_clairify	Whether to use CLAIRify.
GPT_model	The model name of GPT.
OpenAI_API_KEY	API key for OpenAI API
clairify_interval_sec	The interval for accessing the OpenAI API (second)

2 Details of user interface

The proposed user interface was implemented using Flask,[1] a Python-based web framework. We used brat[2] 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.[3]

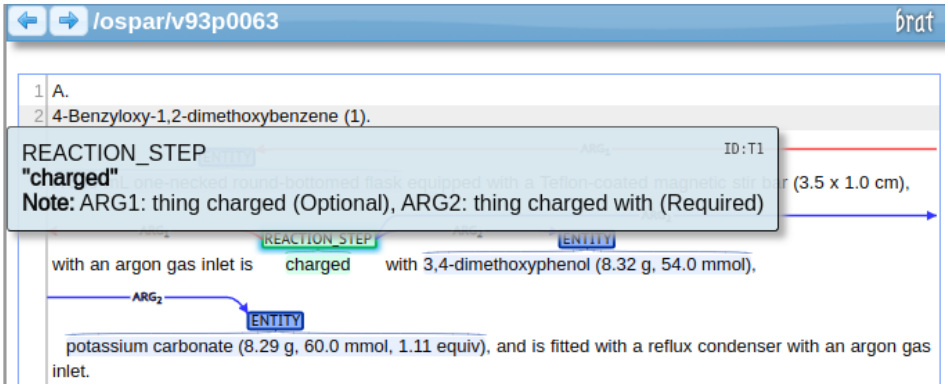


Figure S1: A screen shot of brat. When the user moves a cursor over REACTION_STEP, a roleset for the action is displayed.

3 OSPAR2χDL

This section describes the details of OSPAR2χDL that are not covered in the original paper.

3.1 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 S2 shows χDL actions that can be generated by OSPAR2χDL, along with the corresponding part of the OSPAR rolesets.

Table S2: 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

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 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.

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 S2 shows the rules for converting ARGM into χDL action(s). `MODIFIER` and `TEMPERATURE` 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

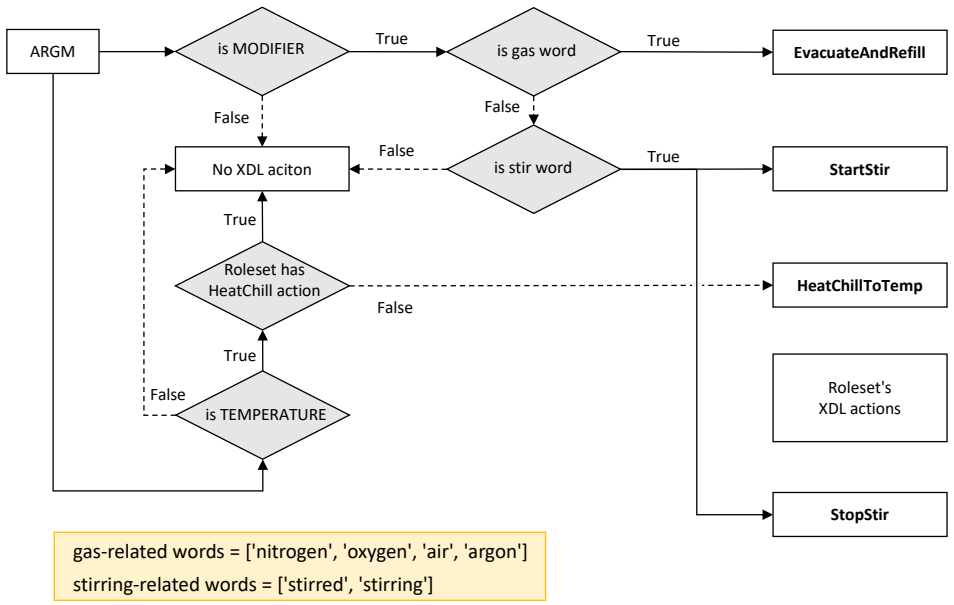


Figure S2: Rules for the converting ARGM into χDL action(s).

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.

3.2 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>` tag in ChemicalTagger.

3.3 Dictionary for interpreting parameters

We created a dictionary to interpret words that represent parameters for TEMPERATURE, TIME, and stirring rates in MODIFIER.

```
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'
}
```

3.4 Effect of modifying the annotation

Figure S3 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.

Table S3: χDL actions that used in this work and their categories. The actions enclosed in parentheses did not appear in the evaluation data.

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

Table S4: Result of explicit actions for each category. The numbers indicate (#found action)/(#all actions). SR is SynthReader, Pipe is Pipeline, O2X is OSPAR2χDL and CLAIR is CLAIRify

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

References

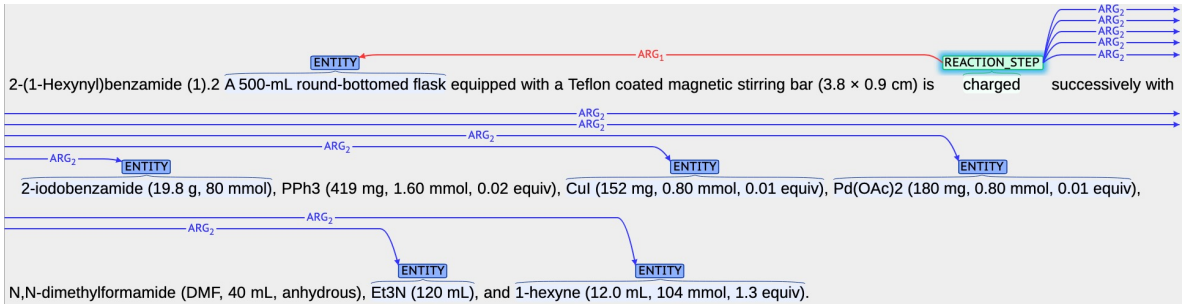
[1] *Flask*, https://flask.palletsprojects.com/en/3.0.x/, (accessed June 21, 2024).

[2] P. Stenetorp, S. Pyysalo, G. Topić, T. Ohta, S. Ananiadou and J. Tsujii, Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics, Avignon, France, 2012, pp. 102–107.

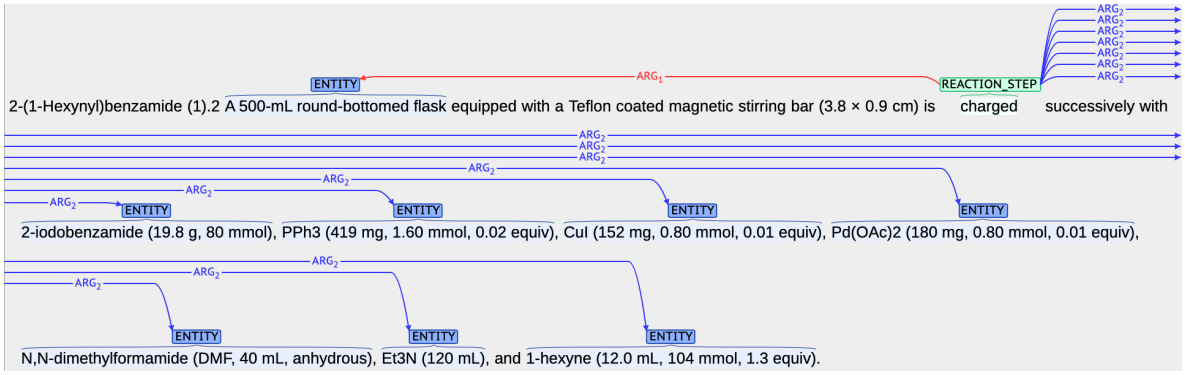
[3] *CodeMirror 5*, https://codemirror.net/5, (accessed June 21, 2024).

[4] S. Okaya, K. Okuyama, K. Okano and H. Tokuyama*, *Organic Syntheses*, 2003, **93**, 63–74.

Annotation by ChemBERT



Annotation by experts (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 S3: Effect of modifying the annotation. Here, red-colored texts in χDL were not generated from the annotation by ChemBERT. The procedure text is based on Okaya et al.,[4] with revisions made through pre-processing in the OSPAR corpus.