Abstract

Machine learning has the potential to provide tremendous value to the life sciences by providing models that aid in the discovery of new molecules and reduce the time for new products to come to market. Chemical reactions play a significant role in these fields, but there is a lack of high-quality open-source chemical reaction datasets for training ML models. Herein, we present ORDerly, an open-source Python package for customizable and reproducible preparation of reaction data stored in accordance with the increasingly popular Open Reaction Database (ORD) schema. We use ORDerly to clean US patent data stored in ORD and generate datasets for forward prediction, retrosynthesis, as well as the first benchmark for reaction condition prediction. We train neural networks on datasets generated with ORDerly for condition prediction and show that datasets missing key cleaning steps can lead to silently overinflated performance metrics. Additionally, we train transformers for forward and retrosynthesis prediction and demonstrate how non-patent data can be used to evaluate model generalisation. By providing a customizable open-source solution for cleaning and preparing large chemical reaction data, ORDerly is poised to push forward the boundaries of machine learning applications in chemistry.

1 Introduction

Advancements in chemistry and material science hinge on the availability of high-quality chemical reaction data, and the advent of machine learning (ML) for science has highlighted the value that data can bring to chemistry. One important application is in the pharmaceutical industry, where figuring out how to make novel molecules remains a significant bottleneck, causing delays in the "design, make, test" cycle. Making a molecule (product) includes predicting the reaction
pathway (retrosynthesis) and suitable reaction conditions (e.g. solvents and reagents), and optimising for one or more outcomes such as reaction yield, selectivity, and conversion. ML is well suited to assist with these tasks, with a range of tools being developed for forward reaction prediction \[2\] \[3\] \[4\], retrosynthesis \[5\] \[6\] \[7\] \[8\] \[9\], condition prediction \[10\] \[11\] \[12\], yield prediction \[13\] \[14\] \[15\], and closed-loop optimisation \[16\] \[17\] \[18\].

Building reaction prediction tools requires access to large datasets for training. Historically, researchers have accessed proprietary in-house datasets or acquired the data through commercial databases such as Reaxys \[19\]. The advantage of commercial databases is both the scale of the datasets available (often millions of reactions) and the annotation already completed by the publishers. Yet, these datasets are not freely available to ML practitioners, stymieing advances in reaction condition prediction in both academia and industry.

Recently, efforts have been made to create openly-accessible databases for chemical reaction data. In particular, the Open Reaction Database (ORD) \[20\] is promising due to its exhaustive schema for describing chemical reaction data and breadth of data already incorporated. Yet, many of the datasets in ORD (license: Creative Commons Attribution Share Alike 4.0 International) require further processing before they can be used in ML pipelines, preventing practical use. This is especially true for the largest dataset in ORD extracted from the US patent literature (the "USPTO dataset" \[21\]). In this work, we endeavor to close this gap.

Herein, we present ORDerly, a new framework for extracting and cleaning data from ORD, accompanied by datasets for three reaction related tasks: retrosynthesis, forward, and condition prediction. By offering an open-source and customizable solution for cleaning chemical reaction data, ORDerly aims to contribute to the development of advanced ML models in chemistry and material science.

The remainder of the paper proceeds as follows. In section 2 we present a formulation of the key reaction prediction tasks considered in this work. This is followed by a brief review of related work in section 3. We then discuss the data extraction and cleaning methodology, and how ORDerly was used for dataset generation, in section 4. This is followed by experimental validation of these datasets with neural network and transformer architectures in section 5, demonstrating that missing key cleaning steps results in a dataset with contamination, which can inflate key performance metrics. We finally discuss the technical limitations of ORDerly in section 6 and then present our conclusions.

2 Problem formulation

As noted by Meng et al. \[22\], reaction related tasks operate on molecules. There are numerous machine readable molecular representations \[23\], including molecular graphs and strings, and in this work molecules are represented as SMILES strings. Each character $m_i$ in a SMILES string represents an atom or a molecular feature (bond, branch, ring closure): $M := m_1, m_2, m_3, \ldots, m_L$, where $L$ is the total number of characters in the string. Molecules can take on one of three roles in a reaction: reactant, product, or agent. A reaction $R$ transforms $N$ reactant molecules (sometimes called educts)
\{M_i^F\}_{i=1}^N \text{ by breaking and forming bonds to form } M \text{ new product molecules } \{M_i^P\}_{i=1}^M \text{ using } K \text{ agent molecules } \{M_i^A\}_{i=1}^K. \text{ Agents are helper molecules that enable the reaction to proceed (e.g., solvents, catalysts).}

\[ R: \{M_i^F\}_{i=1}^N, \{M_i^A\}_{i=1}^K \rightarrow \{M_i^P\}_{i=1}^M, \{M_i^A\}_{i=1}^K \]  

1

Given this view of reactions, we define three different reaction related tasks in this work.

**Forward prediction** is the task of predicting the product of a reaction \( M^P \) given its reactants \( \{M_i^F\}_{i=1}^N \), and potentially agents \( \{M_i^A\}_{i=1}^K \). Probabilistically, the task is to predict the distribution \( p(M^P|\{M_i^F\}_{i=1}^N) \). While experimental evaluation in a wet lab requires expert chemists and is a time intense task, reaction outcome prediction can help as a tool to evaluate the quality of a predicted retrosynthetic route (i.e., the probability that the reaction predicted by the single-step retrosynthesis model leads to the desired product) [24].

**Retrosynthesis** is the task of designing a sequence of \( Z \) reactions \( R_1, R_2, R_3, \ldots, R_Z \) that transform a set of readily available reactant molecules \( \{M_i^F\}_{i=1}^N \) to a desired product(s) \( \{M_i^P\}_{i=1}^M \). Retrosynthesis is done in the reverse direction by starting with the desired product(s) \( \{M_i^P\}_{i=1}^M \) and predicting reactants \( \{M_i^F\}_{i=1}^N \) that would react to form the desired product(s). The predicted reactants \( \{M_i^F\}_{i=1}^N \) then become the products of the next reaction to be predicted \( \{M_i^P\}_{i=1}^M \). This process is repeated until a readily available set of starting reactant molecules are predicted \( \{M_i^F\}_{i=1}^N \). Therefore, the key machine learning task, often called single-step retrosynthesis, is predicting the distribution \( p(\{M_i^F\}_{i=1}^N|\{M_i^P\}_{i=1}^M) \) or the set of reactants that could lead to a given product(s) \( \{M_i^P\}_{i=1}^M \). Single-step retrosynthesis can be seen as the inverse of forward prediction.

**Condition prediction** is the task of predicting the distribution \( p(\{M_i^F\}_{i=1}^N|\{M_i^P\}_{i=1}^M, M^P) \) (i.e., the agents for a reaction given reactants and product). In addition to agents, some models can predict continuous variables such as reaction temperature and concentrations of reactants and agents [10].

3 Related work

3.1 Chemical reaction cleaning tools

Existing tools for cleaning reaction data are primarily targeted at retrosynthesis and forward prediction tasks [23] [26] [22] [28] and have somewhat limited extensibility, given that they are built to take as inputs CSV files or the stationary XML files of the US patent (USPTO) dataset [21] instead of the outputs of continuously updated databases such as ORD [20]. Furthermore, in the original publications, there is little to no discussion of how decisions made during cleaning (e.g. restricting the number of components in a reaction or the minimum frequency of occurrence) impact the datasets being cleaned or performance of models trained on the datasets. We believe that this is in part due to data cleaning historically being viewed as a “low value” task, and therefore not adequately discussed and published on.

USPTO, being the largest open-source chemical reaction dataset, has been cleaned a number of times for different learning tasks. For example, the USPTO-50K [29] [30] and USPTO-MIT datasets [31] are commonly used for benchmarking single-step retrosynthesis and forward predictions model[1] and these benchmarks are available in aggregate benchmarking sets such as the Therapeutics Data Commons (TDC) [32]. However, the code used to process the raw data to generate the aforementioned USPTO benchmarks was not published and, there is no publicly available benchmark for reaction condition prediction extracted from these datasets.

3.2 Forward prediction and single-step retrosynthesis models

Forward prediction and single-step retrosynthesis models both need to predict how bonds might be broken and formed to produce new molecules. A common approach is to enumerate a set of templates for bond changes that happen in particular classes of reactions and use a classifier to predict the most likely template given a set of molecules [3] [33] [34] [35] [36]. Alternatively, some models have been

---

[1] We discuss the difference between these datasets and our dataset in Appendix B.3.2
designed to explicitly predict bond changes \cite{31,37}. One promising approach is to directly predict the SMILES strings of the reactants (single-step retrosynthesis) or products (forward prediction) using a natural language processing model such as a transformer \cite{2,3,7,11}. In this work we use the transformer architecture of Schwaller et al. \cite{2}.

3.3 Condition prediction models

Many reaction condition prediction models have focused on indirect prediction of conditions by learning to predict a measure of reaction performance such as yield \cite{38,39,40}. The downside of these models are that they are usually only scoped to a single class of reactions due to the lack of reliable yield measurements in benchmarks based on the USPTO dataset \cite{41}. However, Gao et al. \cite{10} built a model for reaction condition prediction agnostic of reaction class for sequential prediction of agents and temperature using approximately ten million reactions mined from a closed-source dataset, Reaxys \cite{19}. We train this model with minor modifications on our new open-source condition prediction benchmark.

4 Dataset generation

ORDerly extracts data directly from ORD \cite{20}. Even though the data in ORD is stored in accordance with a structured schema, we found that further effort is required to transform the labeled data into ML-ready datasets. Therefore, ORDerly is centered around a data extraction script and a data cleaning script, both of which take numerous arguments that customize the operations being performed.

4.1 Extraction and cleaning methodology

The extraction script allows the user to choose whether reaction roles should be assigned using the labeling in ORD or using chemically-informed logic on the atom-mapped reaction string (if available). It also enables specification of data source (e.g., USPTO or non-USPTO), allowing users to train models with data from one source and test the performance with data from another source. Creating test sets from different data sources is a robust way to evaluate generalization performance.

We chose cleaning operations motivated by first-principles understanding of chemistry. Cleaning operations on the chemical reaction data include: (1) Restricting the number of reactants and product, preventing multi-step reactions being included in the dataset; (2) Ensuring that all molecules can be sanitized by the cheminformatics package RDKit \cite{42}; (3) Restricting the maximum number of unique catalysts, solvents, and reagents in a reaction based on commonly used experimental amounts; (4) Frequency filtering to remove outliers; (5) Sanity checking the yield (0% ≤ yield ≤ 100%), temperature, and pressure; (6) Removing duplicates, and finally; (7) Applying a random split to create training/validation/test sets, carefully ensuring that any inputs present in the train set (i.e. reactants and products for reaction condition prediction) are not also present in the test set.

Computational details: All extraction/cleaning operations described in this section were performed using a 2022 Mac Studio with an Apple M1 Max chip and 32GB memory. In ORD there are roughly 1.7 million reactions from US patents (USPTO) and 91k reactions that are not from US patents. During handling of the USPTO data in ORD we found that extracting and sanitizing the reaction components using the ORD labeling of components was slightly faster than using our custom logic applied to the reaction string, taking 28 minutes and 48 minutes, respectively. The cleaning steps took 6-8 minutes. Due to the amount of non-patent data being much less, extraction and cleaning of non-USPTO data took only a few minutes.

4.2 Reaction role assignment

We experimented with two approaches to assigning roles to the molecules found in a reaction (e.g., whether a molecule is a reactant or an agent): trusting the labeling of molecules in ORD (referred to as "labeling") or applying chemical reaction logic to identify the role of different molecules from the reaction string (referred to as "rxn string" or "reaction string"). Our reaction logic identified reactants (molecules that contribute atoms to the product(s)) and spectator molecules (molecules that do not contribute atoms to the product(s)) based on the atom-mapping and their position in the reaction SMILES string. Solvents were identified in the list of spectator molecules by cross checking

https://doi.org/10.26434/chemrxiv-2023-qkjtb-v2 ORCID: https://orcid.org/0000-0002-0494-643X Content not peer-reviewed by ChemRxiv. License: CC BY 4.0
4.3 Frequency filtering

Removing rare molecules can increase the signal to noise ratio in a dataset by removing outliers. In this work, we investigated two different strategies for filtering spectator molecules based on their frequency: deleting reactions with rare spectator molecules (rare→delete rxn) or keeping the reactions but mapping the rare molecules to an "other" category (rare→"other") (see Figure 2). We conducted experiments with both the rare→delete rxn and rare→"other" strategies for the task of condition prediction. The frequency threshold was set at 100 in line with previous research [10], though the sensitivity of dataset size to frequency threshold was still investigated (see Appendix D.2). Deleting reactions with rare molecules may create a more cohesive dataset by removing outliers, while renaming rare molecules "other" allows more reactions to be kept, offering more training data for the model.

4.4 Dataset composition

Datasets generated with ORDerly have the following column groups:

- Reaction SMILES (string), is_mapped (bool)
- Reactants & products (SMILES strings)
- Solvents and agents (rxn string data), or solvents, catalysts, and reagents (labeling data) (SMILES strings)
- Temperature, reaction time, yield (floats)
- Procedure details (string)
- Grant date (datetime), date of experiment (datetime), file name (string)

A CSV file is also created to keep track of the frequency of non-SMILES names used to represent molecules, so the most common molecule names could be added to the manual name resolution dictionary.

We used ORDerly to create datasets for three tasks: forward, retrosynthesis, and condition prediction. Several different datasets were created for each task, and the impact of each cleaning step on the dataset size can be found in Table 1. The datasets are freely available and can be downloaded immediately from FigShare or regenerated using the code in the ORDerly Github repository.
Table 1: Number of reactions left in each dataset after cleaning. A description of each dataset can be found in section 4. Note that the actual number of reactions used for training will differ from the dataset size shown below due to train/test splits and augmentation. Non-USPTO-retro had a final dataset size of 20,830 and was cleaned in the same way as ORD-ly-retro.

<table>
<thead>
<tr>
<th>Dataset name:</th>
<th>ORDerly-condition (labeling)</th>
<th>ORDerly-condition (rxn string)</th>
<th>ORDerly-forward</th>
<th>ORDerly-retro</th>
<th>Non-USPTO-forward</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full dataset</td>
<td>1,771,032</td>
<td>1,771,032</td>
<td>1,771,032</td>
<td>1,771,032</td>
<td>91,067</td>
</tr>
<tr>
<td>Too many reactants</td>
<td>1,470,060</td>
<td>1,631,394</td>
<td>1,743,585</td>
<td>1,631,394</td>
<td>43,845</td>
</tr>
<tr>
<td>Too many products</td>
<td>1,329,399</td>
<td>1,593,196</td>
<td>1,740,655</td>
<td>1,593,196</td>
<td>40,770</td>
</tr>
<tr>
<td>Too many solvents</td>
<td>1,222,381</td>
<td>1,388,312</td>
<td>1,689,445</td>
<td>NA</td>
<td>36,522</td>
</tr>
<tr>
<td>Too many agents</td>
<td>1,202,790</td>
<td>1,279,833</td>
<td>1,550,800</td>
<td>NA</td>
<td>31,187</td>
</tr>
<tr>
<td>No reactants/products</td>
<td>1,202,758</td>
<td>1,262,333</td>
<td>1,533,680</td>
<td>1,567,697</td>
<td>31,095</td>
</tr>
<tr>
<td>No solvents</td>
<td>870,888</td>
<td>950,189</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>No agents</td>
<td>135,139</td>
<td>690,234</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Inconsistent yields</td>
<td>126,948</td>
<td>658,071</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Dropping duplicates</td>
<td>76,634</td>
<td>392,996</td>
<td>919,231</td>
<td>941,566</td>
<td>28,496</td>
</tr>
<tr>
<td>Frequency filtering</td>
<td>75,033</td>
<td>356,906</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>

4.5 Forward prediction benchmark

ORD-ly-forward is a benchmark created from USPTO data in ORD for retrosynthesis prediction consisting of reactions with up to two products and three reactants, solvents, and agents. A random 80/10/10 train/val/test split was applied to the benchmark. An additional test set called non-USPTO-forward was created by using all non-USPTO data in ORD (as of August 2023) and cleaning it with the same parameters as those used for ORD-ly-forward. No frequency filtering was applied.

4.6 Single-step retrosynthesis benchmark

ORD-ly-retro is a benchmark created from USPTO data in ORD for retrosynthesis prediction consisting of reactions with one product and up to two reactants. A random 80/10/10 train/val/test split was applied to the benchmark. An additional test set called non-USPTO-retro was created by using all non-USPTO data in ORD (as of August 2023) and cleaning it with the same parameters as those used for ORD-ly-retro. No frequency filtering was applied.

4.7 Condition prediction benchmark

ORD-ly-condition is a benchmark dataset created from USPTO data in ORD for reaction condition prediction, and is, to the best of our knowledge, the first reaction condition benchmark. Each reaction in ORD-ly-condition contains one product and up to two reactants, two solvents, and three agents. A minimum frequency of 100 for the spectator molecules was applied.

5 Results and discussion

Experimental evaluation of the ORD-ly-forward and ORD-ly-retro benchmarks was performed using the Molecular Transformer architecture built by Schwaller et al. [2]. To switch from forward prediction to retrosynthesis prediction no changes to the transformer architecture were necessary, only the data was changed. The ORD-ly-condition benchmark was evaluated together with the impact of different approaches to reaction role assignment and frequency filtering using the neural network architecture built by Gao et al. [10].

5.1 Forward and retrosynthesis prediction with transformers

Transformers were applied to two tasks: forward prediction (predicting products given reactants, solvents, and agents) and retrosynthesis (predicting reactants given a product). For the task of forward
reaction prediction two different modes were tested: mixing the reactants, solvents, and agents, or weakly separating the reactants from the solvents and agents with a ">" token. Untokenized examples of transformer model inputs are shown in (2)-(4). Forward prediction with mixed inputs is a more difficult task, since it is less obvious which atoms (characters) will appear in the product.

Example input, forward (mixed): Cl.CO.CC1(O)CCC(NCc2ccccccc2)C1(C)F.[Pd] (2)
Example input, forward (separated): CC1(O)CCC(NCc2ccccccc2)C1(C)F>CO.[Pd].Cl (3)
Example input, retrosynthesis: C[C@@](O)CC[C@@H](N)[C@@](1)(O)[C@G](1)(C)F (4)

For both forward and retrosynthesis prediction the order of the molecules was randomized, and the dataset was augmented by replacing each SMILES string in the reaction with a random equivalent SMILES string (thus doubling the dataset size), before finally being tokenized [2].

Performance metrics are reported in Table 2 showing that across all tasks only a small percentage of the generated SMILES strings are invalid.

On the forward prediction tasks the accuracies achieved are similar (albeit slightly lower) to the accuracies reported by Schwaller et al. [2] (88-90% top-1 accuracy when trained on the USPTO_MIT [31] dataset), though the accuracies are not directly comparable since different subsets of USPTO were used. As expected the performance with separated agents is higher than mixed, since it is an easier task, and it is encouraging to see that the models get stereochemical information correct most of the time. Accuracy with the retrosynthesis model on the held out test set was roughly 50%, which is similar previous work on retrosynthesis [34]. It is interesting that prediction accuracy on the non-USPTO data was similar on the forward prediction tasks, but markedly worse on the retrosynthesis task.

**Computational details:** The transformer models were trained for around 35 hours (roughly 600 epochs) on a T4 cloud GPU instance provided by lightning.ai. Evaluation was done with the final model checkpoint.

### 5.2 Reaction condition prediction with neural networks

The reaction condition prediction model used in this work predicts five categorical variables: two solvents and three agents. These five molecules form a set (order invariant), though the loss function in the model used to predict the molecules considers them sequentially (with order) since this was found to work better in practice [10]. The metric used to evaluate the accuracy of the model should be order invariant, since the problem is order invariant, and for this reason the accuracy metrics used are top-1 (see appendix C) and top-3 (see Table 3) exact match combination accuracy for each type of component (i.e., solvent, agent). Beam search was used to identify the top-3 highest probability sets of reaction conditions. The top-3 accuracy was compared to the baseline predictive accuracy of simply predicting on the test set the most common molecules found in the train set.

Additionally, we define a metric inspired by Maser et al. [12] called the average improvement over baseline (AIB%):
Table 3: Top-3 metrics on condition prediction with the model architecture of Gao et al.\(^\text{10}\): frequency informed guess accuracy // model prediction accuracy // AIB%.

<table>
<thead>
<tr>
<th>Datasets:</th>
<th>labeling rare→&quot;other&quot;</th>
<th>labeling rare→delete rxn</th>
<th>reaction string rare→&quot;other&quot;</th>
<th>reaction string rare→delete rxn</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solvents</td>
<td>47 // 58 // 21%</td>
<td>50 // 61 // 22%</td>
<td>23 // 42 // 26%</td>
<td>24 // 45 // 28%</td>
</tr>
<tr>
<td>Agents</td>
<td>54 // 70 // 35%</td>
<td>58 // 72 // 32%</td>
<td>19 // 39 // 25%</td>
<td>21 // 42 // 27%</td>
</tr>
<tr>
<td>Solvents &amp; Agents</td>
<td>31 // 44 // 19%</td>
<td>33 // 47 // 21%</td>
<td>4 // 21 // 18%</td>
<td>5 // 24 // 21%</td>
</tr>
</tbody>
</table>

\[
AIB\% = \frac{A_m - A_b}{1 - A_b} \times 100
\]  

where \(A_m\) is the exact match combination accuracy of the model and \(A_b\) is the exact match combination accuracy of choosing the top 3 most common values of a component in the respective train set.

Table 3 shows the predictive performance on the test set using four different flavours of the ORDERly-condition benchmark. All models show an improvement over the frequency informed baseline. The performance of the labeling datasets at first appears to be better than those that use our custom logic to extract reaction components from the reaction string. However, as shown in Figure 3, many of the reactions in datasets where we trust the labeling in ORD have more than three reactants, while most reactions in organic chemistry only have two reactants. Upon manual inspection, we found that many agents were mislabeled as reactants and, therefore, the prediction problem was made significantly easier by only requiring a single catalyst to be predicted (see Appendix E). In contrast, our custom cleaning pipeline that defines components using the reaction string avoided contamination of the desired prediction targets (i.e., the agents) in the inputs, and therefore, better represents the downstream application of reaction condition prediction models. This insight is confirmed in Table 3; there are fewer unique solvents and agents and a higher density of null components when using the ORD labeling instead of the reaction string indicating that many components might be mislabeled as reactants. This discrepancy demonstrates that naive creation of datasets based on ORD can lead to inflated performance metrics.

For the datasets that extract the components from the reaction string, overall top-3 accuracy is less than 25% across solvents and agents. While not directly comparable, our overall accuracy is lower than what Gao et al.\(^\text{10}\) achieved with 50.1% top-3 accuracy across catalysts, solvents and agents. However, Gao et al. trained on approximately ten million reactions, while we train on less than four percent of that (~350k). As shown in Figure 3, we see consistent increases in AIB (%) with the number of data points for the dataset which uses reaction strings and deletes rare reactions, and this scaling performance indicates that as ORD grows, better performance could be achieved, even with potentially fewer data points than used in the paper by Gao et al.

Finally, the approach to dealing with rare values is investigated. The reaction string datasets would have more than 10,000 unique agents (see Table 1) with no frequency based filtering, which would create a sparse OHE. We initially hypothesized that the rare \(\rightarrow\) "other" strategy would allow for better generalisation, since the edge case reactions would be kept in a way that also keeps the OHE at a reasonable size. However, in practice, the rare \(\rightarrow\) delete rxn strategy had better performance across train set sizes, as seen in Figure 4.

Four flavours of ORDERly-condition datasets were presented in this section. Assigning reaction roles using the reaction string and using the rare \(\rightarrow\) delete rxn strategy for rare spectator molecules were chosen as the preferred cleaning hyperparameters for the ORDERly-condition benchmark since these performed best.

**Computational details:** These models were trained on an A10G cloud GPU instance provided by lightning.ai for 100 epochs to minimize cross entropy loss for each reaction component. The best model by validation loss was chosen for evaluation.
6 Technical limitations

6.1 Component labeling

There are two ways of assigning reaction roles to molecules found in ORD files, either relying on the labeling, or identifying reaction roles by considering the atom mapping of a reaction SMILES string. We found that relying on the labeling in ORD mislabels many spectator molecules as reactants, which explains the difference in reactant count distribution seen in Figure 3. Identifying the role of molecules in a reaction provides crucial context to machine learning models, adding domain knowledge to the data thereby improving performance. Atom mapping the reactions with the newest

Table 4: Diversity in the datasets. Frequency filtering was applied for the solvents and agents to create a more dense one-hot encoding. Columns: Number of unique molecules with a frequency above the threshold; number of unique molecules with a frequency below the threshold; percentage of the dataset that is None.

<table>
<thead>
<tr>
<th></th>
<th>labeling</th>
<th>reaction string</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reactants</td>
<td>40,020</td>
<td>317,184</td>
</tr>
<tr>
<td>Products</td>
<td>38,816</td>
<td>382,850</td>
</tr>
<tr>
<td>Solvents</td>
<td>29</td>
<td>85</td>
</tr>
<tr>
<td>Agents</td>
<td>48</td>
<td>255</td>
</tr>
</tbody>
</table>

Figure 3: Distribution of the number of reactants between the reaction string and labeling datasets after completing other cleaning steps. The labeling dataset contains more reactants per reaction on average; this may be due to agents being mislabeled as reactants.

Figure 4: Scaling behaviour of different datasets with respect to overall top-3 AIB (%) for all solvents and agents (third row from Table 3.)

https://doi.org/10.26434/chemrxiv-2023-qkjtb-v2
ORCID: https://orcid.org/0000-0002-0494-643X
Content not peer-reviewed by ChemRxiv. License: CC BY 4.0
algorithm may allow for greater accuracy in identifying reaction roles, however, an atom mapping algorithm was not integrated into ORDerly to keep ORDerly lightweight. With the existing atom mapping in ORD molecules contributing atoms to the product could readily be bundled together and labeled as reactants. However, subdividing spectator molecules into different categories (e.g. agents, reagents, solvents, catalysts, precatalysts, ligands, acids/bases) is a difficult task. The difficulty is compounded by the fact that the same molecule can play different roles depending on the context. The role that a molecule plays in a reaction may more easily identified when only considering one reaction class, since this allows the mechanistic details of the reaction class to be considered. Handling large and diverse datasets inevitably requires generalizations that may result in contradictions upon a more fine-grained inspection. In this work, solvents were separated from the other spectator molecules, because these can somewhat reliably be identified. Catalysts were not separated into their own category, since identifying catalysts is more subtle (especially with organocatalysis), and few reactions in the reaction string datasets contained transition metals.

6.2 Order invariance

Although order of addition may play a role in wet lab chemistry, reaction prediction tasks are often cast as order invariant, where the goal is to predict a set of molecules. However, both of the architectures used for experimental validation of the ORDerly datasets are not agnostic to the ordering of the targets, since the neural networks used predict one molecule at a time in the OHE, and the transformers used predict one token at a time. Incorporating order invariance (and canonicalization) of the molecules into the loss calculation during training may allow for better generalisability of the predictive models, and is an exciting area for further study. It is worth noting that the evaluation metrics used throughout are order invariant.

7 Conclusions

In this work, we presented ORDerly, an open-source framework for preparing chemical reaction data stored in the Open Reaction Database (ORD) for machine learning applications. ORDerly was used to generate benchmark datasets for forward prediction (ORDerly-forward), retrosynthesis (ORDerly-retro), and condition prediction (ORDerly-condition) based on US patent data. Transformer models were trained on the forward prediction and retrosynthesis datasets, and they were found to only generate invalid SMILES strings very infrequently, while also achieving similar test accuracy to that found in the literature on a held-out set of US patents. ORDerly was also used to generate test sets from all non-patent data from ORD, which could serve as a better indication of model generalisation. Accuracy for the forward prediction task was comparable on the non-USPTO test set, while accuracy on the retrosynthesis task was somewhat lower. The condition prediction task was used to investigate different strategies for assigning reaction roles and frequency filtering of the spectator molecules. When building datasets for condition prediction using the labeling in ORD we found contamination of the inputs (reactants) with the outputs (agents), resulting in a problem that was unrealistically easy. We therefore chose to use chemically informed logic to better assign reaction roles for the ORDerly-condition benchmark.

All benchmarks and datasets experimented with in this work, as well as the code used to generate them, are freely available online, and we hope the benchmarks will make reaction prediction tasks more accessible to ML practitioners with limited domain knowledge. ORDerly presents a fully open-source pipeline to go from raw ORD data to a fully trained condition prediction model, allowing for an avenue to leverage the growing contributions to open source chemistry.

8 Acknowledgements

This work is co-funded by UCB Pharma and Engineering and Physical Sciences Research Council via project EPS/S024220/1 EPSRC Centre for Doctoral Training in Automated Chemical Synthesis Enabled by Digital Molecular Technologies. This project was co-funded by European Regional Development Fund via the project "Innovation Centre in Digital Molecular Technologies".
References


https://doi.org/10.26434/chemrxiv-2023-qkjtb-v2 ORCID: https://orcid.org/0000-0002-0494-643X Content not peer-reviewed by ChemRxiv. License: CC BY 4.0


