Standardization of chemical compounds using language modeling

Miruna T. Cretu
IBM Research Europe, Zürich

Alessandra Toniato
IBM Research Europe, Zürich

Alain C. Vaucher
IBM Research Europe, Zürich

Amol Thakkar
IBM Research Europe, Zürich

Amin A. Debabeche
IBM Research Europe, Zürich

Teodoro Laino
IBM Research Europe, Zürich

Abstract

With the growing amount of chemical data stored digitally, it has become crucial to represent chemical compounds consistently. Harmonized representations facilitate the extraction of insightful information from datasets, and are advantageous for machine learning applications. Compound standardization is typically accomplished using rule-based algorithms that modify undesirable descriptions of functional groups, resulting in a consistent representation throughout the dataset. Here, we present the first deep-learning model for molecular standardization. We enable custom schemes based solely on data, which also support standardization options that are difficult to encode into rules. Our model achieves $> 98\%$ accuracy in learning two popular rule-based protocols. When fine-tuned on a relatively small dataset of catalysts (for which there is currently no automated standardization practice), the model predicts the expected standardized molecular format with a test accuracy of 62% on average. We show that our model learns not only the grammar and syntax of molecular representations, but also the details of atom ordering, types of bonds, and representations of charged species. In addition, we demonstrate the model's ability to reproduce a canonicalization algorithm with a 95.6% success rate.

1 Introduction

From deep learning algorithms for forward reaction prediction [1, 2, 3] and retrosynthesis [1, 4, 5], to the prediction of yields [6] and molecular properties [7], artificial intelligence has become an integral part of chemical discovery pipelines. This was made possible thanks to the abundance of freely available molecular databases, with hundreds of millions of compounds relevant to drug and materials discovery [8, 9, 10]. However, the size of the datasets makes human curation campaigns impractical, resulting in the frequent presence of incorrect and inconsistent molecular structure representations [11]. Because the quality of the input data limits the performance of machine learning models, the development of tools to address this issue has received increased attention in recent years [12]. In fact, a 2010 study [11] compiled a series of investigations which concluded that even minor structural errors and inconsistencies within a dataset could result in significant losses in the predictive ability of structure-activity relationship models. Standardization tools aim to correct errors in chemical structure representation, while also generating uniform and self-consistent configurations of atoms and bonds, charges and bond orders, aromaticity and stereochemistry.

NeurIPS 2022 AI for Science Workshop.

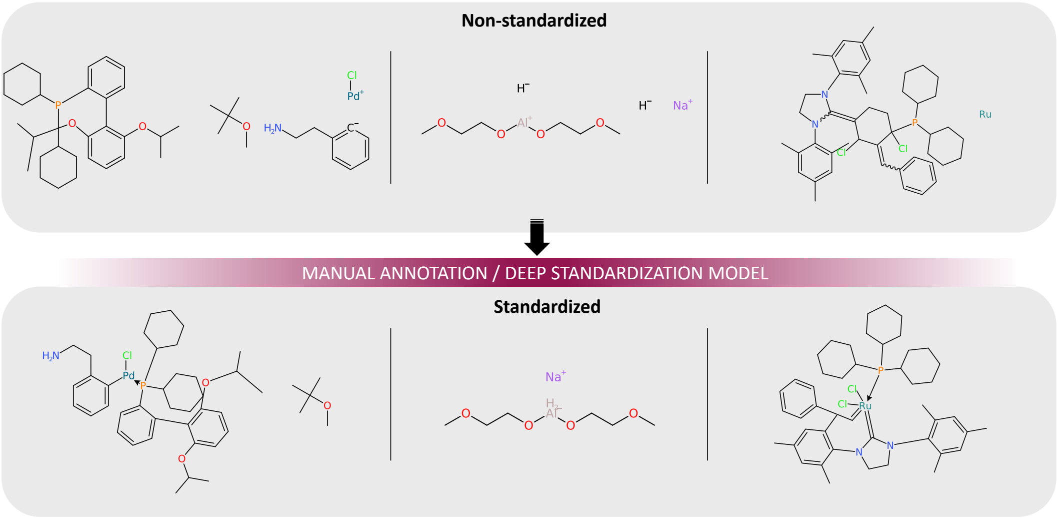

Figure 1: Examples of standardization transformations that cannot be performed by existing rule-based automated protocols due to their complexity. Note that when manually standardizing these compounds, one of many possible conventions was chosen. The language model for standardization presented here correctly predicts all of the above transformations.

Currently, the only approach to chemical data standardization is to format compounds according to a set of rules and conventions [13, 14, 15]. These assume the occurrence of specific patterns in the arrangements of elements, bonds, and charges and necessitate the development of algorithms to convert them to a standard format (which often varies across organizations). Manually crafted and coded rules have inherent disadvantages, the most notable of which is the need for programming expertise and time resources. Even more importantly, it is not always possible to develop a set of rules to automate a chosen standardization protocol, even when experts design specific guidelines to manually standardize the corresponding compounds. Figure 1 depicts a few examples of standardization transformations that cannot be mimicked by a human-written algorithm and thus require manual annotation as of today.

In this work, we propose a deep learning method based on the Transformer architecture [16] that converts molecules represented using the simplified molecular-input line-entry system (SMILES) [17, 18] from their non-standardized to their standardized format. We demonstrate the versatility of the model by training it with two different standardization protocols and allowing the user to select the preferred protocol when standardizing new molecules. We also leverage the pre-trained model to fine-tune it on a dataset where the standardization process cannot be reduced to a set of rules. Thus, when fine-tuned on a dataset consisting of few hundreds of molecules that have been standardized based on human annotation, our model can capture commonalities in molecular structure representation and codify a specific set of rules for consistently modifying compounds.

2 Method

2.1 Model

We repurposed the transformer architecture [16] to execute a translation task from non-standardized to standardized molecular representations. Tokenized SMILES strings are used as source and target inputs to the model and tokenization is performed using a custom regular expression pattern (see Appendix A). Among other aspects, the tokenization ensures the separation of metal atoms and their charge, which simplifies the learning of atom identity rather than its charged state.

The transformer model was implemented using the `OpenNMT-py` library (version 1.0.0). The parameters used are reported in Appendix B.

2.2 Data

The model was trained and tested on compounds belonging to PubChem [8], an open archive of chemical compounds. The data (220k molecules) was acquired in non-standardized and in standardized formats which served as source and target inputs, respectively. We performed validation on 8k compounds and tested on 12k compounds. Upon inspection of the PubChem standardization protocol, we observed that several compounds are subject to addition of stereochemistry based on provided 3D information for the molecule. In absence of 3D information as input to our model, we removed stereochemistry when comparing predictions to true values.

To evaluate the model on another standardization scheme, the source molecules extracted from PubChem were standardized using the ChEMBL curation protocol [14]. Finally, a private dataset is introduced in this work and comprises 866 catalyst molecules which we standardized manually. This was further split into training (766 molecules), validation (50 molecules) and test (50 molecules) sets. We performed 5 splits and report average performance.

3 Results and discussion

3.1 Standardization learning

The following is an analysis of the ability of the model to perform various types of molecule standardization. We used three different datasets to train the model. The source molecules are fixed across them and the targets are generated using: 1) the ChEMBL standardization rules, 2) the PubChem standardization rules, and 3) a combination of the two procedures.

Table 1: Performance of standardization models tailored for different protocols. Accuracies are reported for the whole test dataset, as well as only for compounds that get modified during rule-based standardization.

Standardization protocol	Accuracy (%)		Split
	overall	modified	
ChEMBL	98.8	94.5	Random
ChEMBL	96.7	87.8	Tanimoto
PubChem	98.0	91.5	Random
PubChem	94.9	80.1	Tanimoto
ChEMBL & PubChem	98.5	92.7	Random

First, we show that the model can be adapted to learn different standardization protocols, to accommodate distinct preferences in molecular formatting. Table 1 contains a summary of the results, including accuracy on the entire test dataset as well as solely on the compounds modified during rule-based standardization. The datasets were split in two ways: randomly and based on Tanimoto indexes, which are a popular measure of the structural similarity between compounds [19]. As such, we adopted the method of Kovács et al. [20] to allocate compounds to the training/test datasets so that no compound in the test set is within Tanimoto similarity $\sigma = 0.6$ of any compound in the training set. The intent of such a split is to avoid structural bias and to make a robust evaluation of the model's ability to generalise to unseen structures.

ChEMBL The results from the random split reveal that the model successfully learns the task, with an accuracy of 94.5% for compounds standardized using the ChEMBL protocol [14]. This involves modifications such as converting covalently drawn alkaline metals connected to O or N to ionic forms (e.g. NaO to Na^+O^-), standardizing diazonium N to N^+, removing explicit H atoms etc. The model recognizes the molecules that can be standardized and achieves an overall test accuracy of 98.8%. Hence, the model can also predict that the compounds do not require modification and leaves the string unchanged. When using a Tanimoto split, the accuracy drops to 87.8%, which is a testament to the scaffold bias introduced by the nature of the dataset.

PubChem Contrary to ChEMBL, PubChem standardization [13] uses a more extensive list of rules relying on routines from the OpenEye C++ toolkits [21]. Figure 2 exemplifies a few distinctions between the two approaches. While the ChEMBL protocol maintains the orginal stereochemistry of

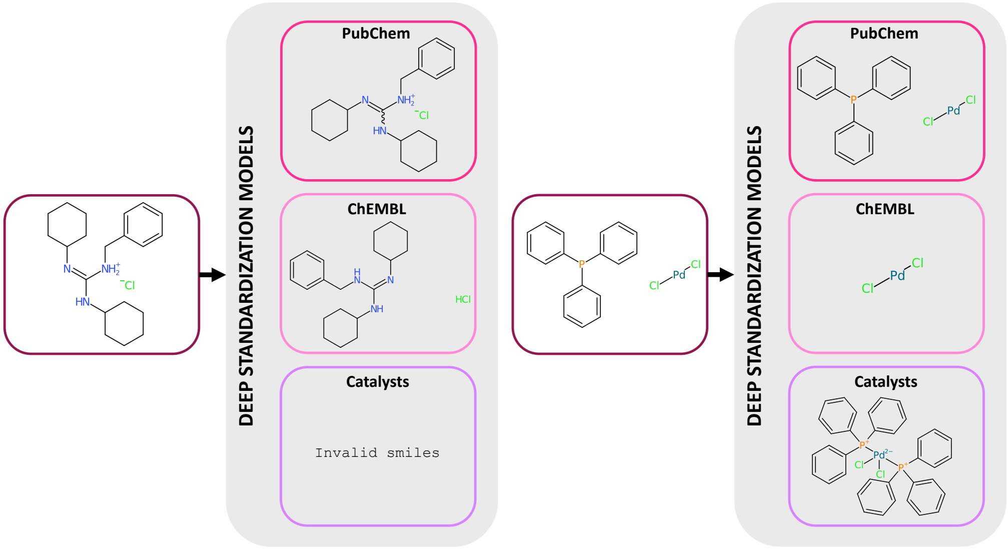

Figure 2: Distinct standardization outcomes for different protocols. All predictions are made using the translation models trained herein. 'PubChem' is the deep standardization model trained on PubChem compounds, 'ChEMBL' is the one trained on compounds following ChEMBL standardization and 'Catalysts' refers to the model fine-tuned on the humanly-curated catalyst dataset. Note that without fine-tuning, the catalyst molecule (right) is erroneously standardized.

the double bond, PubChem standardization converts the original geometry to an unspecified configuration (represented by the wiggly bond). PubChem also standardizes the amine to its protonated state. Overall, with a random split, the model learns the PubChem protocol with a test set accuracy of 98.0%. For compounds that require modification, 91.5% of the predictions match the real structure.

ChEMBL and PubChem Finally, we explored a model that combines the two different standardization protocols presented previously in a prompt-based fashion. The assumption is that competing probability distributions, modelling different standardization protocols, can mutually learn from each other, delivering substantial benefit to the learning. In practice, a `[CHEMBL]` / `[PUBCHEM]` token was added to the input SMILES sequence to designate the preferred type of standardization. The two protocols were successfully learned in a combined model, with an overall test accuracy of 92.7%. We determined the test accuracies with which specifically the ChEMBL / PubChem rules were learned in the combined model. These remain unaltered compared to when the model was trained on one protocol only: 94.3% for ChEMBL and 91.4% for PubChem, which is in line with the expected cross-task benefits of multitasking [22]. This is a key finding that enables the user to train on multiple standardization protocols and query the model effectively.

3.2 Model transferability

Transferability was assessed by fine-tuning the above trained models (see Table 1) on a private human-curated catalyst dataset. Not only do the compounds in this set deviate from the pre-training data in terms of class and vocabulary, but the standardization rules used here are also unique. The performance showed a maximum test set accuracy of $62.0 \pm 4.0\%$ (see Table 2) on a relatively small dataset (see Section 2.2), an increase of 18% compared to a non-pre-trained model. The learning abilities of the model for this dataset are notable, as the performance of a null model in this case (which always returns the input string as a prediction) is 0%. It is also worth noting the variety and complexities of the transformations. Particularly important for catalysts, the model appears to learn the ligation preferences of metal centres. Fig. 1 shows how distinct metals exhibit different coordination behaviours, which are otherwise challenging to capture with rule-based algorithms.

Upon visual inspection of wrong predictions, we observed that the model is prone to fail for large molecules, however the limited size of the test set does not allow for a proper statistical evaluation and further investigation is needed. Fig. 3 reveals that incorrect predictions can be associated to

Table 2: Evaluation of the model's ability to perform standardization of catalysts. The test set top-1 accuracies are reported. Note that the first row refers to a model which is only trained on the catalyst dataset, whereas the next two rows refer to models pre-trained on PubChem and ChEMBL standardized data, respectively, and fine-tuned on the catalyst dataset.

Pre-training	Accuracy (%)
None	44.0 ± 3.8
PubChem	62.0 ± 4.0
ChEMBL	61.6 ± 2.9

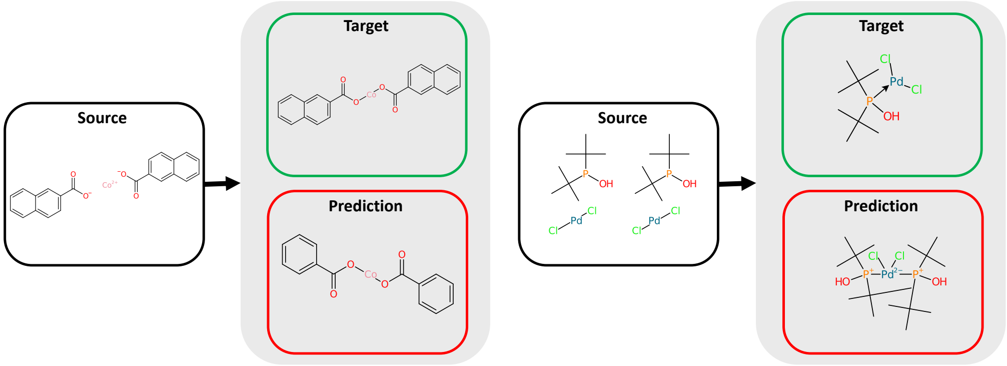

Figure 3: Incorrect predictions of the fine-tuned catalyst model.

incomplete structures or incorrect standardization, and not to chemically invalid structures. This finding, along with the boost in accuracy highlighted in Table 2, suggests that pre-training contributes to preventing the violation of chemical rules, while fine-tuning is accountable for tailoring the model to a specific set of standardization rules. At the same time, we notice that the fine-tuned model loses its ability to standardize molecules belonging to the pre-training dataset (see Fig. 2). A multitask learning approach could in principle overcome the issue.

3.3 Learning canonicalization

As a final experiment, we randomized SMILES strings by doing a cyclic rotation of the atomic indices, and presented the model with the task of recovering the canonical analogues of the molecules. This was achieved with an accuracy of 95.6% on a random test set. Canonicalization was performed using the algorithm provided by RDKit [23].

4 Conclusions

The model for molecule standardization presented in this work is the first attempt to replace its rule-based predecessors. We have demonstrated a robust method that accommodates the variety of molecular representations that exist today. First, it learns two popular standardization procedures with accuracies $> 98\%$. For the compounds that undergo modifications during rule-based protocols, the model predicts the correct outcome with test set accuracies $> 91\%$. The model can be trained on multiple procedures simultaneously, allowing the user to query it in a prompt-based fashion and select the preferred standardization practice. Importantly, this does not reduce the standardization accuracy. When presented with a small catalyst dataset with numerous formatting possibilities, the model learned the preferred standardizations with an average test accuracy of 62%.

Introducing additional features to the current development may further extend the model's capabilities. For example, we plan to explore the effect of the representation of source molecules on the predictive abilities of the model. This will build towards the goal to learn an even more robust standardization, irrespective of the flavour of the input SMILES.

References

[1] Marwin H. S. Segler and Mark P. Waller. Neural-symbolic machine learning for retrosynthesis and reaction prediction. *Eur. J. Chem.*, 23(25):5966–5971, 2017.

[2] Connor W. Coley, Wengong Jin, Luke Rogers, Timothy F. Jamison, Tommi S. Jaakkola, William H. Green, Regina Barzilay, and Klavs F. Jensen. A graph-convolutional neural network model for the prediction of chemical reactivity. *Chem. Sci.*, 10:370–377, 2019.

[3] Philippe Schwaller, Teodoro Laino, Théophile Gaudin, Peter Bolgar, Christopher A. Hunter, Costas Bekas, and Alpha A. Lee. Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction. *ACS Cent. Sci.*, 5(9):1572–1583, 2019.

[4] Marwin H. S. Segler, Mike Preuss, and Mark P. Waller. Planning chemical syntheses with deep neural networks and symbolic AI. *Nature*, 555:604–610, 2018.

[5] Philippe Schwaller, Riccardo Petraglia, Valerio Zullo, Vishnu H. Nair, Rico Andreas Haeuselmann, Riccardo Pisoni, Costas Bekas, Anna Iuliano, and Teodoro Laino. Predicting retrosynthetic pathways using transformer-based models and a hyper-graph exploration strategy. *Chem. Sci.*, 11:3316–3325, 2020.

[6] Philippe Schwaller, Alain C. Vaucher, Teodoro Laino, and Jean-Louis Reymond. Prediction of chemical reaction yields using deep learning. *Mach. Learn.: Sci. Technol.*, 2(1):015016, 2021.

[7] Jie Shen and Christos A. Nicolaou. Molecular property prediction: recent trends in the era of artificial intelligence. *Drug Discovery Today: Technol.*, 32-33:29–36, 2019.

[8] PubChem. https://pubchem.ncbi.nlm.nih.gov/#. (Accessed September 2022).

[9] ChEMBL. https://www.ebi.ac.uk/chembl/. (Accessed September 2022).

[10] ChEBI. https://www.ebi.ac.uk/chebi/downloadsForward.do. (Accessed September 2022).

[11] Denis Fourches, Eugene Muratov, and Alexander Tropsha. Trust, but verify: On the importance of chemical structure curation in cheminformatics and QSAR modeling research. *J. Chem. Inf. Model.*, 50(7):1189–1204, 2010.

[12] A Guide to Molecular Standardization. https://depth-first.com/articles/2020/07/27/a-guide-to-molecular-standardization/. (Accessed September 2022).

[13] Volker D. Hähnke, Sunghwan Kim, and Evan E. Bolton. Pubchem chemical structure standardization. *J. Cheminf.*, 10(1):36, 2018.

[14] A. Patrícia Bento, Anne Hersey, Eloy Félix, Greg Landrum, Anna Gaulton, Francis Atkinson, Louisa J. Bellis, Marleen De Veij, and Andrew R. Leach. An open source chemical structure curation pipeline using RDKit. *J. Cheminf.*, 12(1):51, 2020.

[15] Timur R. Gimadiev, Arkadii Lin, Valentina A. Afonina, Dinar Batyrshin, Ramil I. Nugmanov, Tagir Akhmetshin, Pavel Sidorov, Natalia Duybankova, Jonas Verhoeven, Joerg Wegner, Hugo Ceulemans, Andrey Gedich, Timur I. Madzhidov, and Alexandre Varnek. Reaction data curation I: Chemical structures and transformations standardization. *Mol. Inf.*, 40(12):2100119, 2021.

[16] 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, *Adv. Neural. Inf. Process. Syst.*, volume 30. Curran Associates, Inc., 2017.

[17] David Weininger. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.*, 28(1):31–36, 1988.

[18] David Weininger, Arthur Weininger, and Joseph L. Weininger. SMILES. 2. Algorithm for generation of unique SMILES notation. *J. Chem. Inf. Comput. Sci.*, 29(2):97–101, 1989.

[19] Dávid Bajusz, Anita Rácz, and Károly Héberger. Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations? *J. Cheminf.*, 7(1):20, 2015.

[20] Dávid Péter Kovács, William AU McCorkindale, and Alpha A. Lee. Quantitative interpretation explains machine learning models for chemical reaction prediction and uncovers bias. *Nat. Commun.*, 12(1):1695, 2021.

[21] OpenEye Quacpac C++Toolkit, version 1.9.0. https://www.eyesopen.com/quacpac-tk. (Accessed September 2022).

[22] Andreas Mayr, Günter Klambauer, Thomas Unterthiner, and Sepp Hochreiter. Deeptox: Toxicity prediction using deep learning. *Front. Environ. Sci.*, 3, 2016.

[23] RDKit. https://www.rdkit.org/docs/GettingStartedInPython.html. (Accessed September 2022).

A Tokenizer

SMILES strings were tokenized using the following regex expression:

```
(\%\([0-9]{3}\)|\[[^\]]+]|Br?|Cl?|N|O|S|P|F|I|b|c|n|o|s|p|\||\(|\)|\.|=|#|-|\+|\\|\/
|:|~|@|\?|>>?|\*|\$|\%[0-9]{2}|[0-9])
```

The resulting tokenized string was post-processed using the following expression, to introduce a space between metal atoms and their charge:

```
\[([A-Z][a-z]?)([+-][0-9]*)?\]
```

B Model implementation and training

The model is based on the default transformer implementation provided by `OpenNMT-py`, which was adapted through the following changes: the parameter `layers` was set to 4, `rnn_size` to 256, `word_vec_size` to 256, `max_generator_batches` to 32, `accum_count` to 4 and `label_smoothing` to 0. The model was trained for 120,000 steps. When fine-tuning on the catalyst dataset, 30,000 training steps were used.