Prediction of Drug Metabolites using Neural Machine Translation

Eleni E. Litsa[a], Payel Das[*b], and Lydia E. Kavraki[*a]

[a]Department of Computer Science, Rice University, Houston, TX
[b]IBM Research AI, IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598
[b]Applied Physics and Applied Mathematics,Columbia University, New York, NY 10027
[*]daspa@us.ibm.com, kavraki@rice.edu

SUPPLEMENTARY INFORMATION

S1 Data Preparation

S1.1 Data Augmentation

SMILES Augmentation

- **Pre-training**
  For pre-training the model we explored three variations of the chemical reactions dataset: 1) The raw data where both sides of the reaction are represented with canonical SMILES, 2) an augmented version of the dataset where only the SMILES of the reactants are randomized and 3) a second augmented version where both the SMILES in both sides of the reaction are randomized. For both augmented versions only one additional instance was introduced doubling the size of the initial dataset. The two augmented datasets resulted in better performance comparing to the model trained on the non-augmented dataset. For building the ensemble model, the two models resulted from the two augmented datasets were used as different starting points in order to achieve larger variability among the combined models.

- **Fine-tuning**
  For fine-tuning, we created multiple variations of the training set experimenting with whether both, reactant and product SMILES, are randomized or only the reactant as well as the amount of data augmentation. Regarding the amount of data augmentation, we created multiple versions of the data starting with the raw data where all molecules are represented with canonical SMILES. Then we created three variations of the data by adding one, two and three additional randomized instances respectively. If the randomization of a SMILES resulted in the generation of a SMILES that already existed in the dataset then the additional instance was not added in the dataset. The randomization of the SMILES for the creation of each data variant was done independently in order to increase the diversity among the datasets. We repeated this process twice, generating 3 variants of the dataset where only the reactant SMILES is randomized and three additional variations where both, reactant and product are randomized. We trained multiple models using the raw canonical data as well as the augmented variations of the data and finally experimented by combining the outputs of the models merging their predictions. The selection of the final models were based on the accuracy on the validation set. It is interesting that the best combination we achieved was indeed achieved through multiple variations of the dataset.

SMIRKS Augmentation SMIRKS augmentation was only applied on the dataset of metabolites for fine-tuning and only on the instances that were described with a SMIRKS representation instead of a SMILES string. Such cases were derived from MetaCyc and Recon3D and HMDB. In order to generate valid pairs of parent molecules and metabolites from the SMIRKS instances, we replaced the general atoms in SMIRKS, denoted with a star (*) symbol, with common atoms in organic chemistry (C, O, S, H, N) and subsequently checked the validity of the generated SMILES using RDKit. For cases with a single generic atom (*), either in reactants or products side, we created multiple instances replacing each time the generic atom with a different atom species. For cases with more than one generic atoms we only replaced the generic atoms with C atoms. Each reactant SMILES generated through this process was added in the dataset only if it was not found in the dataset among the raw/non-augmented cases.

S1.2 Data Filtering

The data that we sourced from the various databases in order to construct the dataset for fine-tuning, as well as the data that were generated through data augmentation were filtered. More specifically, the following instances were removed:

- The parent compound or the metabolite is a hydrogen atom.
- The parent compound and the metabolite are both single atoms.
- The metabolite is a single heavy atom while the parent compound has more than 10 heavy atoms. Same for the opposite case where the reactant is a single atom while the product has more than 10 atoms.
- The metabolite consists of two heavy atoms while the parent compound has more than 20 heavy atoms. Same for the opposite case.

S1.3 Data Processing

Below we provide the python functions we used for canonicalizing and also randomizing the SMILES representations using RDKit. Additionally we provide the tokenization function. Tokenization is applied on the input sequence (during training and during inference) prior to applying the translation model. It breaks down the input sequence into tokens similarly to breaking down a sentence into words.

- Canonicalize SMILES

```
def canonicalize_smiles(smiles):
    from rdkit import Chem
    mol = Chem.MolFromSmiles(smiles)
    canonical = Chem.MolToSmiles(mol, isomericSmiles=True)
    return canonical
```

- Randomize SMILES

```
def randomize_smiles(smiles):
    from rdkit import Chem
    mol = Chem.MolFromSmiles(smiles)
    random = Chem.MolToSmiles(mol, canonical=False, doRandom=True, isomericSmiles=False)
    return random
```

- Tokenize SMILES sequences (from https://github.com/pschwllr/MolecularTransformer)

```
def smi_tokenizer(smiles):
    import re
    pattern = "(\[[^\]]+]|Br?|Cl?|N|O|S|P|F|I|b|c|n|o|s|p|\(|
\)|\.|=|#|-|\+|\\\\|\/|:|~|@|\?|>|\*|\$|\%[0-9]{2}|[0-9])"
    regex = re.compile(pattern)
    tokens = [token for token in regex.findall(smiles)]
    assert smi == ''.join(tokens)
    return '␣'.join(tokens)
```

S2 Transformer model: Implementation & Hyperparameters

S2.1 Model architecture

The Transformer model is a sequence translation model based on a deep learning architecture[1]. It consists of two parts: 1) An encoder which operates on the input -tokenized- sequence and maps it to a vector representation in the continuous space, and 2) a decoder, which takes as input the output of the encoder and generates the output sequence, sequentially (one character at a time). Both parts, encoder and decoder, consist of multiple identical layers whose main component is self-attention layers in addition to the standard fully connected neural network layers. The general idea behind the attention layers is that they allow the encoder (or the decoder) to pay attention at different parts of the sequence. Self-attention in particular is used in order to identify dependencies between different parts (words) within a sentence. Technically, attention is a function which maps a query (Q) and a key (K) and a value (V) to an output.

$$Attention(Q,K,V) = \frac{QK^T}{\sqrt{d_k}}V$$

The query, key and value are auxiliary vectors introduced to facilitate the calculations for the attention mechanism. The encoder contains multiple layers of attention which allow the model to focus at different positions. The values of the key and value vectors of the final attention layer from the encoder are used by the decoder allowing it to generate the output sequence while paying attention at different positions of the input sequence.

S2.2 Implementation

For the Transformer model we used the open source implementation from OpenNMT[2].

S2.3 Pre-training Hyperparameters

The Transformer model was pre-trained on the dataset of general chemical reactions based on the specifications given in https://github.com/pschwllr/MolecularTransformer[3]. More specifically, the Transformer model has an RNN size of 256, 8 heads and feed forward size 2048. It was trained for 500,000 training steps with a batch size 4096.

We additionally experimented with:

- Reducing the number of parameters by reducing the size of the feed forward neural network. We experimented with sizes of 512, 1024 and 2048.

- Increasing the batch size to 6000.
- Data augmentation technique: We used SMILES augmentation and we experimented with randomizing both, source and target sequences, and also randomizing only the source sequence while keeping the canonical representation of the output.

S2.4 Fine-Tuning Hyperparameters

For fine-tuning the pre-trained model, we explored the following options:

- Transfer learning by: i)fine-tuning the entire model, and ii) freezing part of the pre-trained model.
- SMILES augmentation technique: i) randomizing both input and output SMILES, ii) randomizing only the input and iii) experimenting with the number of added cases per instance.
- Learning rate schedule: i) Noam scheduler with a linear warm-up and a decrease proportional to the inverse square root of the number of steps[1], and ii) linear warm-up with linear decay.
- Learning rate strength and number of warm-up steps.
- Batch size. We experimented with sizes: 4, 8, 16, 32 and 64.
- Averaging the weights of the model across different checkpoints[4].

The optimal settings were chosen based on the accuracy on the validation set. The most promising configurations were subsequently combined to form an ensemble model in a bagging fashion. The options that were excluded early on, as less promising, were i) freezing part of the pre-trained model, and ii) weight averaging (possibly due to the small number of training steps when fine-tuning). The criterion for selecting the hyper-parameters of the individual models of the ensemble was based on finding a trade-off between maximising the number of correctly identified metabolites and keeping the number of false positives low based on the results on the validation set. This selection process resulted in 6 individual models whose hyper-parameter settings are provided in Table S1. All models use a droput of 0.1, adam optimiser with beta1 0.9 and beta2 0.98. Batch grouping and normalisation are based on sentences (while for pre-training they are tokens-based). The augmentation technique in Table S1 refers to SMILES augmentation and indicates whether the target sequence is randomized or not, in addition to randomizing the input. In the case of fine-tuning, we additionally explored the number of additional entries that are created per training instance.

Table S1. Training specifications of the 6 selected models.

Parameter	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
feed forward size	1024	2048	1024	2048	1024	2048
augmentation (pre-train)	canonical	random	random	random	random	random
batch size (pre-train)	6000	6000	6000	6000	6000	6000
augmentation (fine-tune)	(canonical) ×3	(canonical) ×1	(canonical) ×1	(random) ×1	(random) ×1	no augm.
batch size (fine-tune)	16	8	8	4	4	4
learning rate	0.5	0.3	0.5	0.3	0.2	0.3
learning rate decay method	Noam	Noam	Noam	Noam	Noam	Noam
warmup steps	2000	2000	2000	2000	2000	2000
training steps	16000	10000	15000	12000	12000	10000

S2.5 Inference/Translation Hyperparameters

For translating a new input sequence, we set the following constraint: the length of the generated sequence, that is the predicted SMILES representation, should be between 5 and 120. These values were selected based on the validation data. Especially the lower bound was introduced to avoid trivial small metabolites or even empty sequences. The upper limit proved to be adequate for small molecules like drugs.

For the beam size, we tried the values 2, 5, 10, 15 and 20. In practice, we noticed that it is possible that metabolites that are correctly identified with smaller beam sizes they are no longer being identified with a larger beam size. For that purpose, for a beam size of 10 the output of the model is the union of the output when using a beam size of 10 and a beam size of 5. In general, for beam sizes larger than 10 we are additionally merging the output with a beam size 5. Since there is a significant amount of overlap between the predictions of different beam sizes, this practice does not significantly affect the total number of metabolites that are predicted per molecule.

We additionally explored the effect of adding i) length penalty, and, ii) coverage penalty from the OpenNMT implementation which were not found to be beneficial.

S3 Evaluation on Training & Validation sets

Here we present the evaluation of the ensemble model on the three partitions of the data: training, validation and testing. Before that we need to highlight the differences between the three datasets: The training set consists of a diverse set of human metabolic reactions including metabolism of endogenous compounds and xenobiotics. The validation set and test set include specifically drug molecules. As a result, the molecules in the training set are more diverse in terms of molecular size. Since our work is focused on small molecules and specifically drug-like molecules, for the evaluation on the training and validation set we have excluded molecules with more than 100 atoms and molecules with less than 6 atoms. All the molecules in the test fall within that range. The statistics of the data, after filtering-out the extreme cases, are presented in Table S2. We also indicate the total number of compounds per dataset before filtering out very small and large molecules.

Table S2. Statistics of the three datasets after removing very large and very small molecules.

	training	validation	testing
Content	Xenobiotics & Endogenous	Drugs	Drugs
Total number of compounds	5851	100	84
Tested compounds	5598	98	84
Number of metabolites	10619	232	217
Avg. number of atoms	28.6	25.6	24.6
Avg. number of metabolites per compound	1.9	2.4	2.6
Max number of atoms	100	65	62
Min number of atoms	6	8	8

The evaluation of the ensemble model on the three partitions of the data is presented in Table S3. The results correspond to beam size 5 which corresponds to roughly top-10 predictions (on average 10 predictions per input molecule). The table shows: i) the percentage of molecules for which the model has identified at least one metabolite, ii) the percentage of metabolites that have been identified with respect to the total number of verified metabolites (recall), iii) the percentage of identified metabolites with respect to the output size of the model (precision), and, iv) the percentage of molecules for which the model did not predict any valid SMILES.

According to the results, the higher precision and recall values are obtained for the training set as expected. A less expected outcome is the higher performance on the test set comparing to the validation set. The explanation behind this lies possibly on the difference in the construction of the two datasets: The cases in the validation set were randomly sampled from the content of DrugBank with only restriction to include metabolites from hydrolases and glucurosyltransferases on top of the prevalent CYP450 metabolites. On the contrary, the big majority of the cases in the test set were manually selected. The molecules obtained from the GLORY test set ere manually chosen. Regarding the molecules from DrugBank, 19 drugs were manually selected while the rest 36 were random. Inadvertently the manual selection of test cases introduces a bias in terms of complexity. More specifically, we noticed that the validation set included cases with increased complexity, that is large number of atoms and complex structure, comparing to the molecules in the GLORY test set as well as the additional test cases

Table S3. Evaluation of the ensemble model on the three partitions of the data for beam size 5 (top-10).

	at least one metabolite	recall (%)	precision (%)	invalid (%)
Training	84.9	81.5	24.4	0
Validation	73.5	56.5	14.0	0
Test	90.5	57.6	14.5	0

from DrugBank. Taken into account the larger diversity of the validation set, the performance on the validation set seems encouraging. Regarding the percentage of invalid predictions, we see that the model has generated at least one valid SMILES prediction for all cases in the training, validation and test sets for a beam size of 5. We noticed that in the training set, which includes molecules on the edge of the upper limit of the molecular size (100 atoms), we can see that invalid predictions are possible for larger beam sizes, but still very unlikely.

S4 Accuracy per enzyme for the full test set

Table S4 presents the performance of the ensemble model for each enzyme family for the test set of 84 drugs for beam sizes of 5, 7 and 10. The beam sizes of 5, 7 and 10 correspond to top-10, top-13 and top-20 predictions.

Table S4. Accuracy per enzyme family.

	Oxidation	UDP-GT	Sulfo-transferases	Other Trasferases	Hydrolases	Unspecified	All
Total	127	18	7	4	9	52	217
Beam 5	73	8	5	2	7	30	125
Beam 7	77	9	5	2	7	32	132
Beam 10	82	10	5	2	7	33	139

S5 Invalid Predictions and Post-processing

Table S5 shows the average number of invalid predicted SMILES per input for the ensemble model. It additionally shows the effect of post-processing steps for different beam sizes.

Table S5. Average number of invalid predictions per input and the effect of post-processing on the total output size.

Beam Size	Average invalid predictions per input	Total output size before filtering	Total output size after filtering
2	0.29	421	419
5	0.93	863	861
10	2.5	1685	1680
15	3.6	2445	2438

S6 Non-metabolized drugs

Although the training set we used for fine-tuning does not contain negative cases, we challenged our method using a set of non-metabolizing drugs. More specifically, through DrugBank we sourced 74 drugs for which it is explicitly stated that no metabolites have been observed in humans. We applied MetaTrans, as well as the three existing tools, on these cases to investigate their capability to identify non-metabolizing molecules.

For MetaTrans with a beam size of 2, the structure of the parent molecule was among the predicted molecules for 38 cases (51.4%). BioTransformer reported zero metabolites for 9 cases while for 2 additional cases the parent structure was among the predicted ones. In the case of GLORYx, the parent structure was found among the predictions for 6 cases within the top-5 or even top-10 predictions. Regarding SyGMa, we noticed that the parent compound is always within the list of output metabolites and therefore we considered a drug a drug as non-metabolizing if the list of metabolites includes only the parent compound which was the case for 3 drugs. Looking more closely at MetaTrans, although it has identified the unchanged parent structure among the predicted metabolites for more than half of the drugs, still its capacity to identify non-metabolizing drugs is limited. More specifically, the unchanged structure was also identified among the predicted metabolites even for cases in the testset of metabolizing drugs but at a lower extent (42.4% for the test set of 85 drugs). The same pattern was also observer for larger beam sizes: the percentage of drugs for which the unchanged parent structure was observed among the predicted metabolites was higher in the set of non-metabolising drugs comparing to the set of metabolising drugs. For larger beam sizes though the discrepancy was smaller. However, MetaTrans provides intentionally a diverse output, mostly through ensembling, and therefore it is very unlikely to provide as output zero metabolites or solely the unchanged parent structure as BioTransformer and SyGMa can do. Still, none of the compared methods demonstrates a clear advantage on the dataset of non-metabolizing drugs. Nevertheless, we should mention that the fact that no metabolites have been identified for the specific drugs may not necessarily mean that no metabolites can be formed.

S7 Additional Experiments

The method we have presented takes as input the SMILES representation of a chemical compound and predicts the structures of possible human metabolites. We have explored a variation of this method which additionally considers the enzyme in the input sequence. More specifically the input to the sequence translation model consists of the SMILES of the chemical compound concatenated with the enzyme information. The enzyme is represented using the first two numbers of the EC (Enzyme Commission) number. The EC number is preceded by a special token to separate the two sequences. The motivation behind this approach is that the formed metabolites depend not only on the parent structure but also on the metabolising enzyme. Additionally, by specifying the enzyme in the input sequence, the predicted metabolites are associated with a specific enzyme. This way the user can control the enzymes for which metabolites are predicted.

Our analysis showed that with this approach the model learnt to predict relevant metabolites for each enzyme. If for example the input enzyme is a CYP450 enzyme then the predicted metabolites are oxidized versions of the input molecule. In the case of sulfotransferases, the output molecules are conjugated with a sulfate structure. However, we are presenting the approach that does not include the enzyme as input for the following reasons: 1) In practical applications it is more useful to let the model predict which metabolites are possible rather than requesting metabolites through specific enzymes otherwise metabolites through unexamined enzymes may be missed. Additionally our analysis shows that even if the enzyme is not specified in the input the predicted metabolites are still relevant. This means that even when the enzyme is not specified the model is able to identify which reactions are possible. For example, we noticed that in the cases where the model predicted metabolites through transferase reactions, the molecule had actually transferase metabolites. This is a very significant outcome because it indicates that the model can predict which reactions are possible for a given compound. 2) For a big part of our dataset the enzyme was not indicated and therefore we had to use either part of the dataset or the entire dataset without the enzyme being always indicated. 3) The overall performance of the model which included the enzyme information was no better than the model without the enzyme possibly because the enzyme information was not included in the dataset for pre-training (this information is not relevant anyway for chemical reactions) and therefore more data are needed for the model to learn this additional information.

References

1. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, “Attention is all you need,” pp. 5998–6008, 2017.

2. G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush, “Opennmt: Open-source toolkit for neural machine translation,” *CoRR*, vol. abs/1701.02810, 2017.

3. P. Schwaller, T. Laino, T. Gaudin, P. Bolgar, C. A. Hunter, C. Bekas, and A. A. Lee, “Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction,” *ACS Central Science*, vol. 5, no. 9, pp. 1572–1583, 2019.

4. P. Izmailov, D. Podoprikhin, T. Garipov, D. P. Vetrov, and A. G. Wilson, “Averaging weights leads to wider optima and better generalization,” *CoRR*, vol. abs/1803.05407, 2018.