Levenshtein Augmentation Improves Performance of SMILES Based Deep-Learning Synthesis Prediction

Dean Sumner,[1] Jiazhen He,[1] Amol Thakkar, [1 2] Ola Engkvist,[1] and Esben Jannik Bjerrum[1 *]

[1] Hit Discovery, Discovery Sciences, R&D, AstraZeneca, Gothenburg, 431 50, Sweden.

[2] Department of Chemistry and Biochemistry, University of Bern, Bern, CH-3012, Switzerland.

*Corresponding authors: esben.bjerrum@astrazeneca.com

Abstract

SMILES randomization, a form of data augmentation, has previously been shown to increase the performance of deep learning models compared to non-augmented baselines. Here, we propose a novel data augmentation method we call "Levenshtein augmentation" which considers local SMILES sub-sequence similarity between reactants and their respective products when creating training pairs. The performance of Levenshtein augmentation was tested using two state of the art models - transformer and sequence-to-sequence based recurrent neural networks with attention. Levenshtein augmentation demonstrated an increase performance over non-augmented, and conventionally SMILES randomization augmented data when used for training of baseline models. Furthermore, Levenshtein augmentation seemingly results in what we define as *attentional gain* – an enhancement in the pattern recognition capabilities of the underlying network to molecular motifs.

Introduction

Recent developments and the accessibility of AI methods have driven an increase in the application of probabilistic machine learning models to chemical problems. One such problem is computer aided synthesis prediction (CASP), which has been explored for the past 60 years, and one which is now experiencing an increased level of interest due to technological advances in deep-learning research. Previously, rules governing chemical reactivity were manually curated by expert chemists[1] - a method which is ultimately infeasible considering the rate of growth of the chemical literature. This lead to a wish to automatically curate templates based on bond breaking, bond formation, and atom mappings between reactants and products. These templates constitute a set of generalizable rules that capture reaction transformations and this methodology became known as template based synthesis prediction. Once the templates has been obtained, it is possible to train neural networks to infer the most probable template

or “policy” to apply for a given set of reactants for the successful prediction of the ground-truth synthetic product[2], or with respect to retrosynthesis, a set of inferred reactant predictions as demonstrated by Segler and Waller[3].

An alternative and more probabilistic method to the temple-based approach, operating on the level of SMILES string tokenization, is known as template-free synthesis prediction. This method initially arose from early work on sequence based machine learning-models from the field of neural machine translation (NMT)[4,5], and resulted in the subsequent development of chemoinformatic strategies. The template-free approach works by framing synthesis prediction as a token-based language translation task where reactants, analogous to language-A, are mapped to their corresponding products, analogous to language-B. During inference, predictions are generated via the probabilistic and recurrent sampling from the set of available SMILES tokenizations, such that when presented with source reactants, the model acts to infer, token by token, the most likely identity of the respective product target. Template-free methods in synthesis prediction encompass both sequence-to-sequence (seq2seq) recurrent neural networks (RNN)[6], and more recent architectural developments such as self-attending transformer models[7]. These modelling paradigms have recently been applied to the domain of chemical synthesis prediction, and retrosynthesis prediction[8,9], demonstrating their applicability and suitability both in CASP and in the domain of de novo molecular design[10,11].

However, to approximate the underlying distribution of training data and generalize well to held out data, such probabilistic models require large datasets respective to the domain of interest. It is often the case that we are presented with a situation where data is limited or particular classes from the data is scarce, arising from a bias towards using positive reactions that are frequently used. In such limited data scenarios it has been shown that SMILES randomization, originally called SMILES enumeration[12], can be used to artificially augment the available data, resulting in increases in model performance[13–15]. Although a variety of SMILES randomization strategies have recently been employed for retrosynthesis using transformer models[15], none so far have investigated the effect of data augmentation on the level of attention mechanisms – an important, and intrinsic feature, of pattern recognition in state of the art RNN and transformer architectures respectively.

In order to exploit the attention mechanism, we propose an augmentation strategy we call Levenshtein augmentation, named after a well-known similarity metric from bioinformatic sequence alignment workflows [16] following adaptation from work in coding theory[17]. We apply this metric to mediate a balance between sequence novelty and retention of the local sub-sequence similarity between reactant and

product SMILES during generation of training pairs for the sequence-to-sequence models. We hypothesized that this would result in attentional gain and an increase in model performance under data limitations by enhancing local pattern recognition on the sub sequence level during the translation task. To investigate this hypothesis, we compare model performance when trained on canonicalized data, normal SMILES randomized data, and Levenshtein augmented data. Finally, we assess model performance on the level of full sequence accuracy and the ability of the model to capture the ground molecular truth in its top-N predictions, followed by visualization and inspection of the resulting gain in attention on the SMILES sequences.

Methods

Data and preprocessing

The data is a subset of the US patent and trade office data (USPTO)[18] as curated by Liu *et al, 2017*[8] and consists of 50,000 unique canonicalized reactions which together represents a total of 10 different reaction types. The reaction-type profile is unbalanced and biased towards heteroatom alkylation and arylation reactions, which together represent 30% of all reaction types, followed by acylation and related processes (24%), and protections and deprotections (18%). The remaining 28% of reactions represent a variety of reaction-types such as C–C bond formations, heterocycle formations, reductions, oxidations, functional group interconversion, and functional group additions.

Data is formatted according to Liu *et al* [8]. Each reaction is stripped of conditions and reagents, only reactions with atom mappings between source and target resulting in a singel atom mapped product are considered. Reactions are further filtered for a maximum of two reactants per observation, in the event that a reaction results in multiple products, each product is subset as a new observation along with all reactants. The resulting dataset is split into train, test, and validation in the ratio (8:1:1) *,* followed by the application of augmentations a) normal randomization and b) Levenshtein augmentation on the train set. This resulted in the final experimental datasets we term, *canonical*, *normal,* and *Levenshtein.* All tests during inference were performed on canonicalized SMILES of the products obtained using reactants in their canonical SMILES forms.

Levenshtein Augmentation:

Two data augmentation strategies were used, the first method - normal randomization involves a strategy in which the atom ordering of reactants and products are randomly permuted 10-fold before being converted to non-canonical SMILES using RDKit[19]. The second method, Levenshtein augmentation,

consists of 10-fold augmentation that utilizes recursive similarity selection in which local substructure between the SMILES randomized source reactants and the target product are maintained. The process is illustrated in Figure 1.

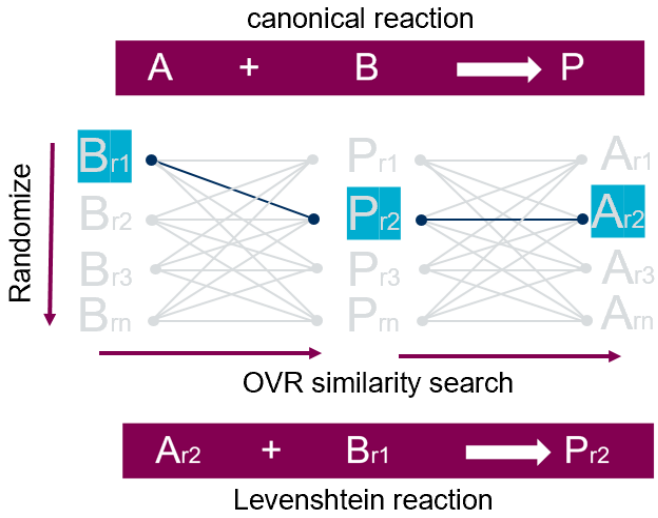

Fig 1: Schematic showing the Levenshtein method. Firstly, reactant B_{r1} is selected at random from its SMILES randomized pool, followed by $P_{r2,}$ which is selected based on its Levenshtein similarity to $B_{r1.}$ Finally, A_{r2} is likewise selected based on similarity to P_{r2}. This process is repeated N-times until the desired number of augmentations are achieved.

Levenshtein similarity computes a score based on the minimum number of insertions, deletions and additions required to change one SMILES string into another, an approach which is insensitive to differences in sequence length during scoring operations. Higher scores signify a larger Levenshtein distance and hence a greater degree of dissimilarity. We converted the score into a similarity ratio, with a value of 1 indicating an identical match between source and query sequences, and 0 signifying complete uniqueness.

Artificial Neural Networks

Transformer model

The transformer model architecture was taken from the Molecular Transformer as previously outlined [20]. The model consists of 8 attention heads in total, 4 encoding and 4 decoding layers with each layer split into discrete sub-layers, as described previously[7], and summarized as follows. The inputs are firstly transformed onto an N-sized embedding layer followed by a subsequent positional encoding, before passing through the encoding block of the transformer.

The encoder embeddings are duplicated and processed in parallel over 4 attentional heads, with key, value and query dimensions denoted as d_k, d_v, d_q, each with dimension d_{dim} *set to* 256 units. These layers were then passed into a multi-head attention block where scaled dot-product attention (eq.1) was applied to each head, followed by concatenation and linear transformation of attention outputs.

$$Attn(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad \text{(eq.1)}$$

Finally, the concatenation layer is passed through a feed-forward network followed by layer normalization (dropout rate p=0.1) and summed with the outputs of a residual connection.

The decoder embeddings are likewise split over 4 attentional heads before being concatenated at the multi-head attention sub-layer using scaled-dot product as described above. The resulting outputs are then further concatenated with the encoder block outputs and passed through a second multi-head attention sub-layer followed by concatenation and linear transformation. This concatenation layer is passed to a final feed-forward network with batch normalization (dropout rate p=0.1) and a SoftMax activation function to produce the final set of decoder outputs.

The model was trained using a batch size of 4096 over a course of 500,000 steps or until convergence. The kernel initialization method was Glorot-normal with back-propagation performed using an adaptive moment estimation (Adam) optimizer, β_1 and β_2 parameters were chosen at 0.9 and 0.998 respectively. The learning rate was set dynamical according to Noam learning rate scheduler (eq.2) initialized to 2.0 and decaying accordingly after a warm-up duration of 8000 steps. Label smoothing was not applied during training. The Molecular Transformers reporting of the top-N accuracies was used as provided when testing on the preprocessed test-set.

$$Noam\ lrate = d_{model}^{-0.5} . min\left(step_{num}{}^{-0.5}, step\_num\ .\ warmup_{steps}{}^{-1.5}\right) \quad \text{(eq.2)}$$

LSTM based Sequence-to-Sequence Model

The Seq2Seq model consists of an encoder RNN and a decoder RNN, with long short-term memory cells (LSTM). The encoding block consists of an embedding layer of 256 dimensions and 5 stacked bidirectional LSTM layers with hidden size of 512 and dropout of 0.3. The encoder block returns a tensor of encoded outputs (e_{output}) and a tuple of LSTM hidden-state and cell-state tensors of layers in the last

step across both directions([e_{hidden}, e_{cell}]).The hidden and cell states for both directions are then summed and passed to the decoder. Similarly, the encoded outputs e_{output} are summed for both directions and used to compute the attention with the output from the current step of decoder.
The decoder block consists 256 dimensions of 5 stacked unidirectional LSTM layers with hidden size of 512 and dropout of 0.3, an attention layer, a concatenation layer and a linear layer. An attention layer then computes the scaled dot product attention between the current e_{output} and the decoding h-state tensor d_{hidden} and derives a context vector which captures relevant information from every source token to help predict the current target token. The context vector and the decoder LSTM output (d_{output}) are then passed through a concatenation layer that contains a linear layer followed by a hyperbolic tangent activation function. The linear layer is applied to reshape the output from concatenation layer to the vocabulary size. Finally, a SoftMax activation function is applied to obtain the probabilities of each token.

The model was trained using a batch size of 128 over 28 epochs or until convergence. The weights initialization method was Glorot normal and back-propagation was performed using an adaptive moment estimation (Adam) optimizer with default beta and momentum terms as defined by PyTorch. The learning rate was initialized to $1e^{-4}$ and changed adaptively according to the updates of the Adam optimizer.

Results

Pairwise Sequence Similarity after Augmentation

The distribution of similarity scores between reactants and their respective products for normal randomization, and Levenshtein augmentation were measured using the similarity ratio as described above, the results are illustrated below in Figure 2. Normally randomized SMILES, as suggested by its name, resulted in a normal distribution with mean centered on a similarity score of 0.5 (Figure 2.A). Conversely, Levenshtein augmented SMILES (Figure 2.B) resulted in a higher degree of similarity, giving a uniform distribution spread over the upper range of similarity scores. The similarity scores of the two-augmentation methods compared to canonicalized SMILES demonstrate a quantifiable shift with Jenson-Shannon divergence index of 0.457 and 0.124 for the normal and Levenstein method respectively.

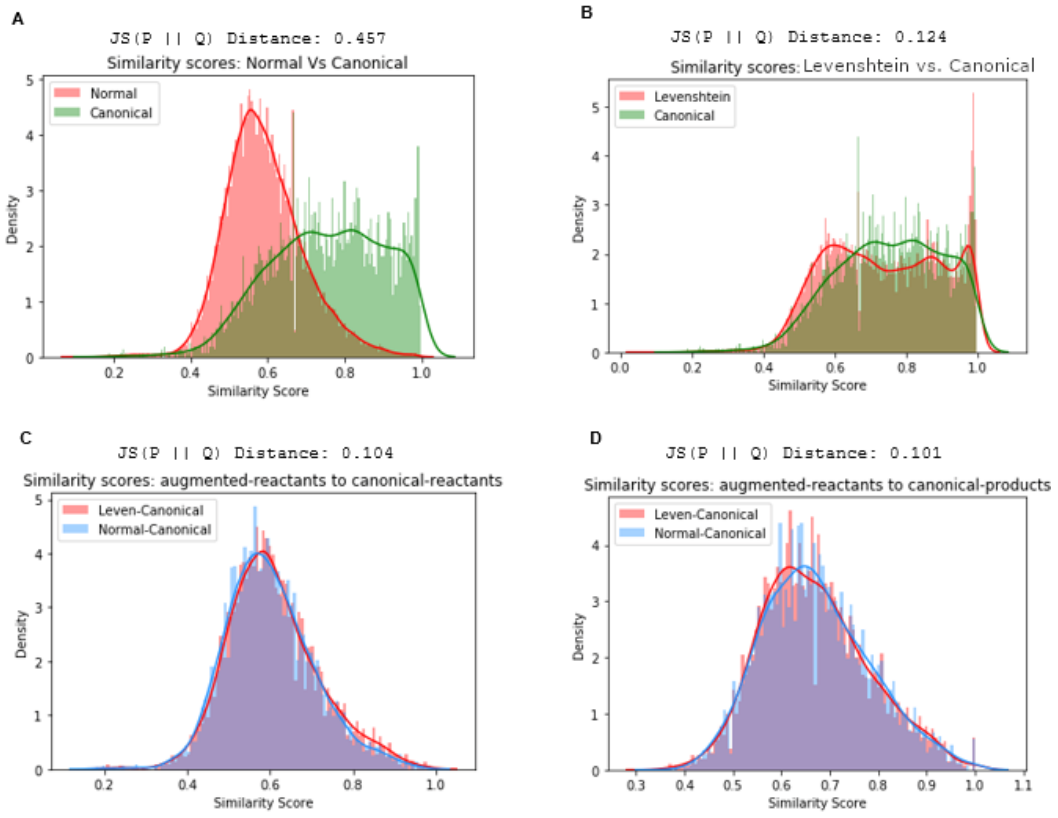

Figure 2: **A)** Comparison of reactant-product similarity score distributions between normally randomized and canonicalized SMILES. **B)** Comparison of reactant-product similarity score distributions between Levenshtein augmented and canonicalized SMILES**. C)** Shows similarity score distributions between augmented reactants and canonicalized reactants. **D)** Shows similarity score distributions between augmented reactants and canonical products.

RNN Model Performance

The RNN and Transformer model were trained on the respective augmented datasets: canonical, normal, and Levenshtein, and tested on the canonicalized test set. The results of the LSTM-based model were assessed on full sequence accuracy. Table 1 demonstrates that the application of augmentation leads to an improvement to the top-1 accuracy when assessed on the canonicalized test data for all augmentation methods. Levenshtein augmentation demonstrated the most significant increase in performance with an overall accuracy of 66.7%, which represents an increase of 7.9 percentage points in accuracy from the canonical baseline of 58.9%. Conversely the application of normal randomization resulted in an overall top-1 accuracy of 60.3%, an increase from baseline of 1.4 percentage points. Taken together these results demonstrate the significant improvements in performance of the Levenshtein augmentation method for the LSTM-based model.

Table 2: Transformer and RNN model performances across all data sets augmentations for USPTO 50K. All models were tested using the canonical test dataset, and the performance was assessed on the level of full sequence accuracy after conversion to canonical SMILES. Bold shows best per column.

	Transformer Test Performance				LSTM model Test Performance
Train set	Top-1	Top-3	Top-5	Top-10	Top-1
Canonical	41.10 %	47.30 %	48.70 %	49.90 %	58.88%
Normal	40.40 %	46.3 0%	48.40 %	50.60 %	60.30%
Levenshtein	**41.50 %**	**48.10 %**	**50.00 %**	**51.40 %**	**66.73%**

Transformer Model Performance

Next, the performance of the different randomization methods are examined for the transformer model (Table 2) by assessing the ability of the model to recover the ground truth from the n-best predictions following a beam search operation. Normal randomization resulted in a decrease in accuracy when assessed across the top-1:5 accuracies, with a slight increase in performance from a baseline of 0.7 percentage points when assessed against the top-10 accuracy. Conversely, Levenshtein augmentation demonstrated an increase in accuracy across all the top-X accuracies compared to baseline (0. 4, 0.8, 1.3 and 1.5 percentage points respectively).

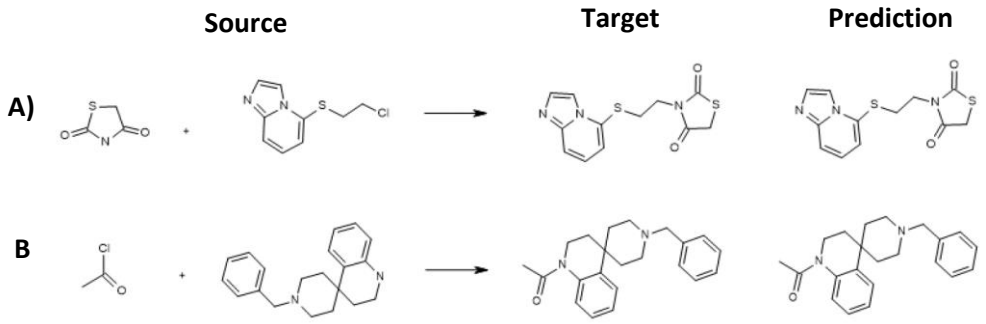

Figure 4: Two examples of reactions from the test-set and respective predictions taken from the transformer model. A) SN2 reaction B) Amide-formation.

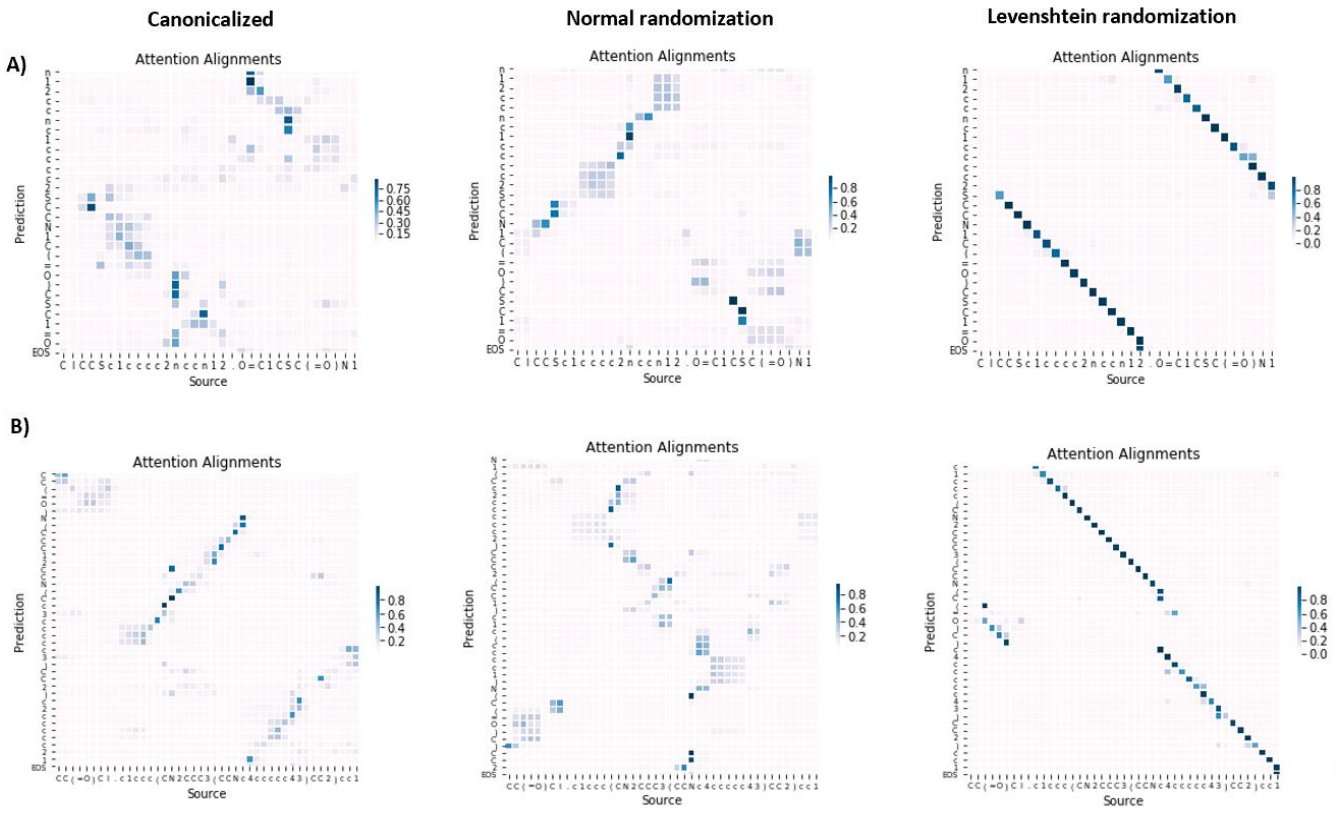

Figure 5: Transformer heatmap visualization of attention weightings across three different data sets, canonical, normal randomization, and Levenshtein augmented of the reactions **A** and **B** from Figure 4 (above). Levenshtein augmentation results in a higher degree of attention respective to raw canonicalized data, while also demonstrating less attention to off diagonal motif respective to normal randomization.

Attention Scores

To further probe these results, the average attention weights were extracted from the final multi-head attention block of the transformer to generate a set of alignment heatmaps between source reactants and inferred products (Figure 5) for the SN2 reaction and amide formation given in Figure 4. The results indicate that Levenshtein augmentation demonstrated on average, a greater degree of on-diagonal attention compared to both the normal randomized, and canonicalized data. Furthermore, Levenshtein augmentation led to the exclusion of many of the low-level attention scores, which are present to a greater extent in both the normal randomized and canonicalized attention representations.

Discussion

Augmentation

The increase in performance observed under Levenshtein augmentation may be explained by a process conceptually similar to feature disentanglement[21]. By maximizing the Levenshtein similarity between source and target within the input space, we encourage the model to isolate and attend to relevant parts of the latent representation across enumerated forms. Although these randomized SMILES modify the input space, the representations remain faithful to the underlying data structure - the molecular graph. In this respect, Levenshtein augmentation fulfils the purpose of normal-randomization, which is to abstract beyond SMILES syntax and learn the functional chemical dependencies between reaction components and increase observation availability. Furthermore, Levenstein augmentation makes it easier for the model to extract and attend to sub-sequence motifs as the translation task often regress to a direct sequence copy task of the SMILES sequence. Conversely, normal randomization does not retain sub-sequence alignment on the SMILES representation, arguably decreasing the ability of the model to attend to relevant motifs between alternative representations under data limitations. This lack of relational dependency when using normal randomization under data limitation, makes it more difficult for the network to learn the translation, this helps explain the observed reduction in performance reported in Table 2.

Intra Model Performance

Augmentation across both Levenshtein and Normal randomization led to a greater degree of model performance for the LSTM-based model comparative to the transformer model where gains where modest. It is possible that the multi-head attention averaging in the transformer resulted in attenuation of pair-matching on the local level when applying the Levenshtein method. Conversely, the direct attention alignment of the LSTM-based model is better able to preserve local pair-matching information.

A possible criticism of the here employed Levenshtein augmentation method is that the alignments employed carry a risk of leaking information about the product to the input for the algorithms. The first choice of a reactant is random, while the second choice of reactant is conditioned on the form of the product SMILES (c.f. Figure 1). This choice of second reactant can't be known a priori during the sampling or prediction phase and may be suboptimal when unbiased product generation is wanted. A better Levenshtein augmentation algorithm for unbiased prediction may be to select both reactants by random, and then select the most similar alignment product using the Levenshtein scores. We did however test the trained models on the canonical dataset with canonical SMILES which is not biased in this way. The

test dataset reactant pairs SMILES form may thus not match up with a given product similar to the matches in the training set, which would more likely lead to diminished performance. However, the distributions of similarities for both the canonical and Levenshtein SMILES pairs have similar distributions (c.f. Figure 2), and the model may still be within its applicability domain even given reactants as canonical SMILES. Thus, this does not change the conclusion that Levenshtein augmentation performs better than both canonical and normal randomization approaches.

Inter model Performance

The disparity in absolute performance observed between the transformer- and LSTM-models may result from the transformer being ill suited for small datasets, here we use USPTO-50K which is a subset of, and considerably smaller than the MIT-USPTO data benchmarked against in previous studies. Furthermore, the transformer was used out-of-box according to the hyperparameter and architectural configurations of those determined by the Molecular Transformer[20] which were trained on a much larger dataset. Conversely, the LSTM-based model was adapted from previous internal work and it's hyperparameters tuned to the USPTO-50K dataset. However, the intention herein was not to benchmark inter-model performance, but intra-model performance with respect to the data augmentation techniques.

Comparison to literature

SMILES randomization has been proven to increase the performance of both retrosynthesis prediction, synthesis predictions and de-novo molecular generation[10,11,14]. In the case of retrosynthesis, Tetko *et al*[15], employed several means of randomization with a transformer model on the MIT-USPTO dataset. Randomization to products with an permutation factor of 1 (N=1) resulted in an increase in top-1 model performance by 2.8 percentage points compared to the baseline canonical model. Shuffling the order of reactants in the training set with permutation factor N=20 resulted in an increase from baseline of 11.2 percentage points. for top-1 accuracy, and an increase of approximately 12.5 percentage points for top-10 accuracy comparative to the canonical baseline.

In the forward synthesis case, a comparison to baseline is lacking, however an accuracy of 91.9% for top-1 was reported when using an augmentation factor N=100 and beam width of 10 - reflecting a similar performance reported by the molecular transformer by Schwaller *et al*[20]. Similarly, the molecular transformer benefited from randomization of SMILES using an augmentation factor N=1, which resulted in an increase of 0.8% from the canonical baseline model.

The difference in the performance between the Tetko and Schwaller models, and the model performance presented in this study could be because we apply training to a subset of the MIT-UPSTO dataset (USPTO

-50K) while the former models train on the full MIT-USPTO dataset. This difference in dataset size results in significantly fewer unique SMILES reactions available during training of our models. In the case of Tetko, the MIT-USPTO dataset is further augmented by an permutation factor N=100, while in this work we limit the permutation factor to N=10.

Outlook

An interesting outlook is to explore the differences between each augmentation strategy, canonical forms, and the corresponding relationship within latent space representations. This experiment focuses on the pairwise similarity within the input space, however the assumption that this relationship is preserved within the network latent space cannot be guaranteed. To investigate whether this is the case, it could be possible to quantify the latent space cosine similarity between canonical reaction SMILES and augmented counterparts, both on the full reaction level and on reaction sub-component level. The notion of input-latent space consolidation under data augmentation has previously been explored from work on chemical heteroencoders[13] and more generally as a means of contrastive self-supervised learning methods in computer vision[24]. Furthermore recent work[25] has demonstrated that attention weights in synthesis prediction may provide an unsupervised method for atom mapping between reactants and products. An interesting outlook would be to assess if unsupervised atom map extraction as described above, can be improved by utilization of Levenstein augmentation.

Conclusions

We conclude that Levenshtein augmentation, an augmentation technique which acts to balance SMILES sequence novelty with retention of local sub-sequence alignment, results in an increase in model performance over normal randomization, which acts via SMILES randomization alone. Improvement is more apparent for the seq2seq LSTM-based model over the transformer where improvements are more modest. We assess augmentation methods for model biasing across several cases, both in terms of similarity scores on the SMILES component level and on the full reaction level where we conclude performance is not based on train-test mappings between reactants and products. A comparison to the SMILES string components is then made to confirm that neither method is unintentionally resulting in canonical SMILES during the generation of augmented sets, which may otherwise result in bias caused by covariance shift during training and testing.

The results of this experiment suggest that augmentation has greater potential when applied to RNN architectures over transformers, a difference which may be explained by multi-head attention averaging with respect to the transformer mechanism. This averaging may result in a global SMILES application of

attention, comparative to the LSTM-based model which does not split attention over multiple heads and therefore applies attention instead on a local sub-sequence level. Therefore, the LSTM-based model is better able to take advance of the retention of locality alignment which is ultimately, the basis for the Levenshtein augmentation method.

Finally, we show that there is a qualitative difference on the level of attention alignment scores using Levenshtein augmentation compared to normal SMILES randomization based augmentation, offering a mechanistic insight into the observed increase in performance under data augmentation. Together these results build confidence in the Levenshtein augmentation method for data augmentation in SMILES based deep learning reaction informatics.

Declarations

Availability of Data and Materials

All code used in the production of this work will be made available under an MIT license at: https://github.com/MolecularAI

Competing Interests

The authors declare that they have no competing interests.

Funding

Amol Thakkar was supported financially by the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement no. 676434, "Big Data in Chemistry" ("BIGCHEM," http://bigchem.eu).

Acknowledgements

The authors would like to thank the Molecular AI group at AstraZeneca for their support.

References

1. Corey, E. J., Long, A. K. & Rubenstein, S. D. Computer-assisted analysis in organic synthesis. *Science* **228**, 408–418 (1985).
2. Kayala, M. A., Azencott, C.-A., Chen, J. H. & Baldi, P. Learning to predict chemical reactions. *J Chem Inf Model* **51**, 2209–2222 (2011).
3. Segler, M. H. S., Preuss, M. & Waller, M. P. Planning chemical syntheses with deep neural networks and symbolic AI. *Nature* **555**, 604–610 (2018).

4. Schwaller, P., Gaudin, T., Lányi, D., Bekas, C. & Laino, T. “Found in Translation”: predicting outcomes of complex organic chemistry reactions using neural sequence-to-sequence models. *Chemical Science* **9**, 6091–6098 (2018).
5. Nam, J. & Kim, J. Linking the Neural Machine Translation and the Prediction of Organic Chemistry Reactions. *arXiv:1612.09529 [cs.LG]* 1–19 (2016).
6. Bahdanau, D., Cho, K. & Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv:1409.0473 [cs, stat]* (2016).
7. Vaswani, A. *et al.* Attention is All you Need. in *Advances in Neural Information Processing Systems 30* (eds. Guyon, I. et al.) 5998–6008 (Curran Associates, Inc., 2017).
8. Liu, B. *et al.* Retrosynthetic Reaction Prediction Using Neural Sequence-to-Sequence Models. *ACS Cent. Sci.* **3**, 1103–1113 (2017).
9. Karpov, P., Godin, G. & Tetko, I. V. A Transformer Model for Retrosynthesis. in *Artificial Neural Networks and Machine Learning – ICANN 2019: Workshop and Special Sessions* (eds. Tetko, I. V., Kůrková, V., Karpov, P. & Theis, F.) vol. 11731 817–830 (Springer International Publishing, 2019).
10. Sattarov, B. *et al.* De Novo Molecular Design by Combining Deep Autoencoder Recurrent Neural Networks with Generative Topographic Mapping. *J. Chem. Inf. Model.* **59**, 1182–1196 (2019).
11. Gómez-Bombarelli, R. *et al.* Automatic chemical design using a data-driven continuous representation of molecules. *ACS Cent. Sci.* **4**, 268–276 (2018).
12. Bjerrum, E. J. SMILES Enumeration as Data Augmentation for Neural Network Modeling of Molecules. *arXiv:1703.07076 [cs.LG]* (2017).

13. Bjerrum, E. J. & Sattarov, B. Improving Chemical Autoencoder Latent Space and Molecular De Novo Generation Diversity with Heteroencoders. *Biomolecules* **8**, 131 (2018).
14. Kimber, T. B., Engelke, S., Tetko, I. V., Bruno, E. & Godin, G. Synergy Effect between Convolutional Neural Networks and the Multiplicity of SMILES for Improvement of Molecular Prediction. *arXiv:1812.04439 [cs, stat]* (2018).
15. Tetko, I., Karpov, P., Deursen, R. & Godin, G. Augmented Transformer Achieves 97% and 85% for Top5 Prediction of Direct and Classical Retro-Synthesis. *arXiv:2003.02804 [cs.LG]* (2020).
16. Levenshtein Distance - an overview | ScienceDirect Topics. https://www.sciencedirect.com/topics/computer-science/levenshtein-distance.
17. Levenshtein, V. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady* **10**, 707–710 (1966).
18. Lowe, D. Chemical reactions from US patents (1976-Sep2016). (2017) doi:10.6084/m9.figshare.5104873.v1.
19. *RDKIT: Open source cheminformatics*.
20. Schwaller, P. *et al.* Molecular Transformer: A Model for Uncertainty-Calibrated Chemical Reaction Prediction. *ACS Cent. Sci.* **5**, 1572–1583 (2019).
21. Higgins, I. *et al.* Towards a Definition of Disentangled Representations. *arXiv:1812.02230 [cs, stat]* (2018).
22. Arús-Pous, J. *et al.* Randomized SMILES strings improve the quality of molecular generative models. *Journal of Cheminformatics* **11**, 71 (2019).

24. Chen, T., Kornblith, S., Norouzi, M. & Hinton, G. A Simple Framework for Contrastive Learning of Visual Representations. *arXiv:2002.05709 [cs, stat]* (2020).
25. Schwaller, P., Hoover, B., Reymond, J.-L., Strobelt, H. & Laino, T. Unsupervised Attention-Guided Atom-Mapping. *Chemrxiv* (2020) doi:10.26434/chemrxiv.12298559.v1.