doi.org/10.26434/chemrxiv.12339368.v1

SMILES Pair Encoding: A Data-Driven Substructure Tokenization Algorithm for Deep Learning

Xinhao Li, Denis Fourches

Submitted date: 20/05/2020 • Posted date: 21/05/2020

Citation information: Li, Xinhao; Fourches, Denis (2020): SMILES Pair Encoding: A Data-Driven Substructure Tokenization Algorithm for Deep Learning. ChemRxiv. Preprint.
https://doi.org/10.26434/chemrxiv.12339368.v1

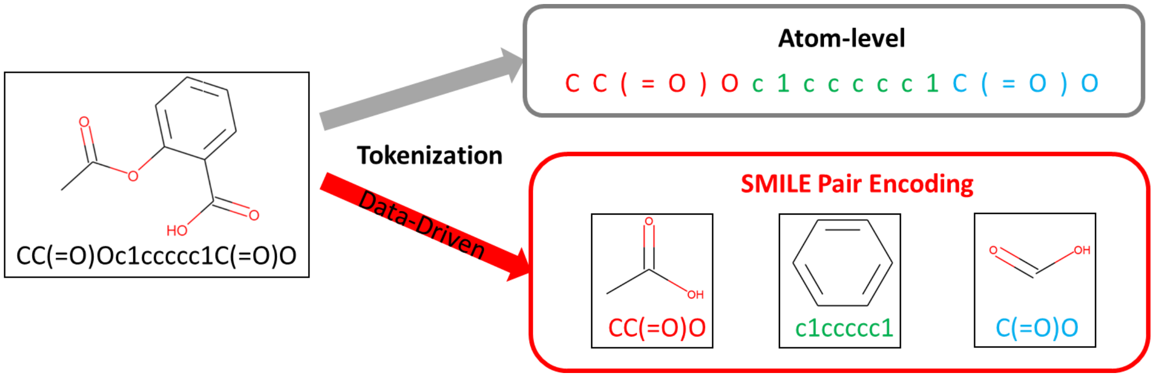
SMILES-based deep learning models are slowly emerging as an important research topic in cheminformatics. In this study, we introduce SMILES Pair Encoding (SPE), a data-driven tokenization algorithm. SPE first learns a vocabulary of high frequency SMILES substrings from a large chemical dataset (e.g., ChEMBL) and then tokenizes SMILES based on the learned vocabulary for deep learning models. As a result, SPE augments the widely used atom-level tokenization by adding human-readable and chemically explainable SMILES substrings as tokens. Case studies show that SPE can achieve superior performances for both molecular generation and property prediction tasks. In molecular generation task, SPE can boost the validity and novelty of generated SMILES. Herein, the molecular property prediction models were evaluated using 24 benchmark datasets where SPE consistently either did match or outperform atom-level tokenization. Therefore SPE could be a promising tokenization method for SMILES-based deep learning models. An open source Python package SmilesPE was developed to implement this algorithm and is now available at https://github.com/XinhaoLi74/SmilesPE.

SMILES Pair Encoding: A Data-Driven Substructure Tokenization Algorithm for Deep Learning

Xinhao Li & Denis Fourches*

Department of Chemistry, Bioinformatics Research Center, North Carolina State University, Raleigh, NC 27695, United States.

* *To whom correspondence should be sent. Email: dfourch@ncsu.edu*



Abstract

SMILES-based deep learning models are slowly emerging as an important research topic in cheminformatics. In this study, we introduce SMILES Pair Encoding (SPE), a data-driven tokenization algorithm. SPE first learns a vocabulary of high frequency SMILES substrings from a large chemical dataset (e.g., ChEMBL) and then tokenizes SMILES based on the learned vocabulary for deep learning models. As a result, SPE augments the widely used atom-level tokenization by adding human-readable and chemically explainable SMILES substrings as tokens. Case studies show that SPE can achieve superior performances for both molecular generation and property prediction tasks. In molecular generation task, SPE can boost the validity and novelty of generated SMILES. Herein, the molecular property prediction models were evaluated using 24 benchmark datasets where SPE consistently either did match or outperform atom-level tokenization. Therefore SPE could be a promising tokenization method for SMILES-based deep learning models. An open source Python package *SmilesPE* was developed to implement this algorithm and is now available at https://github.com/XinhaoLi74/SmilesPE.

1. Introduction

Over the past few years, the cheminformatics community has witnessed dramatic advances in using deep learning neural networks (DLNN) to tackle challenging tasks ranging from molecular property prediction[1–5] to *de novo* molecular generation and optimization[6–8]. The success of deep learning techniques in natural language processing (NLP) makes the use of text-based molecular representations an attractive research area[9]. A vital part of processing text-based chemical representations for deep learning models is how to break them into a sequence of standard units (or 'tokens'), a process called tokenization. The tokens are supposed to contain the essential structural information that are able to reliably and consistently characterize the molecules. Through the specific type of neural network architectures such as recurrent neural network (RNN)[10], convolutional neural network (CNN)[11] or Transformer[12], the models will process the tokens and learn how to extract the useful knowledge for solving chemical problems.

In that context, SMILES[13,14] (Simplified Molecular Input Line Entry System) is the most popular text representation of chemicals; it encodes a molecular graph as a sequence of characters. One approach of SMILES tokenization is to simply break the SMILES character by character (character-level tokenization). One issue of character-level tokenization is that some chemically meaningful information of a single atom is represented by multiple characters and may thus result in ambiguous meanings. With character-level tokenization, '[C@@H]' is tokenized into six characters '[', 'C', '@', '@', 'H' and ']' even though it encodes the stereochemistry of a carbon atom. The token 'C' can be the symbol of carbon or part of the symbol of chlorine ('C' and 'l'). Atom-level tokenization is a more commonly used tokenization method that follows the character-level tokenization with some modifications to ensure atoms are extracted as tokens: (1) multicharacters element symbols such as ‘Cl’ and ‘Br’ are considered as individual tokens; (2) special characters encoded between brackets are considered as tokens (*e.g.*, ‘[nH], ‘[O-]’ and ‘[C@]’).

In this study, we proposed the SMILES Pair Encoding (SPE) method, a data-driven substructure tokenization algorithm for deep learning applications. The SPE is inspired by the byte pair encoding (BPE) algorithm[15], one major tokenization method in NLP. BPE was initially developed as a data compression algorithm and further adopted as a subword tokenization algorithm. The core idea of BPE tokenization is to keep the more frequent words as unique tokens whereas less frequent words will be further decomposed into subword units. Similar to BPE, SPE identifies and keeps the frequent SMILES substrings as unique tokens. Starting from atom-level tokens, SPE generates the SMILES substring tokens by iteratively merging the high frequency token pairs from a large chemical dataset. SPE enhances the widely used atom-level tokenization in two major aspects:

1. **Chemically meaningful substructures**: SPE ensures that the most common SMILES substrings will be represented as unique tokens. The SMILES substrings encode molecular substructures which contain richer information and better reflect the molecular functionalities compared to the atom-level tokens.
2. **Shorter input for deep learning models**: The input token sequences from SPE are shorter than those from atom-level tokenization. The shorter input can reduce the computational cost and accelerate DLNN model training. In addition, it will relieve the burden of long-range dependencies[16] for RNN-based models which usually caused by the exploding and vanishing gradients.

Herein, we performed two case studies to showcase the potential of SPE in both molecular generation and molecular property prediction. The goal of these case studies is to evaluate whether

SPE tokenization could represent a better alternative to the atom-level tokenization. We demonstrate that for both generative and predictive (QSAR) tasks, the SPE tokenization shows superior performances compared to the atom-level tokenization. In the molecular generation case study, we trained RNN-based language models (LM) with SPE and atom-level tokenization, respectively. One common issue of LM-based molecular generative models is the low validity rates of generated SMILES. Our result shows that SPE tokenization significantly improves the validity rate compared to the atom-level tokenization. In the second case study, we compared the two tokenization methods using 24 datasets for QSAR modeling purposes. The SPE did achieve better or comparable prediction performances for 23 out of the 24 datasets and on average, did offer a five times speed up for model training.

Our major contributions of this study are:

1. Propose a new SMILES tokenization algorithm that would be useful for a wide range of cheminformatics DLNN modeling tasks;
2. Develop an open source Python package *SmilesPE*, which enables the training of SMILES pair encoding on a large dataset and the use of trained SPE vocabulary to tokenize SMILES for deep learning applications. SmilesPE is freely available at https://github.com/XinhaoLi74/SmilesPE and can be installed via pip.

2. Method

2.1. SMILES Pair Encoding

The SMILES pair encoding algorithm consists of two major steps: the vocabulary training step which learns the high frequency SMILES substrings from a large chemical dataset and the tokenization step which applies the trained vocabulary to new SMILES, returning a sequence of tokens. In this section, we describe how to train a SPE vocabulary and how to use the trained vocabulary to tokenize SMILES for deep learning.

A SMILES Pair Encoding (SPE) vocabulary is trained according to the following steps:

- **Step 1**: Tokenize SMILES from a large dataset (e.g., ChEMBL[17]) at atom-level;
- **Step 2**: Initialize the vocabulary with all unique tokens;
- **Step 3**: Iteratively count the occurrence of all token pairs in the tokenized SMILES, merge the most frequent occurring token pair as a new token and add it to the vocabulary. This step will stop when one of the conditions is met: (1) A desired vocabulary size is achieved or (2) No pair of tokens affords a frequency larger than a given frequency threshold. The maximum vocabulary size and frequency threshold are hyperparameters for training SMILES pair encoding.

After training the SPE vocabulary, we can then tokenize SMILES based on the trained vocabulary. The SMILES substrings in the trained vocabulary are ordered by their frequency. During the tokenization process, each SMILES string is first tokenized at atom-level. SPE can then iteratively check the frequency of all pairs of tokens and merge the pair of tokens that have the highest frequency count in the trained SPE vocabulary until no further merge operation can be conducted.

It is worth noting that the proposed algorithm can also be applied to other popular text-based representation of chemicals for DLNN applications such as DeepSMILES[18] and SELFIES[19]. DeepSMILES is a variant of SMILES which changes the way of representing branches and rings. It has the same atom-level characters as SMILES. Moreover, SELFIES represents all information of a molecular graph (atoms, bonds, branches and rings) as characters in brackets. These characters can be directly recognized as tokens by the atom-level tokenization we used. As a result, we can train a specific SPE vocabulary dedicated for DeepSMILES or SELFIES without any modification.

2.2. Dataset Preparation

ChEMBL25[17] was used to train the SPE vocabulary and language models. The QSAR benchmark datasets were taken from a previous study by Cortés-Ciriano et.al[20] that include the curated pIC50 values for 24 protein targets. All molecules were standardized with the following steps using MolVS[21] and RDKit[22] packages in Python: (1) Sanitizing with RDKit; (2) Replace all atoms with the most abundant isotope for that element; (3) Remove counterions in the salts and neutralize the molecules; (4) Remove the mixtures. The canonical SMILES were then generated for modeling. After curation, about 1.7 million ChEMBL25 SMILES did remain. The QSAR benchmark datasets are summarized in **Table 1**.

Table 1. Summary of QSAR benchmark.

Targets	**Number of Molecules***
A2a	199
Dopamine	469
Dihydrofolate	573
Carbonic	591
ABL1	755
opioid	777
Cannabinoid	1,086
COX-1	1,306
Monoamine	1,307
LCK	1,336
Glucocorticoid	1,387
Ephrin	1,507
Caspase	1,584
Coagulation	1,591
Estrogen	1,622
B-raf	1,717
Glycogen	1,724
Vanilloid	1,761
Aurora-A	2,084
JAK2	2,388

COX-2	2,759
Acetylcholinesterase	2,966
erbB1	4,742
HERG	5,010

* Sorted from small to large

2.3. Machine Learning

The molecular generation was formulated as a language modeling task[23]. The RNN-based language models were trained using a large chemical data set to predict the next token t_{i+1} given a sequence of tokens {t_1, t_2, …., t_i} preceding it. The models learn a probability distribution of the training molecules and can then sample from the learned distribution to generate new molecules.

The QSAR models were developed using the MolPMoFiT framework from our previous study[24]. MolPMoFiT is an effective transfer learning method for QSAR modeling, which uses the chemical language model pre-training + task-specific fine-tuning strategy[25] to enable the knowledge learned from the large unlabeled chemical data to be transferred to smaller supervised datasets.

2.4. Evaluation Metrics

2.4.1. Evaluation Metrics for Generative Models

- **Validity**: the percentage of generated SMILES that can be converted to valid molecules;
- **Novelty**: the percentage of valid molecules that are not included in the training set;
- **Uniqueness**: the percentage of valid molecules that are unique.

2.4.2. Evaluation Metrics for QSAR Models

All 24 QSAR benchmark datasets correspond to regression tasks. The root-mean-square-error (RMSE), coefficient of determination (R^2) and mean absolute error (MAE) were used as evaluation metrics for the regression models.

Cohen's d[26] (eq 1) measures the relative performances of two methods. The $\bar{x}_1$ and $\bar{x}_2$ are the mean values for each group of results. The SD_1 and SD_2 are the standard deviations for each group of results. A positive *d* value means method 1 has a larger mean than method 2 while a negative *d* value means method 1 has a smaller mean than method 2. The thresholds of small, medium and large effects are set to 0.2, 0.5 and 0.8 as recommended[26,27]. The effect with a |*d*| (absolute value of *d*) less than 0.2 as no difference; between 0.2 and 0.5 as of minor difference; between 0.5 and 0.8 as medium difference; greater than 0.8 as large difference. In the following analysis, method 1 references as SPE tokenization and method 2 references as atom-level tokenization.

$$\text{Cohen's } d = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{(SD_1^2 + SD_2^2)/2}} \qquad (1)$$

2.5. Experiments

Training a SPE vocabulary. SPE is a data-driven algorithm so that both the quality and quantity are crucial. SMILES augmentation[24,28–31] is widely used as a data augmentation technique in deep learning applications. In order to capture common SMILES substrings in both canonical and non-canonical SMILES, we generated one non-canonical SMILES for each canonical SMILES in the curated ChEMBL dataset. As a result, 3.4M SMILES were obtained for training the SPE vocabulary. The maximum vocabulary size was set to 30,000 and the frequency threshold is set to 2000 to ensure the common SMILES substrings can be included in the vocabulary.

Language models. We trained two language models with SPE tokenization and atom-level tokenization, respectively. 9 million SMILES (1 canonical + 5 non-canonical SMILES for each compound) generated from the curated ChEMBL25 dataset are used for the model training. The model architecture we choose for language modeling is AWD-LSTM[32] (ASGD Weight-Dropped

LSTM), a variant of LSTM (long short-term memory) models that is enhanced with various kinds of dropouts and regularizations. Specially, dropouts are applied to embedding layer, input layer, weights and hidden layers. It has shown strong performances on language modeling in NLP. We choose the same model architecture and hyperparameters as our previous study[24]. The models have an embedding layer with a size of 400, three LSTM layers which 1152 hidden units per layer, and a softmax layer. We apply embedding dropout of 0.1, input dropout of 0.6, weight dropout of 0.5 and hidden dropout of 0.2. Both models are trained with a base learning rate of 0.008 for 10 epochs using one cycle policy[33].

Molecular Generation and Evaluation. For each language model, ten sampled sets of 1,000 SMILES strings were generated and evaluated with validity, novelty, and uniqueness. The validation of generated SMILES is evaluated by RDKit.

QSAR models. The QSAR models were fine-tuned on the pre-trained language models following the procedure of MolPMoFiT. All the models were tuned with base learning rates and training epochs on the validation sets and evaluated on the test sets on ten random 80:10:10 splits. SMILES augmentation was applied as descripted in our previous study[24]. During training, the SMILES of training sets were augmented 25 times and the SMILES of validation sets were augmented 15 times. Test time augmentation (TTA) was applied to compute the final predictions: for each compound, the final prediction is generated by averaging predictions of the canonical SMILES and four augmented SMILES.

2.6. Implementation

The SPE algorithm was implemented in Python. We implemented machine learning models using PyTorch[34], fastai[35] and MolPMoFiT. The MolPMoFiT code is available at https://github.com/XinhaoLi74/MolPMoFiT.

3. Result and Discussion

3.1. SMILES Pair Encoding on ChEMBL

A dataset with ~3.4 million SMILES generated from the curated ChEMBL dataset, containing both canonical and non-canonical SMILES, was used to train a SPE vocabulary. The trained SPE vocabulary contained 3,002 unique SMILES substrings with length ranges from 1 to 22 (**Figure 1**). The length was computed by counting the number of atom-level characters in the SMILES substrings. As shown in **Figure 2**, the SMILES substrings are human-readable and mostly correspond to chemically meaningful substructures and functional groups. The full list of SPE vocabulary can be found in the project GitHub repository. Some machine learning architectures[36] and techniques[31,37] can interpret the model predictions by computing the importance/contribution scores of the input tokens. In this regard, the SMILES substrings are more interpretable than individual atom characters.

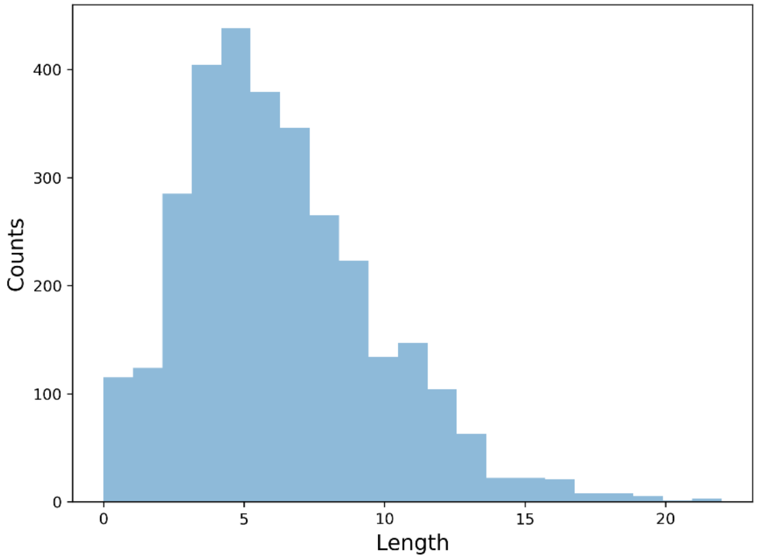


Figure 1. Distribution of length of SMILES Pair Encoding substrings trained on ChEMBL.

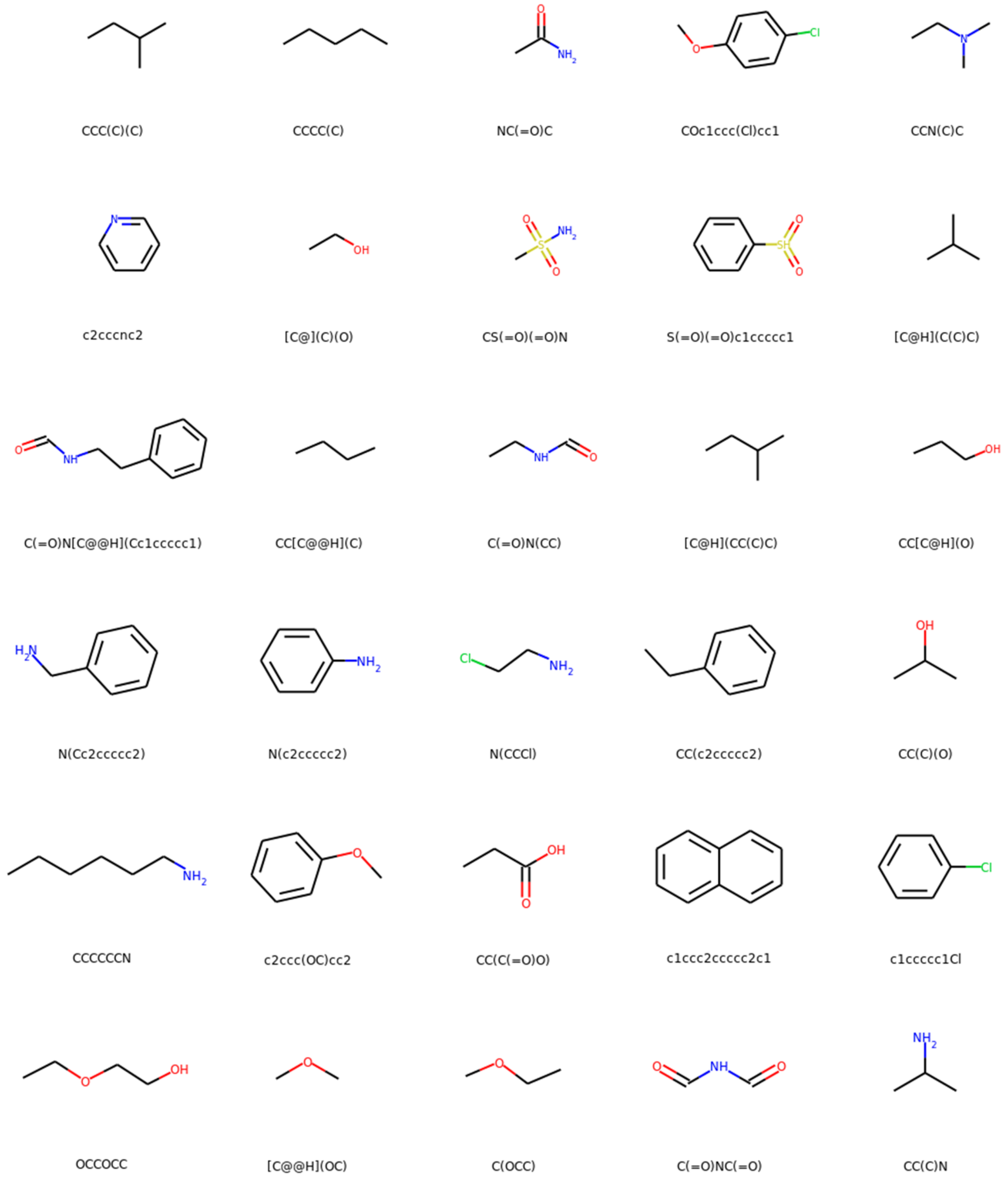


Figure 2. Representative SPE fragments.

Table 2 shows some examples of tokenized SMILES from SPE. Compared to atom-level tokenization, SPE provides a more compact representation of SMILES for deep learning models. **Figure 3** shows the results of SPE and atom-level tokenization on the ChEMBL dataset. The SPE tokenization has a mean length of approximately 6 tokens while the atom-level tokenization has a mean length of approximately 40. Such shorter input sequences can dramatically benefit DLNN models in different aspects. Due to the sequential nature of RNN-based models, they require longer training time and suffer long-term dependencies in case of long input sequences. The newer Transformer models[12] replace the recurrent components with attentions, which makes them no longer suffer the long-term dependency issue. However, the computational cost will scale quadratically with the length of input sequence due to the mechanism of self-attention. As a result, for the same deep learning application, SPE can save the computational cost and accelerate the training and inference processes.

Table 2. Example of Tokenized SMILES.

SMILES	**Tokenized SMILES Substrings**
CC(CCCCC(=O)Nc1ccc(C(F)(F)F)cc1)NCC(O)c1cccc(Cl)c1	'CC(', 'CCCC', 'C(=O)Nc1ccc(', 'C(F)(F)F)cc1)', 'N', 'CC(O)', 'c1cccc(Cl)c1'
CCC(O)(C(=O)Nc1ccccc1Cl)C(F)(F)F	'CCC(O)(', 'C(=O)N', 'c1ccccc1Cl)', 'C(F)(F)F'
O=C1CS/C(=N/N=C\c2ccco2)N1Cc1ccccc1	'O=C1', 'CS', '/C(', '=N/N', '=C\\', 'c2ccco2)', 'N1', 'Cc1ccccc1'

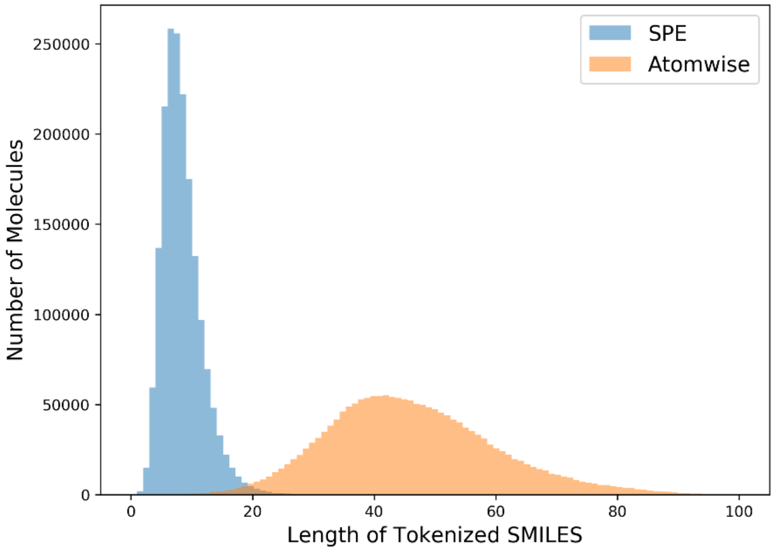


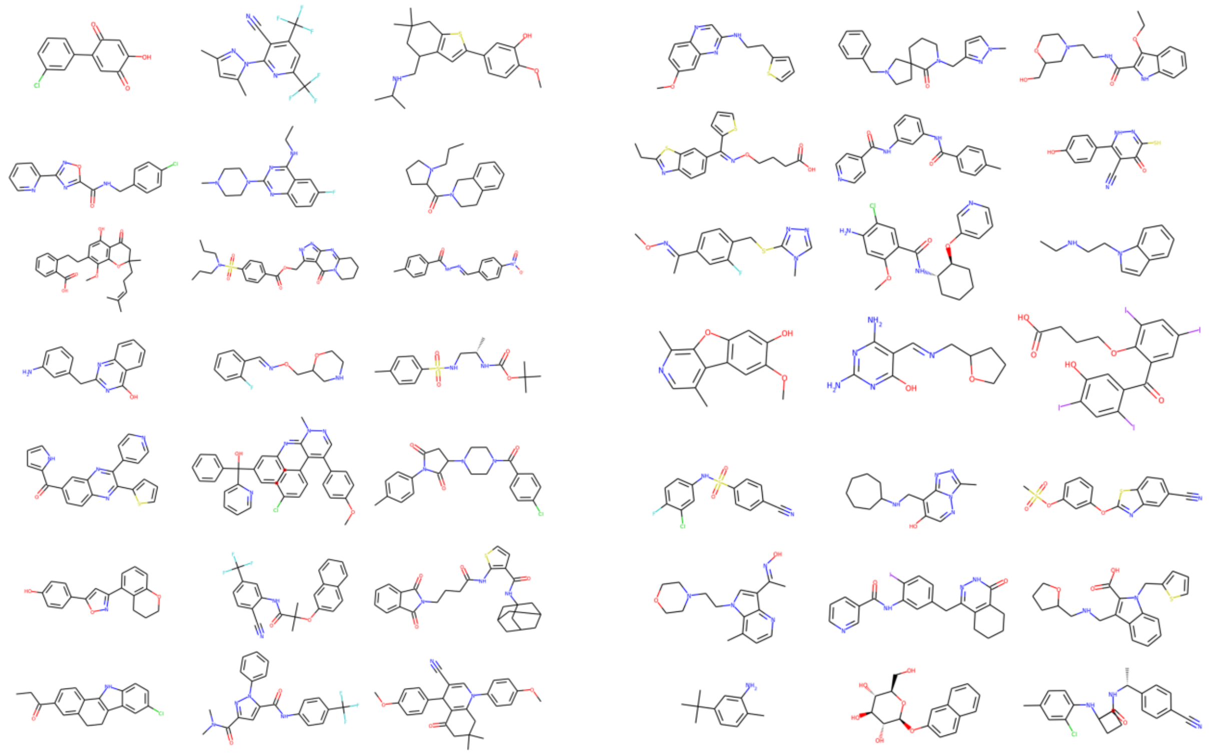
Figure 3. Distribution of length of tokenized SMILES of ChEMBL. Blue: SMILES Pair Encoding tokenization; Orange: Atom-level tokenization.

3.2. Molecular Generation Case Study

We evaluated the performance of SPE versus atom-level tokenization on molecular generation using an RNN-based language model architecture described in the **Experiments Section**. The models were trained using 9 million SMILES (1 canonical + 5 non-canonical SMILES for each compound) generated from the curated ChEMBL25 dataset. We compared the validity, novelty and uniqueness of ten sets of 1,000 sampled SMILES from each model. The results are summarized in **Table 3**. The model trained with atom-level tokenization can only produce 58.1% valid SMILES whereas the model trained with SPE tokenization can produce 93.1% valid SMILES. The invalidation of SMILES-based generative models is mainly due to the constraints of SMILES syntax: (1) the missing of ring or branch closures; (2) wrong bond numbers of atoms. Instead of generating a molecule atom-by-atom, the model trained with SPE tokenization uses a fragment-by-fragment approach, which is naturally more error proofing. In addition, SPE tokenization also has a higher novelty compared to the atom-level tokenization (97.3% vs. 96.7%). Both models can generate 100% unique molecules. **Figure 4** shows some sampled molecules.

Table 3. Metrics for Molecular Generation.

	SMILES Pair Encoding	Atom-level
Validity	**0.931 ± 0.008**	0.581 ± 0.011
Novelty	**0.973 ± 0.006**	0.967 ± 0.006
Uniqueness	1.0	1.0



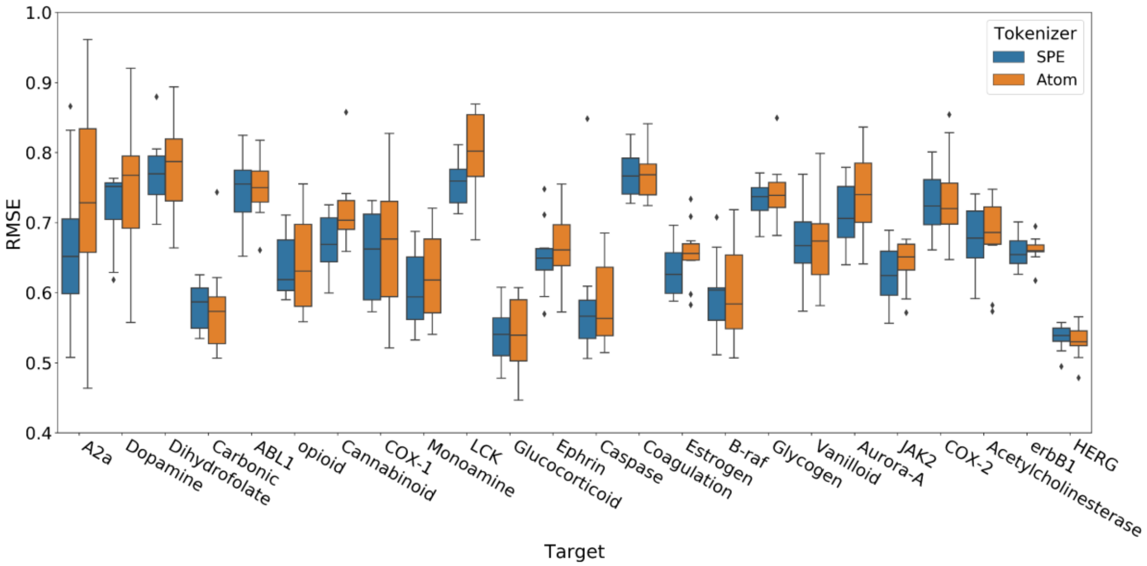
(a) SMILES Pair Encoding

(b) Atom Level

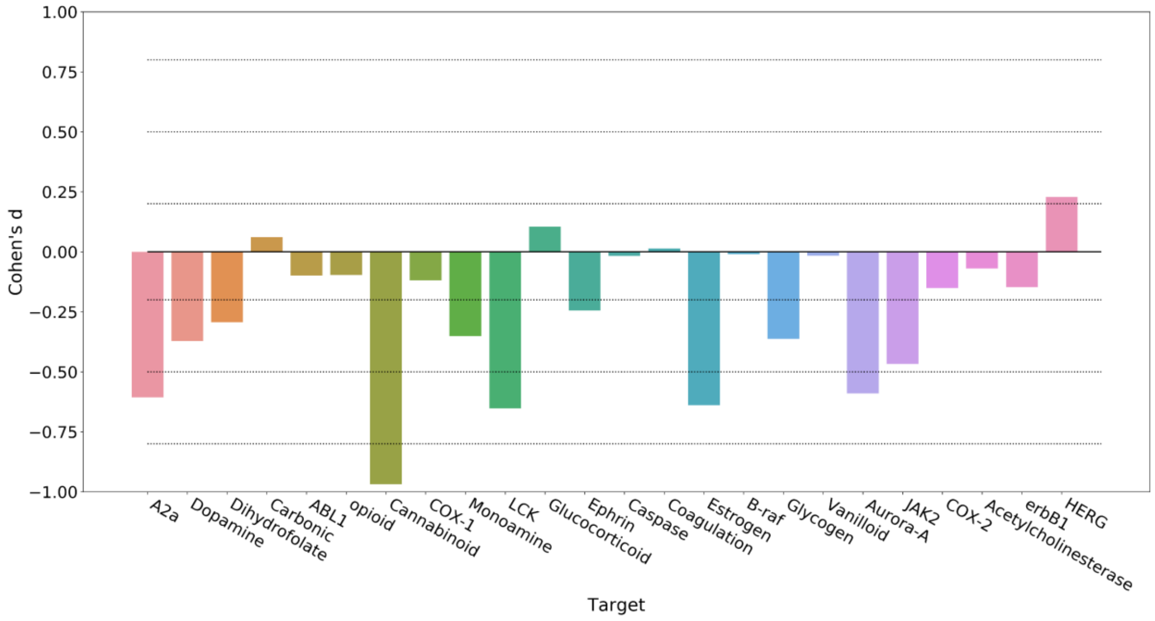
Figure 4. Random sampled examples of Generated Molecules. (**a**) examples from the model trained with SMILES Pair Encoding tokenization; (**b**) examples from the model trained with atom level tokenization

3.3. Molecular Property Prediction Case Study

We also compared the performances for molecular property prediction models trained with the two tokenization methods on 24 regression datasets (pIC50). The models were evaluated on ten 80:10:10 random splits. RMSE (**Figure 5a**), R^2 and MAE (**Tables S1** and **S2**) were used as evaluation metrics. The Cohen's d was used to measure the relative performance of the two methods (**Figure 5b**). The thresholds of small, medium and large effects were set to 0.2, 0.5 and 0.8 as recommended[26,27]. As shown in **Figure 5**, models trained with SPE tokenization showed comparable or better performances for 23 out of 24 datasets compared to those trained with atom-level tokenization. Specifically, SPE tokenization showed a large effect on Cannabinoid and medium effect on A2a, LCK, Estrogen and Aurora-A. In addition to the strong performances, the models with SPE were trained on average 5 times faster due to the shorter input sequence.



(**a**)



(**b**)

Figure 5. Results of QSAR benchmark. (**a**) Test set RMSE (**b**) The effect size (Cohen's *d* value) of difference between models trained with SPE tokenization and atom-level tokenization. A positive *d* value means atom-level tokenization performances better than SPE tokenization. A negative *d* value means SPE tokenization performances better than atom-level tokenization. The size effect with a |*d*| (absolute value of *d*) less than 0.2 as no difference; between 0.2 and 0.5 as of minor difference; between 0.5 and 0.8 as medium difference; greater than 0.8 as large difference.

Conclusion

In this study, we proposed SMILES Pair Encoding (SPE), a data-driven substructure tokenization algorithm for deep learning. SPE learns a vocabulary of high frequency SMILES substrings from ChEMBL and then tokenizes new SMILES into a sequence of tokens for deep learning models. SPE splits SMILES into human-readable and chemically explainable substrings and shows superior performances on both generative and predictive tasks compared to the atom-level tokenization. In the generative task, it leads to a significantly higher validity and novelty of generated SMILES. In the predictive tasks, SPE shows better or comparable performances on 23 out of 24 datasets. In addition to the strong performances, SPE has shorter input sequences which saves the computational cost of both model training and inferencing. SPE could represent a better tokenization method for the development of future deep learning applications in cheminformatics.

Reference

(1) Muratov, E. N., Bajorath, J., Sheridan, R. P., Tetko, I. V, Filimonov, D., Poroikov, V., Oprea, T. I., Baskin, I. I., Varnek, A., Roitberg, A., et al. (2020) QSAR without Borders. *Chem. Soc. Rev.*

(2) Chen, H., Engkvist, O., Wang, Y., Olivecrona, M., Blaschke, T. (2018) The Rise of Deep Learning in Drug Discovery. *Drug Discov. Today 23,* 1241–1250.

(3) Lavecchia, A. (2019) Deep Learning in Drug Discovery: Opportunities, Challenges and Future Prospects. *Drug Discov. Today 24,* 2017–2032.

(4) Stokes, J. M., Yang, K., Swanson, K., Jin, W., Cubillos-Ruiz, A., Donghia, N. M., MacNair, C. R., French, S., Carfrae, L. A., Bloom-Ackerman, Z., et al. (2020) A Deep Learning Approach to Antibiotic Discovery. *Cell 180,* 688-702.e13.

(5) Maziarka, Ł., Danel, T., Mucha, S., Rataj, K., Tabor, J., Jastrzębski, S. (2020) Molecule Attention Transformer.

(6) Elton, D. C., Boukouvalas, Z., Fuge, M. D., Chung, P. W. (2019) Deep Learning for Molecular Design - A Review of the State of the Art. *Mol. Syst. Des. Eng. 4,* 828–849.

(7) Polykovskiy, D., Zhebrak, A., Sanchez-Lengeling, B., Golovanov, S., Tatanov, O., Belyaev, S., Kurbanov, R., Artamonov, A., Aladinskiy, V., Veselov, M., et al. (2018) Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models.

(8) Brown, N., Fiscato, M., Segler, M. H. S., Vaucher, A. C. (2019) GuacaMol: Benchmarking Models for de Novo Molecular Design. *J. Chem. Inf. Model. 59,* 1096–1108.

(9) Öztürk, H., Özgür, A., Schwaller, P., Laino, T., Ozkirimli, E. (2020) Exploring Chemical Space Using Natural Language Processing Methodologies for Drug Discovery. *Drug Discov. Today 00.*

(10) Lipton, Z. C., Berkowitz, J., Elkan, C. (2015) A Critical Review of Recurrent Neural Networks for Sequence Learning.

(11) Kim, Y. (2014) Convolutional Neural Networks for Sentence Classification.

(12) Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., Polosukhin, I. (2017) Attention Is All You Need. In *Advances in Neural Information Processing Systems*; Vol. 2017-Decem, pp 5999–6009.

(13) Weininger, D. (1988) SMILES, a Chemical Language and Information System. 1.

Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Model. 28*, 31–36.

(14) Weininger, D., Weininger, A., Weininger, J. L. (1989) SMILES. 2. Algorithm for Generation of Unique SMILES Notation. *J. Chem. Inf. Model. 29*, 97–101.

(15) Sennrich, R., Haddow, B., Birch, A. (2016) Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*; Association for Computational Linguistics: Stroudsburg, PA, USA; pp 1715–1725.

(16) Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. (2014) Learning Phrase Representations Using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*; Association for Computational Linguistics: Stroudsburg, PA, USA; Vol. 281, pp 1724–1734.

(17) Gaulton, A., Bellis, L. J., Bento, A. P., Chambers, J., Davies, M., Hersey, A., Light, Y., McGlinchey, S., Michalovich, D., Al-Lazikani, B., et al. (2012) ChEMBL: A Large-Scale Bioactivity Database for Drug Discovery. *Nucleic Acids Res. 40*, 1100–1107.

(18) O'Boyle, N., Dalke, A. (2018) DeepSMILES: An Adaptation of SMILES for Use in Machine-Learning of Chemical Structures. *chemRxiv*.

(19) Krenn, M., Häse, F., Nigam, A., Friederich, P., Aspuru-Guzik, A. (2019) Self-Referencing Embedded Strings (SELFIES): A 100% Robust Molecular String Representation.

(20) Cortés-Ciriano, I., Bender, A. (2019) Deep Confidence: A Computationally Efficient Framework for Calculating Reliable Prediction Errors for Deep Neural Networks. *J. Chem. Inf. Model. 59*, 1269–1281.

(21) Swain, M. MolVS: Molecule Validation and Standardization.

(22) Landrum, G. RDKit: Open-Source Cheminformatics.

(23) Segler, M. H. S., Kogej, T., Tyrchan, C., Waller, M. P. (2018) Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks. *ACS Cent. Sci. 4*, 120–131.

(24) Li, X., Fourches, D. (2020) Inductive Transfer Learning for Molecular Activity Prediction: Next-Gen QSAR Models with MolPMoFiT. *J. Cheminform. 12*, 27.

(25) Howard, J., Ruder, S. (2018) Universal Language Model Fine-Tuning for Text Classification. In *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*; Vol. 1, pp 328–339.

(26) Cohen, J. (1988) *Statistical Power Analysis for the Behavioral Sciences*; L. Erlbaum Associates: Hillsdale, N.J.

(27) Nicholls, A. (2016) Confidence Limits, Error Bars and Method Comparison in Molecular Modeling. Part 2: Comparing Methods. *J. Comput. Aided. Mol. Des. 30*, 103–126.

(28) Bjerrum, E. J. (2017) SMILES Enumeration as Data Augmentation for Neural Network Modeling of Molecules.

(29) Arús-Pous, J., Johansson, S. V., Prykhodko, O., Bjerrum, E. J., Tyrchan, C., Reymond, J.-L., Chen, H., Engkvist, O. (2019) Randomized SMILES Strings Improve the Quality of

Molecular Generative Models. *J. Cheminform. 11*, 1–13.

(30) Arús-Pous, J., Blaschke, T., Ulander, S., Reymond, J. L., Chen, H., Engkvist, O. (2019) Exploring the GDB-13 Chemical Space Using Deep Generative Models. *J. Cheminform. 11*, 20.

(31) Karpov, P., Godin, G., Tetko, I. V. (2020) Transformer-CNN: Swiss Knife for QSAR Modeling and Interpretation. *J. Cheminform. 12*, 17.

(32) Merity, S., Keskar, N. S., Socher, R. (2018) Regularizing and Optimizing LSTM Language Models. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*.

(33) Smith, L. N. (2018) A Disciplined Approach to Neural Network Hyper-Parameters: Part 1 -- Learning Rate, Batch Size, Momentum, and Weight Decay.

(34) Adam Paszke; Sam Gross; et al. (2017) Automatic Differentiation in PyTorch. *31st Conf. Neural Inf. Process. Syst. (NIPS 2017)*.

(35) Howard, J., Gugger, S. (2020) Fastai: A Layered API for Deep Learning. *Information 11*, 108.

(36) Zheng, S., Yan, X., Yang, Y., Xu, J. (2019) Identifying Structure–Property Relationships through SMILES Syntax Analysis with Self-Attention Mechanism. *J. Chem. Inf. Model. 59*, 914–923.

(37) Goh, G. B., Hodas, N. O., Siegel, C., Vishnu, A. (2017) SMILES2Vec: An Interpretable General-Purpose Deep Neural Network for Predicting Chemical Properties.

Supplemental Materials

Table S1. Performance of QSAR models trained with SPE tokenization.

Targets	RMSE	R^2	MAE
A2a	0.669±0.111	0.724±0.116	0.523±0.078
Dopamine	0.719±0.055	0.522±0.080	0.551±0.045
Dihydrofolate	0.771±0.053	0.545±0.049	0.586±0.042
Carbonic	0.582±0.035	0.781±0.046	0.427±0.032
ABL1	0.746±0.050	0.637±0.067	0.566±0.045
opioid	0.636±0.046	0.742±0.042	0.477±0.032
Cannabinoid	0.671±0.041	0.715±0.036	0.500±0.021
COX-1	0.655±0.064	0.486±0.064	0.489±0.046
Monoamine	0.604±0.054	0.645±0.046	0.454±0.037
LCK	0.758±0.035	0.673±0.048	0.589±0.028
Glucocorticoid	0.541±0.042	0.692±0.057	0.426±0.030
Ephrin	0.652±0.051	0.647±0.037	0.494±0.026
Caspase	0.586±0.098	0.835±0.079	0.440±0.093
Coagulation	0.771±0.037	0.622±0.039	0.580±0.031
Estrogen	0.630±0.037	0.780±0.024	0.459±0.032
B-raf	0.599±0.057	0.756±0.046	0.452±0.036
Glycogen	0.731±0.028	0.593±0.041	0.548±0.023
Vanilloid	0.669±0.055	0.543±0.080	0.522±0.032
Aurora-A	0.711±0.049	0.723±0.033	0.530±0.035
JAK2	0.623±0.046	0.725±0.041	0.467±0.026
COX-2	0.728±0.043	0.605±0.050	0.537±0.032
Acetylcholinesterase	0.675±0.0049	0.749±0.033	0.495±0.033
erbB1	0.658±0.023	0.757±0.011	0.492±0.019
HERG	0.536±0.019	0.625±0.033	0.395±0.019

Table S2. Performance of QSAR models trained with atom-level tokenization.

Targets	RMSE	R^2	MAE
A2a	0.776±0.224	0.612±0.215	0.550±0.151
Dopamine	0.748±0.097	0.479±0.140	0.576±0.071
Dihydrofolate	0.794±0.101	0.525±0.101	0.592±0.072
Carbonic	0.578±0.069	0.792±0.046	0.421±0.052
ABL1	0.750 0.046	0.635±0.046	0.574±0.034
opioid	0.642±0.072	0.735±0.066	0.485±0.049
Cannabinoid	0.717±0.055	0.679±0.034	0.552±0.033
COX-1	0.665±0.094	0.484±0.089	0.478±0.058
Monoamine	0.624±0.061	0.633±0.053	0.467±0.048
LCK	0.835±0.165	0.591±0.181	0.617±0.047
Glucocorticoid	0.535±0.058	0.695±0.074	0.411±0.042
Ephrin	0.664±0.055	0.636±0.043	0.508±0.027
Caspase	0.587±0.061	0.837±0.050	0.444±0.046
Coagulation	0.770±0.037	0.622±0.045	0.582±0.023
Estrogen	0.655±0.044	0.761±0.028	0.474±0.031
B-raf	0.599±0.067	0.762±0.046	0.443±0.043
Glycogen	0.744±0.045	0.579±0.052	0.555±0.040
Vanilloid	0.670±0.065	0.542±0.082	0.515±0.040
Aurora-A	0.744±0.073	0.698±0.044	0.547±0.040
JAK2	0.642±0.035	0.708±0.042	0.481±0.023
COX-2	0.736±0.064	0.596±0.074	0.543±0.048
Acetylcholinesterase	0.679±0.060	0.745±0.044	0.485±0.037
erbB1	0.661±0.019	0.754±0.013	0.492±0.016
HERG	0.531±0.025	0.637±0.030	0.391±0.020