Transformer-based Approach for Predicting Chemical Compound Structures

Yutaro Omote
Ehime University
omote@ai.cs.ehime-u.ac.jp

Kyoumoto Matsushita
Fujitsu Laboratories, Ltd.
m.kyoumoto@fujitsu.com

Tomoya Iwakura
Fujitsu Laboratories, Ltd.
iwakura.tomoya@fujitsu.com

Akihiro Tamura
Doshisha University
aktamura@mail.doshisha.ac.jp

Takashi Ninomiya
Ehime University
ninomiya@cs.ehime-u.ac.jp

Abstract

By predicting chemical compound structures from their names, we can better comprehend chemical compounds written in text and identify the same chemical compound given different notations for database creation. Previous methods have predicted the chemical compound structures from their names and represented them by Simplified Molecular Input Line Entry System (SMILES) strings. However, these methods mainly apply handcrafted rules, and cannot predict the structures of chemical compound names not covered by the rules. Instead of handcrafted rules, we propose Transformer-based models that predict SMILES strings from chemical compound names. We improve the conventional Transformer-based model by introducing two features: (1) a loss function that constrains the number of atoms of each element in the structure, and (2) a multi-task learning approach that predicts both SMILES strings and InChI strings (another string representation of chemical compound structures). In evaluation experiments, our methods achieved higher F-measures than previous rule-based approaches (Open Parser for Systematic IUPAC Nomenclature and two commercially used products), and the conventional Transformer-based model. We release the dataset used in this paper as a benchmark for the future research[1].

1 Introduction

Knowledge of chemical substances is necessary for developing new materials and drugs, and for synthesizing products from new materials. To utilize such knowledge, researchers have created databases containing the physical property values of chemical substances and the interrelationships among chemical substances.

It is thought that several billions of chemical compounds exist (Lahana, 1999; Hoffmann and Gastreich, 2019), but only a portion of these are entered into chemical databases. Even PubChem[2], one of the largest databases of chemical compounds, includes the information of only approximately 100 million chemical compounds. Moreover, databases for chemical domains are manually maintained, which consumes much time and cost. One of the time consuming processes is the integration of the same chemical compounds with different notations. For instance, a chemical structure can be derived from partial structures which are given notational variants, or the notation can fluctuate for a given chemical compound (Watanabe et al., 2019). Therefore, a system that automatically predicts a chemical compound structure from its chemical compound names would improve the database creation procedure.

Structures are most commonly predicted from their notations by rule-based conversion methods (Lowe et al., 2011). Although rule-based conversion can accurately predict the structures of chemical compounds based on systematic nomenclatures such as the International Union of Pure and Applied Chemistry (IUPAC) [3] nomenclature, it often fails the structure prediction of chemical compound names that violate these nomenclatures (e.g., Synonyms[4]).

To improve the low prediction performance of compounds with non-IUPAC names, we propose neural network-based models that predict chemical compound structures represented as Simplified Molecular Input Line Entry System (SMILES) (Weininger, 1988) strings from chemical compound names categorized as Synonyms[5]. In this work, we use the Transformer-based sequence-

[1] http://aiweb.cs.ehime-u.ac.jp/pred-chem-struct

[2] https://pubchem.ncbi.nlm.nih.gov/

[3] https://iupac.org

[4] PubChem's definition of chemical compound names other than IUPAC names

[5] Our Synonyms excludes DATABASE IDs from the original definition of Synonyms because DATABASE IDs can be efficiently recognized by rules.

Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pages 154–162
December 4 - 7, 2020.

Name Type	Name
IUPAC	2-acetyloxybenzoic acid
DATABASE ID (CAS registry number)	50-78-2
ABBREVIATION	ASA
COMMON	aspirin

Table 1: Examples of "aspirin" representations. In this table, ABBREVIATION and COMMON are Synonyms.

to-sequence neural network model (Vaswani et al., 2017) for machine translation, which achieves a state-of-the-art performance in various tasks among the sequence-to-sequence neural network models such as recurrent neural network-based models. To improve the conventional Transformer-based model, we introduce the following two chemical-structure oriented features:

1. A loss function considering the constraints on the number of atoms of each element in the chemical structure.

2. A multi-task learning for predicting both SMILES strings and IUPAC International Chemical Identifier (InChI) (Heller et al., 2015) strings, which are representations for denoting chemical compound structures as strings.

For our experiments, we created a dataset from PubChem for predicting chemical compound structures represented by SMILES strings from Synonyms. The experimental results demonstrate the Transformer-based conversion methods achieve higher F-measures than the existing rule-based methods. In addition, our two proposals (i.e., constraining the number of atoms of each element and multi-task learning of both SMILES strings and InChI strings) improve the performance of the conventional Transformer-based method.

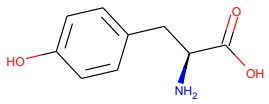
2 Preliminary

2.1 Chemical Compound Names

In PubChem, the text names of chemical compounds are represented by three main types of notational categories: IUPAC, DATABASE ID, and Synonyms. IUPAC is a systematic nomenclature for chemical compound names. DATABASE ID is the unique identifier of a chemical compound in a database. An example is the Chemical Abstracts Service (CAS) [6] registry number. The Synonyms

[6] https://www.cas.org/

[Chemical Structure]



[SMILES]

N[C@@H](Cc1ccc(O)cc1)C(=O)O

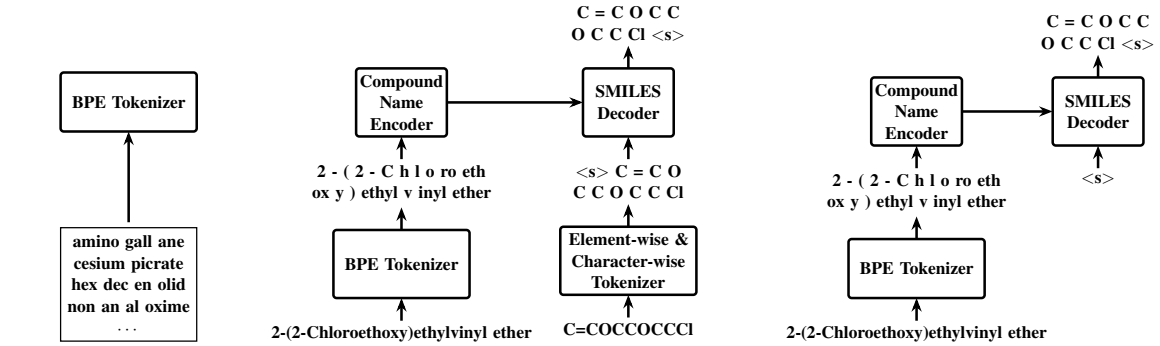
[InChI]

InChI=1S/C9H11NO3/c10-8(9(12)13)5-6-1-3-7(11)4-2-6/h1-4,8,11H,5,10H2,(H,12,13)/t8-/m0/s1

Figure 1: Chemical structure of L-tyrosine (top), and its SMILES (middle) and InChI (bottom) representations

naming category in PubChem includes ABBREVIATION and COMMON. As an example, Table 1 shows various "aspirin" representations.

The IUPAC nomenclature provides a systematic naming under standardized rules, which are easily and accurately converted by rule-based conversion methods (Lowe et al., 2011); (Heller et al., 2015). Provided they are registered in the database, DATABASE IDs are easily converted to their corresponding chemical compounds using dictionary-lookup methods. However, neither rule-based nor dictionary-based approach can convert chemical compound names that are not covered by the rules or dictionaries. Unlike IUPAC and DATABASE ID notations, the naming patterns of Synonyms are complex and widely variable. In many cases, the chemical compound names appearing in documents cannot be converted by rule-based or dictionary-based approaches. Consequently, the prediction performance of chemical compound names is worse in Synonyms than in IUPAC, as shown in section 6.1. In our preliminary experiments, the highest F-measure obtained with an existing tool exceeded 0.96 on IUPAC data, but was reduced to 0.75 on Synonyms data.



(a) Learning Tokenization (b) Training Model (c) Inference

Figure 2: Overview of Transformer-based prediction of SMILES strings from chemical compound names

2.2 Representation of Chemical Compound Structures

For multi-task learning, we represented chemical compound structures as SMILES strings and InChI strings. These two representations are major notations of chemical compound structures. We use SMILES strings as the target representation because they are simpler than InChI strings but were sufficiently representative for our purpose (i.e., creating a chemical compound database).

The SMILES (Weininger, 1988) notation system was designed for modern chemical information processing. Based on the principles of molecular graph theory, SMILES allows rigorous structure specification using a very small and natural grammar. SMILES strings are composed of atoms and symbols representing their bonds, branches, rings, and other structural features, assembled into a linear expression of the two-dimensional structure of a molecule. An example of a SMILES string is shown in Figure 1. In this work, we used Canonical SMILES because it uniquely determines the correspondence between chemical structures and SMILES strings.

In the InChI (Heller et al., 2015) representation, the information of a chemical compound structure is represented by five layers. In Figure 1, the layers are separated by "/" symbols. Each layer adds detailed information to the following layer. Because these layers are interrelated, InChI strings are more complex than SMILES strings.

3 Proposed Methods

This section presents our proposed methods, namely, our tokenizer training method and sequence-to-sequence models. Let $\mathcal{X}$ and $\mathcal{T}$ be a set of chemical compound names and a set of SMILES strings, respectively. We define a training dataset consisting of n samples as $D = \langle (X_1, T_1), ..., (X_n, T_n) \rangle$, where $X_i \in \mathcal{X}$ is a chemical compound name and $T_i \in \mathcal{T}$ is the SMILES string of X_i for $1 \leq i \leq n$. Our objective is to learn a mapping function f that realizes $f(X_i) = T_i$ from D.

Figure 2 overviews the Transformer-based prediction of SMILES strings from chemical compound names, where <s> is a special symbol denoting the start and end of a sequence. Chemical compound names, SMILES, and InChI are long strings without explicit boundaries (such as white spaces in English text). Therefore, to convert chemical compound names to SMILES strings, we propose (a) training of a tokenizer and (b) a Transformer-based approach.

3.1 Tokenizer

Chemical compound names can be tokenized by the Open Parser for Systematic IUPAC Nomenclature (OPSIN) (Lowe et al., 2011) tokenizer, a rule-based parser that generates SMILES and InChI strings from chemical compound names (mainly, from IUPAC names). However, some chemical compound names, especially Synonyms, cannot be tokenized by rule-based tokenizers such as OPSIN. In particular, the OPSIN tokenizer is limited to chemical compound names covered by its dictionary and rules; meanwhile (as mentioned above) chemical compound names lack explicit word-boundary markers. To overcome these restrictions, we propose a method that trains tokenizers for Synonyms, SMILES, and InChI representations. Note that InChI is used in a multi-task learning.

To eliminate the unknown tokens, our tokenizer learning method is unsupervised and covers a large

set of chemical compound names. The tokenization is performed by byte pair encoding (BPE) (Sennrich et al., 2016)[7]. The BPE-based tokenizer was learned by fastBPE[8] . First, the chemical compound names obtained by the OPSIN tokenizer were segmented because fastBPE requires segmented input text. By virtue of the newly obtained BPE dictionary, the BPE-based tokenizer can tokenize chemical compound names that cannot be handled by the OPSIN tokenizer.

When tokenizing the SMILES strings, each element (e.g., "C", "O", "Cl") identified by regular expressions was regarded as one token. The remaining symbols not covered by regular expressions were divided into single characters, each regarded as one token.

For tokenizing InChI strings, the model was learned on SentencePiece (Kudo and Richardson, 2018), a unigram-based unsupervised training method for word segmentation. Note that InChI strings cannot be tokenized by BPE because the segmentations of InChI strings are not preliminarily given.

3.2 Transformer-based Prediction of SMILES Strings from Chemical Compound Names

The Transformer model consists of stacked encoder and decoder layers. Based on self-attention, it attends to tokens in the same sequence, i.e., a single input sequence or a single output sequence. The encoder maps an input sequence to a sequence of vector representations. From this vector representations, the decoder generates an output sequence.

The Transformer-based model predicts SMILES strings from chemical compound names, so its input is a chemical compound name and its output is a SMILES string. During the learning process, the following objective function is minimized:

$$\mathcal{L}_{smiles} = -\log P(T|X;\boldsymbol{\theta}_{enc},\boldsymbol{\theta}_{smiles}), \quad (1)$$

where $\boldsymbol{\theta}_{enc}$ and $\boldsymbol{\theta}_{smiles}$ are the parameter sets of the compound name encoder and SMILES decoder, respectively, and $X = \langle x_1, x_2, \ldots, x_n \rangle$ is the word sequence of a chemical compound name segmented by the BPE model. $T = \langle t_1, t_2, \ldots, t_m \rangle$ is the sequence of elements and symbols in the correct SMILES string of X.

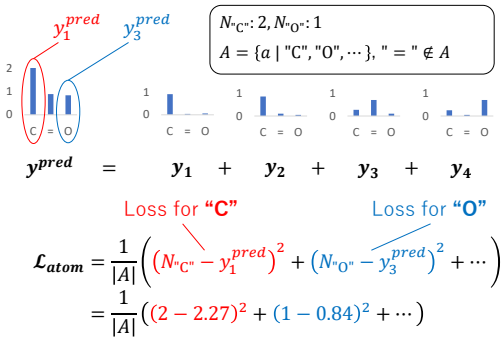


Figure 3: Calculating the constraints on the number of atoms of each element

3.3 Training with a Constraint on the Number of Atoms

To correctly predict the chemical structure from a chemical compound name, the number of atoms of each element included in the chemical structure must be fixed. In this subsection, we propose a softmax-based loss function that constrains the number of atoms of each element, that is, we minimize the difference between the numbers of atoms of each element in the predicted and correct SMILES strings. The differences are measured by their squared errors.

The squared errors are computed using the Gumbel softmax (Jang et al., 2016) function, which obtains the probability distribution of the number of atoms of each element in a predicted SMILES string. Let $\boldsymbol{\pi_i} = (\pi_{i1}, \pi_{i2}, \ldots, \pi_{i|\mathcal{V}|})$ be the probability distribution of the i-th output token from the Transformer model. Then, $\boldsymbol{y_i} = (y_{i1}, y_{i2}, \ldots, y_{i|\mathcal{V}|})$ for the i-th output token with Gumbel softmax is calculated as follows:

$$y_{ij} = \frac{\exp\left((\log(\pi_{ij}) + g_{ij})/\tau\right)}{\sum_{k=1}^{|\mathcal{V}|} \exp\left((\log(\pi_{ik}) + g_{ik})/\tau\right)}, \quad (2)$$
$$g_{ij} = -\log(-\log(u_{ij})),$$
$$u_{ij} \sim \text{Uniform}(0, 1),$$

where $\mathcal{V}$ represents the vocabulary set of SMILES, and τ is a hyperparameter of Gumbel softmax. The distribution $\boldsymbol{y_i}$ approximates an one-hot vector as τ decreases, and a uniform distribution as τ increases. In this work, τ was set to 0.1.

Using Equation 2, the loss function under the

[7] In preliminary experiments, BPE achieved a higher F-measure than SentencePiece (Kudo and Richardson, 2018). Therefore, it was used for tokenizing the chemical compound names.

[8] https://github.com/glample/fastBPE

proposed constraints is given by

$$\mathcal{L}_{atom} = \frac{1}{|A|}\sum_{a\in A}(N_a(T) - y^{pred}_{idx(a)})^2, \quad (3)$$
$$\boldsymbol{y^{pred}} = \boldsymbol{y_1} + \boldsymbol{y_2} + \cdots + \boldsymbol{y_m}$$
$$= (y^{pred}_1, y^{pred}_2, \ldots, y^{pred}_{|\mathcal{V}|}),$$

where A is a set of elements, $N_a(T)$ is a function that returns the number of atoms of element a in SMILES string T, and $idx(a)$ is a function that returns the index of element a in $\mathcal{V}$. Note that A contains only elemental symbols, and the other features such as symbols representing bonds are absent. More formally, "C", "O" $\in A$, "=", "#" $\notin A$, and $\mathcal{V} \supset A$. Each dimension of $\boldsymbol{y^{pred}}$ is an estimation of the frequency of the corresponding token of the vocabulary $\mathcal{V}$ in the predicted SMILES. The proposed constraint calculation uses only the estimation of the elements in $\mathcal{V}$. The frequencies of elements not included in the correct SMILES are set to 0.

As an example, Figure 3 shows how the number of atoms of each element is constrained when the correct SMILES string is "CC=O". As "C" and "O" are elements and "=" is a subsidiary symbol representing a double bond, the proposed constraint function treats the number of atoms of each element ("C" and "O") as the error to be minimized, and disregards the "=" symbol.

The objective function under the proposed constraints is defined as follows:

$$\mathcal{L}_{smiles} + \lambda_{atom}\mathcal{L}_{atom}, \quad (4)$$

where λ_{atom} is a hyperparameter that controls the degree of considering $\mathcal{L}_{atom}$.

3.4 Multi-task Learning for Predicting both SMILES Strings and InChI Strings

The same chemical structure is differently represented in a SMILES string and an InChI string. Assuming that the models for predicting SMILES and InChI strings compensate each other, we propose a multi-task learning method that shares the encoder of the name-to-SMILES and name-to-InChI conversion models, and trains both models at the same time.

Let $\mathcal{I}$ be the set of InChI strings. We define a training dataset consisting of n samples as $\tilde{D} = \langle (X_1, T_1, I_1), ..., , (X_n, T_n, I_n)\rangle$, where $X_i \in \mathcal{X}$, $T_i \in \mathcal{T}$, and $I_i \in \mathcal{I}$ for $1 \leq i \leq n$. The objective is to learn a function $\tilde{f}$ from $\tilde{D}$. $\tilde{f}(X_i)$ predicts both T_i and I_i.

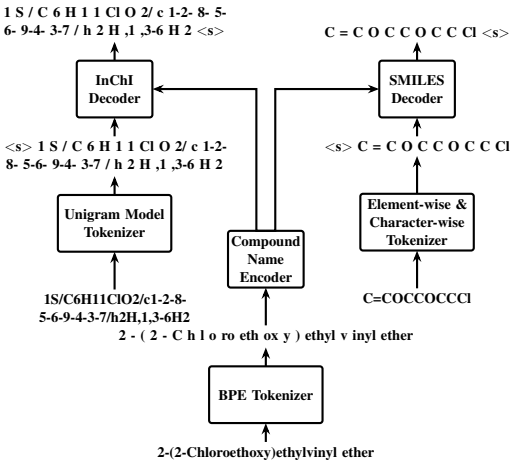


Figure 4: Overview of multi-task learning for predicting both SMILES strings and InChI strings

Split	Size
Training	5,000,000
Development	1,113
Test	11,194

Table 2: Sizes of the training, development, and test datasets

Specifically, the proposed multi-task learning minimizes the following objective function:

$$\mathcal{L}_{smiles} + \lambda_{inchi}\mathcal{L}_{inchi}, \quad (5)$$
$$\mathcal{L}_{inchi} = -\log P(I|X; \boldsymbol{\theta}_{enc}, \boldsymbol{\theta}_{inchi}),$$

where $\boldsymbol{\theta}_{inchi}$ and $\boldsymbol{\theta}_{enc}$ are parameter sets for the InChI decoder and shared encoder, respectively, and λ_{inchi} is a hyperparameter that controls the degree of considering $\mathcal{L}_{inchi}$. $\mathcal{L}_{smiles}$ is calculated by Eq. 1. The method is overviewed in Figure 4.

4 Experimental Settings

4.1 Data Set

In all experiments, the data comprised a chemical compound name and a correct SMILES string. Using the dump data of PubChem[9] (97M compound records), the chemical compound names were converted to Synonyms associated with each CID[10], and the correct SMILES strings were converted from isomeric SMILES strings[11] to canon-

[9] ftp://ftp.ncbi.nlm.nih.gov/pubchem/

[10] PubChem's compound identifier for a unique chemical structure

[11] SMILES strings written with isotopic and chiral specifications

method		recall	precision	F-measure
Rule-based	OPSIN	0.693	0.836	0.758
	tool A	0.711	0.797	0.752
	tool B	0.653	0.800	0.719
Transformer-based (BPE)	`transformer`	0.793	0.806	0.799
	`atomnum`	0.798	0.808	0.803
	`inchigen`	**0.810**	0.819	0.814
Transformer-based (OPSIN-TK + BPE)	`transformer`	0.763	0.873	0.814
	`atomnum`	0.768	0.876	0.818
	`inchigen`	0.779	**0.886**	**0.829**
Transformer-based (OPSIN-TK)	`transformer`	0.755	0.868	0.808
	`atomnum`	0.757	0.867	0.808
	`inchigen`	0.754	0.869	0.807

Table 3: Evaluation results of each converter for Synonyms. Transformer-based ones are our proposed methods. We evaluated the Transformer-based ones with different three tokenizers, BPE, OPSIN-TK+BPE, and OPSIN-TK.

ical SMILES strings using RDKit[12]. Note that in PubChem, the Synonyms includes the IUPAC names, common names, and IDs of the compounds in chemical compound databases. Here, we used the isomeric SMILES strings because they least overlap with their corresponding CIDs. In the multi-task learning, the InChI strings are also associated with CIDs.

From the dump data, 10,000 CIDs and 100,000 CIDs were randomly selected as the development and test datasets, respectively, and only the two chemical compound names with the longest edit distance were assigned to each CID.

To create Synonyms in the development and test data, chemical compound names like IDs in the chemical compound databases were removed using manually created regular expressions.

In the development and test datasets, duplicate chemical compound names with different CIDs were removed[13]. From the development and test datasets, we removed 820 and 8,241 duplicates, respectively.

As the training dataset, we selected chemical compound names that were categorized as Synonyms that could be tokenized by the OPSIN tokenizer. The size of each dataset is listed in Table 2.

4.2 Parameter Settings

The hyperparameters of the Transformer model were set as follows: number of stacks in the encoder and decoder layers = 6, number of heads = 8, embedding dimension = 512, and dropout probability = 0.1. The loss functions $\mathcal{L}_{smiles}$ and $\mathcal{L}_{inchi}$ were computed using a label-smoothing cross entropy with the smoothing parameter ϵ set to 0.1. The learning rate was linearly increased to 0.0005 over the first 4,000 steps. In later steps, it was decreased proportionally to the inverse square root of the step number. The optimizer was an Adam (Kingma and Ba, 2015) optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 10^{-8}$. The model parameters were updated 300,000 times. The hyperparameters λ_{atom} and λ_{inchi} for controlling the degree of constraint consideration were set to 0.7 and 0.3, respectively. The number of merge operations for the BPE-based tokenizer of chemical compound names was set to 500. The vocabulary size for the tokenizer of InChI strings was set to 1,000. We tuned the hyperparameters for our constraints and subword on the development data.

To present the results of our Transformer-based models, we averaged the last 10 checkpoints (saved at 1,000-step intervals) of the Transformer models. We used beam search with a beam size of 4 and length penalty $\alpha = 0.6$ (Vaswani et al., 2017). The maximum output length of an inference was set to 200.

5 Experimental Results

5.1 Prediction Performance

The results are shown in Table 3. Here, tool A and tool B are two commercially available tools, `atomnum` indicates the method based on the number of atoms described in section 3.3, and `inchigen` denotes the multitask learning method

[12] https://github.com/rdkit/rdkit

[13] The same chemical compound name may have more than one CID.

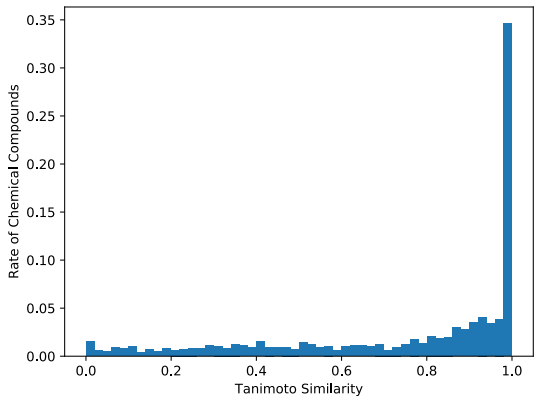


Figure 5: Histogram of Jaccard similarities between incorrect structures generated by `inchigen` with BPE and their correct structures

described in section 3.4. The notations BPE and OPSIN-TK indicate the use of the BPE-based and OPSIN tokenizers, respectively.

As confirmed in Table 3, the proposed methods attained higher prediction performance the existing rule-based methods and the conventional Transformer-based model. `inchigen` with BPE showed 0.056, 0.062, and 0.095 points higher F-measure than OPSIN, tool A, and tool B, respectively.

The F-measure was further improved by combining the two tokenizers (see the results of OPSIN-TK+BPE in Table 3). In the OPSIN-TK+BPE method, the Transformer-based method with BPE predicted the structures from chemical compound names that could be tokenized by the OPSIN tokenizer. The highest F-measure and precision (0.829 and 0.886, respectively) were achieved by `inchigen` with OPSIN-TK+BPE.

In the Transformer-based models, the OPSIN tokenizer obtained higher precision than the BPE-based tokenizers because approximately 11.5% (1,293 / 11,194) of the chemical compounds in the test set could not be tokenized by OPSIN. Consequently, the precision was improved by the reduced number of outputs. In contrast, the recall was lower than in the BPE-based tokenizers.

These results clarify the impact of tokenizer outputs on the recall, precision, and F-measure scores.

5.2 Error Analysis

Most of the predictions in the Transformer-based approach were grammatically correct SMILES strings. In this context, "grammatically correct" means that the chemical structure can be visualized from the predicted SMILES string using RDKit, and does not require the correct SMILES string of a chemical compound name. In particular, `inchigen` with BPE achieved grammatically correct predictions for 99 % of the test data, 10.6–17.4 % higher than OPSIN, tool A, and tool B. To evaluate the usefulness of the Transformer-based approach, we also analyzed the proportion of incorrect structure predictions that were grammatically correct SMILES strings but did not match the correct SMILES strings.

To this end, we measured the Jaccard similarity (Tanimoto similarity)[14] between each structure that was incorrectly predicted by `inchigen` with BPE and the correct structure. The Jaccard similarity, a common technique for measuring chemical compound similarities, is defined as follows:

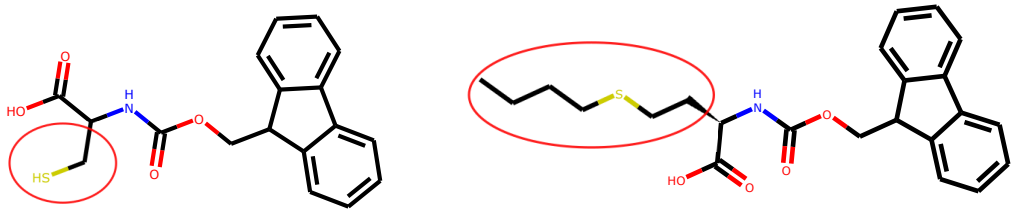
$$J(X,Y) = \frac{\boldsymbol{v}_X \cdot \boldsymbol{v}_Y}{|\boldsymbol{v}_X + \boldsymbol{v}_Y| - \boldsymbol{v}_X \cdot \boldsymbol{v}_Y},$$

where the $\boldsymbol{v}_X$ and $\boldsymbol{v}_Y$ are binary chemical fingerprints of chemical compounds X and Y, respectively, represented by binary vectors. $|\boldsymbol{v}|$ is the L1 norm of $\boldsymbol{v}$, and $\boldsymbol{v}_X \cdot \boldsymbol{v}_Y$ is the inner product of $\boldsymbol{v}_X$ and $\boldsymbol{v}_Y$. Here, a chemical fingerprint expresses a chemical compound structure as a calculable vector. A famous type of fingerprint is a series of binary digits (bits) that represent the presence or absence of particular partial structures in the chemical compound. For example, the Molecular Access System key (Durant et al., 2002), which is used as the fingerprints in the present evaluation, comprises 166 partial structures of chemical compounds. Figure 5 is a histogram of the Jaccard similarity scores obtained in this analysis. We find that most of the incorrect SMILES strings generated by `inchigen` with BPE possessed high Jaccard similarities to the correct SMILES strings. The average Jaccard similarity was 0.753.

An incorrect structure generated by `inchigen` with BPE is compared with its correct structure in Figure 6. The two structures differed only by whether ethylsulfanylbutane or methanethiol was bonded in the partial structures enclosed by the red ellipses. In other words, the two structures are very similar (Jaccard similarity = 0.76).

From this result, we observe that even when the proposed method generates an incorrect structure,

[14] Jaccard similarity, also called the Tanimoto similarity, measures the similarities between pairs of chemical compounds.



(a) Predicted by `inchigen` with BPE

(b) Correct Chemical Structure

Figure 6: Example of a chemical structure mistakenly for **"fmoc-l-buthionine"**. The red-edged ellipses enclose the partial structures that differ between the two chemical structures.

the outcome does not deviate greatly from the correct structure.

6 Related Work

6.1 Predicting SMILES Strings from Chemical Compound Names

OPSIN (Lowe et al., 2011) is a rule-based parser that generates SMILES strings and InChI strings from chemical compound names (mainly from IUPAC names). The OPSIN tokenization approach is based on regular grammar. From a tokenized chemical name, an XML parse tree is constructed. Stepwise operations on this tree are continued until the structure has been reconstructed from the name. The construction is performed on substructures associated with the terms.

As mentioned earlier, many of chemical compound names described in papers and patents do not comply with IUPAC names or other systematic nomenclatures, so are difficult to reconstruct using rule-based methods. In our preliminary experiments using OPSIN and commercially available tools, the F-measures of predicting the IUPAC names in the dataset ranged from 0.878 to 0.960. However, on the Synonyms dataset, the F-measures fell to 0.719-0.758.

6.2 Deep Learning methods using SMILES

Recently, SMILES strings have been applied to chemical reaction prediction (Nam and Kim, 2016; Schwaller et al., 2019). The method of Nam and Kim (2016) predicts SMILES strings representing products from SMILES strings representing reactants and reagents. This method employs a sequence-to-sequence model with an attention mechanism based on a recurrent neural network (Bahdanau et al., 2015). Schwaller et al. (2019) achieved higher accuracy than Nam and Kim (2016)'s model by applying the conventional Transformer model (Vaswani et al., 2017).

Similarly to our study, their models adapt SMILES strings to sequence-to-sequence models, but our target task (predicting chemical structures from their chemical compound names) differs from theirs. To improve the accuracy of our target task, we will improve the update speed and quality of our chemical compounds databases. We also intend to solve other chemistry problems, including chemical reactions, by predictive machine learning.

7 Conclusions

This paper introduced our Transformer-based prediction methods, which convert chemical compound names to SMILES strings trained with the constraint of the number of atoms of each element in the SMILES string. We also proposed a multi-task learning approach that simultaneously learns the conversions to SMILES strings and InChI strings. In an experimental comparison evaluation, our proposed method achieved higher F-measures than the existing methods.

In future work, we intend to explore various tokenization methods, and further improve the prediction performance. We also hope to apply the proposed loss function to multi-task learning.

Acknowledgments

The research results were achieved by the RIKEN AIP-FUJITSU Collaboration Center, Japan.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Joseph L. Durant, Burton A. Leland, Douglas R. Henry, and James G. Nourse. 2002. Reoptimization of mdl keys for use in drug discovery. *Journal of Chemical Information and Computer Sciences*, 42(6):1273–1280.

Stephen R Heller, Alan McNaught, Igor Pletnev, Stephen Stein, and Dmitrii Tchekhovskoi. 2015. Inchi, the iupac international chemical identifier. *Journal of cheminformatics*, 7(1):23.

Torsten Hoffmann and Marcus Gastreich. 2019. The next level in chemical space navigation: going far beyond enumerable compound libraries. *Drug Discovery Today*, 24(5):1148 – 1156.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *ArXiv*, abs/1611.01144.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Roger Lahana. 1999. How many leads from hts? *Drug Discovery Today*, 4(10):447 – 448.

Daniel M. Lowe, Peter T. Corbett, Peter Murray-Rust, and Robert C. Glen. 2011. Chemical name to structure: Opsin, an open source solution. *Journal of Chemical Information and Modeling*, 51(3):739–753.

Juno Nam and Jurae Kim. 2016. Linking the neural machine translation and the prediction of organic chemistry reactions.

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

Rico Sennrich, Barry Haddow, and Alexandra Birch. 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)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, L ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Taiki Watanabe, Akihiro Tamura, Takashi Ninomiya, Takuya Makino, and Tomoya Iwakura. 2019. Multi-task learning for chemical named entity recognition with chemical compound paraphrasing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6243–6248.

David Weininger. 1988. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36.