STOUT V2.0: SMILES to IUPAC name conversion using transformer models

Kohulan Rajan[1], Achim Zielesny[2], and Christoph Steinbeck[1]*

Institute for Inorganic and Analytical Chemistry, Friedrich Schiller University Jena, Lessingstr. 8, 07743 Jena, Germany

Institute for Bioinformatics and Chemoinformatics, Westphalian University of Applied Sciences, August-Schmidt-Ring 10, 45665 Recklinghausen, Germany

*Corresponding author: christoph.steinbeck@uni-jena.de

Abstract

Naming chemical compounds systematically is a complex task governed by a set of rules established by the International Union of Pure and Applied Chemistry (IUPAC). These rules are universal and widely accepted by chemists worldwide, but their complexity makes it challenging for individuals to consistently apply them accurately. A translation method can be employed to address this challenge. Accurate translation of chemical compounds from SMILES notation into their corresponding IUPAC names is crucial, as it can significantly streamline the laborious process of naming chemical structures. Here, we present STOUT (SMILES-TO-IUPAC-name translator) V2.0, which addresses this challenge by introducing a transformer-based model that translates string representations of chemical structures into IUPAC names. Trained on a dataset of nearly 1 billion SMILES strings and their corresponding IUPAC names, STOUT V2.0 demonstrates exceptional accuracy in generating IUPAC names, even for complex chemical structures. The model's ability to capture intricate patterns and relationships within chemical structures enables it to generate precise and standardised IUPAC names.

Deterministic algorithms for systematically naming chemical structures have been available for many years. Also, this work has only been possible through an academic license for OpenEye's Lexichem software.

Scientific Contribution:

STOUT V2.0, built upon transformer-based models, is a significant advancement from our previous work. The user-friendly web application enhances its accessibility and utility. By making the model and source code fully open and well-documented, we aim to promote unrestricted use and encourage further development.

Graphical Abstract

O=C1C2=C(N=CN2C)N(C(=O)N1C)C
STOUT
2
1,3,7-trimethylpurine-2,6-dione

Keywords

Transformers, STOUT, SMILES to IUPAC name, Chemical name translation, Deep Learning

Introduction

Chemists usually assign chemical structures a name when they are discovered or synthesised for the first time. These names can be trivial or systematic. The systematic names must follow a set of rules specified by the International Union of Pure and Applied Chemistry (IUPAC) [1] [2–4] [5]. These rules are comprehensive and complex, making it difficult to apply them consistently, especially for large datasets of chemical compounds or highly complex chemical structures.

Several commercial software packages are available to generate IUPAC names from chemical structures [6–9]. Chemaxon [10] and OpenEye [11] offer their rule-based software under an academic license to generate IUPAC names. This software enables chemists to create IUPAC names for specified structures automatically. Due to their deterministic algorithms, these rule-based software packages are reliable and widely used. The work presented here was only possible due to their existence (see below).

Machine learning, particularly deep neural networks, has shown promise in various domains [12,13], including natural language processing (NLP) [14]. It has been successfully applied in tasks like language translation [15], demonstrating the ability to learn complex patterns and relationships from large datasets [16,17].

This success has inspired researchers to explore the application of neural networks in cheminformatics, particularly in tasks like predicting chemical reactions [18], Optical Chemical Structure Recognition (OCSR) [19], drug discovery [20], generating novel molecules [21,22], molecular design and optimising [23] and many more [24].

Recent studies have explored the use of sequence-based neural networks for translating between chemical representations, such as SMILES strings and IUPAC names [25–27]. These studies highlight the potential of machine learning to address the challenges associated with IUPAC name generation, paving the way for more accessible and efficient tools for chemists.

However, we have no intention of competing with commercial rule-based tools that are more reliable due to their deterministic nature.

In recent years, OPSIN (Open Parser for Systematic IUPAC Nomenclature) [28] has been developed as the only open-source rule-based system for parsing IUPAC names into SMILES strings. It uses a regular grammar approach to guide tokenisation and constructs an XML parse tree from the IUPAC name, which is then processed to reconstruct the chemical structure. In this work, we use OPSIN to retranslate the IUPAC names generated by STOUT.

In our previous work [26], we introduced STOUT (SMILES-TO-IUPAC-name translator), a deep learning model based on a sequence-to-sequence (seq2seq) architecture with an encoder and decoder using Recurrent Neural Networks (RNNs) with Gated Recurrent Units (GRUs) [26]. STOUT was trained on a dataset of 60 million molecules from PubChem [29] and corresponding IUPAC names generated with ChemAxon's molconvert software [30]. The model achieved promising results, with an average BLEU score of approximately 90% and a Tanimoto similarity index of over 0.9, indicating high accuracy in predicting IUPAC names from SMILES strings and vice versa. However, there were still areas for improvement, particularly in model architecture, tokenisation strategies, and the handling of stereochemical information.

This work presents STOUT V2.0, a transformer-based model for SMILES to IUPAC name translation (see Figure 1). The model was trained on a dataset of nearly 1 billion SMILES and corresponding IUPAC names, generated using OpenEye's Lexichem software. This new version achieves very high accuracy on test and benchmark datasets, demonstrating improved capability in producing longer IUPAC names with fewer errors overall. The models were trained entirely on TPU VMs, and after finalising the training, they were optimised using TensorFlow to run efficiently on CPUs. and have been made available along with the accompanying code as open-source resources. To enhance accessibility for users with limited or no programming experience, a web application has been developed and is accessible at https://stout.decimer.ai. We demonstrate that neural networks can accurately perform the non-trivial task of converting SMILES to IUPAC names. However, we accomplished this with the generous access to Lexichem software that OpenEye provided us. The use of deterministic algorithms for generating IUPAC names, such as Lexichem, in production environments is always recommended due to the higher error rates associated with neural machine translation.

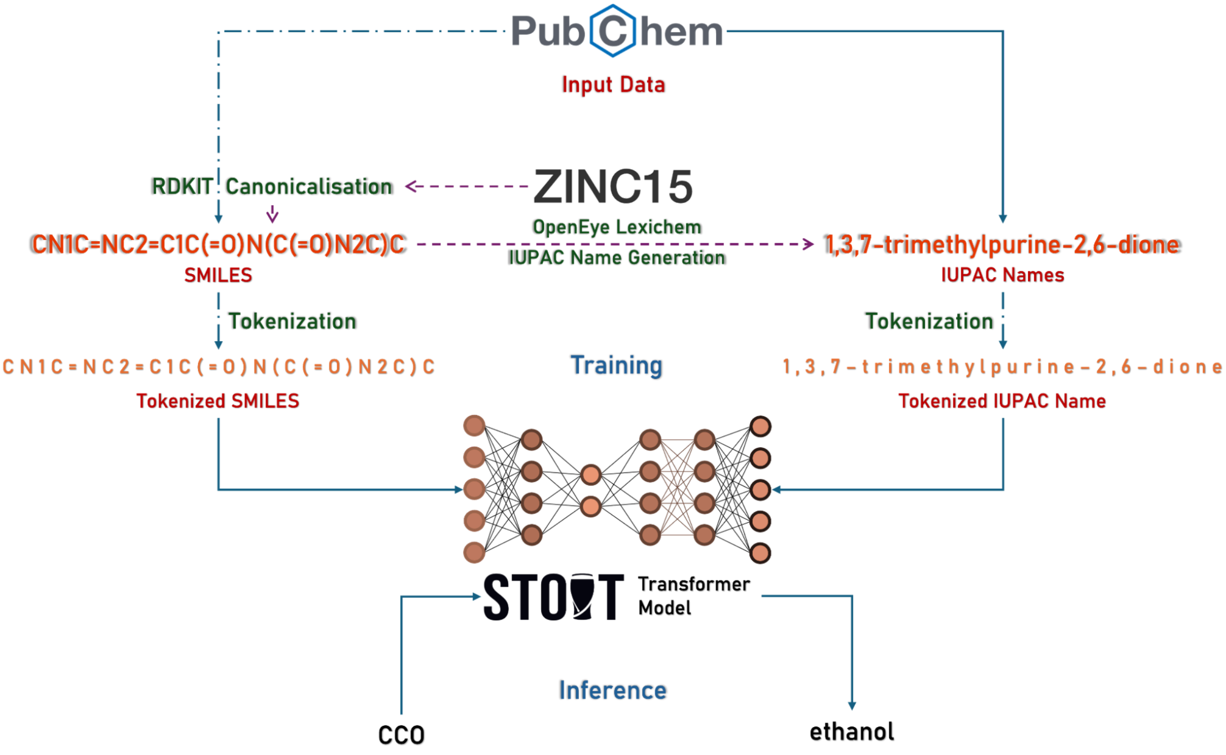


Figure 1: Workflow of SMILES to IUPAC name translator version 2.0

Methods

In STOUT V2, the main focus was improving the model and determining the appropriate token space and the maximum length of the input and output strings during training. In addition, an appropriate string representation and the assessment of the models' performance concerning training set size were investigated.

Datasets

Most of the approaches in this work were carried out using structures from PubChem, one of the largest databases for chemical structures. For large-scale training and analysis, the whole ZINC 15 database was used. PubChem was primarily selected because it publishes SMILES strings of chemical structures and their IUPAC names. PubChem contains IUPAC names systematically generated using the OpenEye Lexichem software [31].

Training Dataset Generation

The training datasets were downloaded from the respective data sources as SMILES. The downloaded SMILES strings were parsed through RDKit to obtain kekulised canonical SMILES and stereochemical information.

Experiment 1: Impact of Dataset Size and Tokenisation on Model Performance:

This experiment investigated the impact of dataset size and tokenisation on model performance using data from the PubChem database. The SMILES of each chemical structure and its system-generated IUPAC name were directly downloaded from PubChem and used as-is, with all stereochemical information retained.

From the 110 million compounds downloaded from PubChem[29] using the chemfp[32] implementation of the MaxMin[33] algorithm, approximately 51 million were selected as a subset. From this subset, a training dataset of 50 million compounds and a testing dataset of 1 million compounds were chosen. Furthermore, from the 50 million subset, an additional 11 million compound subset was selected using the MaxMin algorithm. From this, a training set of 10 million compounds and a test set of 1 million compounds were selected. As a final step, from the 11 million compounds subset, a 1 million compound training set and a 250,000 test set were selected using the MaxMin algorithm.

The tokenisation methods for SMILES followed the procedures outlined in Appendix 1 and Figure 2. IUPAC names were tokenised both by meaningful segments and character-wise, resulting in additional training datasets. Here, characterwise splitting means that each character in the IUPAC name has been separated into a token, increasing the maximum length of the tokenised IUPAC names.

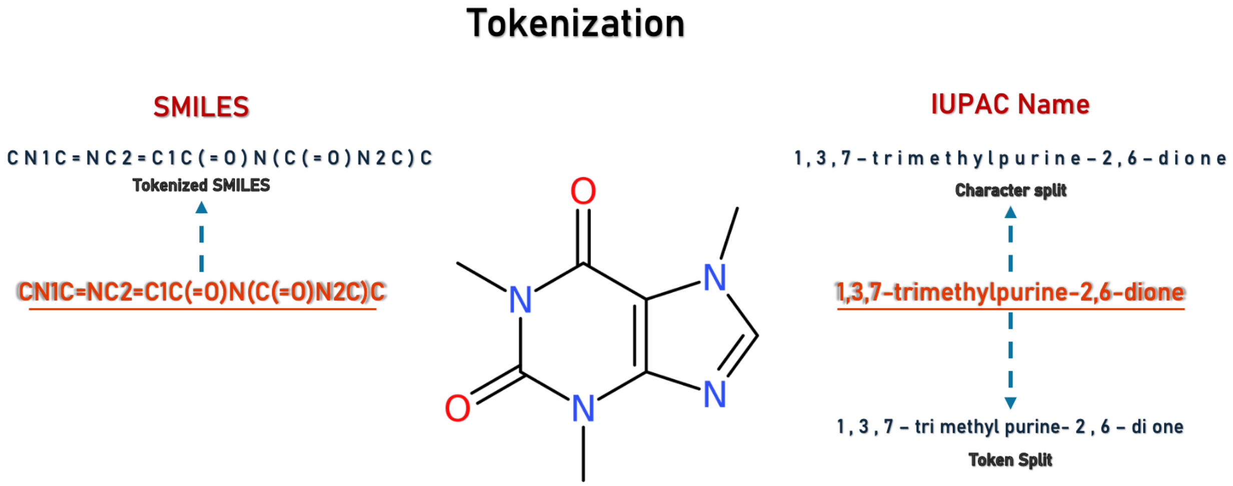

Tokenization
SMILES
C N 1 C = N C 2 = C 1 C (= O) N (C (= O) N 2 C) C
Tokenized SMILES
CN1C=NC2=C1C(=O)N(C(=O)N2C)C
O
N
N
N
N
O
IUPAC Name
1 , 3 , 7 – t r i m e t h y l p u r i n e – 2 , 6 – d i o n e
Character split
1,3,7-trimethylpurine-2,6-dione
1 , 3 , 7 – tri methyl purine- 2 , 6 – di one
Token Split

Figure 2: Comparison of tokenisation methods for the chemical compound caffeine. The figure illustrates SMILES representation of its respective tokenisation processes and contrasts IUPAC name character-level splitting with token-level splitting. The molecular structure image is included for reference.

The effect of increasing dataset size on the number of unique tokens and the maximum length of split strings was also analysed. Table 1 summarises the datasets used for training the models for SMILES strings to IUPAC name translations, with the IUPAC names split by characters or tokens. In total, 6 combinations of training datasets were generated.

Table 1: Summary of Datasets for SMILES to IUPAC Name Translation with Character and Token splits

	IUPAC name character split				**IUPAC name token split**			
	Input token size	Max length	Output token size	Max length	Input token size	Max length	Output token size	Max length
1 Mio	125	150	64	150	125	150	855	150
10 Mio	126	200	64	300	126	200	1,199	300
50 Mio	132	400	66	400	132	400	1,501	400

Experiment 2 - Training with Increased Data and Longer Sequences:

The final model selected from experiment 1 was trained on 110 million compounds and corresponding IUPAC names retrieved from PubChem. The datasets were retained as they are. However, SMILES strings longer than 600 characters were removed from the training dataset along with the corresponding IUPAC names, which retained a dataset size of 111,104,000 molecules. From this, one million molecules were selected using the chemfp implementation of the MaxMin algorithm to test the final model.

SMILES were used as input, and IUPAC names were tokenised by character. This contained 132 unique SMILES tokens with a maximum length of 600 characters, while the IUPAC name contained 68 unique tokens with a maximum length of 600 characters. Compared with the 50 million datasets, the maximum length of input and output tokens has increased by 200 characters, while the size of input and output tokens has remained relatively unchanged.

Experiment 3 - Large-Scale Training and Generalization Analysis:

The training dataset for this experiment was generated using the ZINC 15 dataset combined with PubChem molecules. The ZINC 15 dataset was downloaded as SMILES and parsed through RDKit to generate canonical SMILES with stereochemical information and kekulised. Any SMILES that did not parse through RDKit were rejected. The final set of SMILES strings was used to generate IUPAC names using the OpenEye Lexichem software. Several factors, including consistency with PubChem, influenced this decision: PubChem employs OpenEye Lexichem to generate IUPAC names. Using the same software ensured consistency with a major reference in the field. Also, OpenEye offers academic licenses, allowing us to use the software for scientific purposes without prohibitive costs.

The resulting dataset contains 883,897,289 SMILES, along with their corresponding IUPAC names. This dataset was then combined with the PubChem dataset from Experiment 4. SMILES strings with lengths above 600 characters were removed from the dataset for being less frequent in the dataset, resulting in a total of 999,637,326 molecules. The SMILES strings were tokenised as described previously, and the IUPAC names were split character-by-character. The resulting tokens included 132 SMILES tokens and 76 unique IUPAC name tokens. The maximum length of the SMILES input was set to 600 characters. In comparison, the maximum length of the IUPAC names was set to 700 characters, and after training for predictions, nearly 1,000 characters were allowed.

The SMILES to IUPAC and the IUPAC to SMILES models were trained using the same dataset.

Testing datasets

Experiment 1 - Impact of Dataset Size and Tokenisation on Model Performance:

The resulting six models were tested on three testing datasets: one with 250,000 molecules, one with 972,817 molecules, and another with 1 million molecules. To obtain the same token space as the 10 million training dataset, the test dataset has been reduced from 1 million molecules to 972,817 molecules for testing the model trained on the 10 million molecules.

The test datasets were split into meaningful segments, and character splits, resulting in three more testing datasets. Six combinations of test datasets were generated in total. Table 2 provides a comprehensive summary of the test datasets corresponding to the training datasets.

Table 2: Test Datasets Compared to Training Datasets

	IUPAC character-wise			IUPAC tokenised		
Train data size	**1 million**	**10 million**	**50 million**	**1 million**	**10 million**	**50 million**
Test data size	**265,332**	**972,817**	**1,024,000**	**265,332**	**972,817**	**1,024,000**

Experiment 2 - Training with Increased Data and Longer Sequences:
As mentioned before, a test dataset of 1 million data points was selected from the downloaded data set from PubChem. After filtering this set to match the maximum lengths of IUPAC names and SMILES, 834,774 molecules remained.

Experiment 3 - Large-Scale Training and Generalization Analysis:
In experiment 3, the dataset from experiment 2 was used to test the model.

For external validation, we tested the final model using the ChEBI database molecules not included in the Training dataset from experiment 3. Lexichem software from OpenEye was used here to generate the IUPAC names for the molecules found on the ChEBI dataset. A diverse set of 1,500 examples was selected from this dataset. SMILES with lengths exceeding 600 characters were removed, resulting in 1,485 data points. The IUPAC names for these data points were generated using OpenEye Lexichem and then predicted using the STOUT models trained on datasets of 50 million, 100 million, and approximately 1 billion samples.

TFRecord Generation

It is recommended that TFRecords be used as a basic data structure to speed up the training process using TPUs. All STOUT training datasets were converted from string format to TFRecord files and saved in 100MB chunks. This approach enhances the speed of reading TFRecords through the network. After generating the TFRecords, they were moved to Google Cloud Buckets in the same location where the TPUs were created.

Model selection

In this work, the model implemented was from the 2017 publication “Attention is All You Need” by Vaswani et al. [34]. The transformer model implemented follows a contemporary architecture for sequence-to-sequence tasks, drawing on key elements from recent advances in deep learning. Here, the model features a stack of 4 transformer layers, each incorporating a multi-head attention mechanism with 8 heads and feed-forward networks with an internal dimension (dff) of 2048 and an embedding size (d_model) of 512, as detailed in the architecture proposed by Vaswani et al. A dropout rate of 0.1 is applied to prevent overfitting.

The optimisation process employs the Adam optimiser with a custom learning rate scheduler that adjusts based on the model dimensions, as suggested in the transformer model's original implementation. A custom schedule function defines this learning rate schedule, dynamically modifying the learning rate throughout training. The loss is calculated using the Sparse Categorical Cross-Entropy loss function, with a mechanism to ignore padding tokens. This is achieved by applying a boolean mask that excludes padded elements from the loss computation, ensuring the model does not learn from these irrelevant input parts. Key metrics for evaluation include training loss, which is tracked using a Keras Mean metric, and training accuracy, which is measured using Sparse Categorical Accuracy. These metrics provide critical insights into the model's performance and guide adjustments during the training process.

The transformer is initialised with specific parameters for input and output vocabulary sizes and maximum sequence lengths for both input and target sequences. This comprehensive configuration and detailed implementation demonstrate the model's capability to handle complex language translation tasks, aligning with the current popular methodologies in neural machine translation.

Model Training

All STOUT models were completely trained on the Google Cloud Platform (GCP) using TPU V4 VMs. Training larger models requires considerable time, and using TPUs helps accelerate the training process. GCP offers a variety of TPUs; in this work, the models were trained on a TPU VM pod slice with 128 nodes. To enable training on TPU devices, all datasets were converted to TFRecord files.

TPUs (Tensor Processing Units) are specialised hardware accelerators designed to optimise machine learning workloads, specifically neural network training. The choice of TPU V4 and the configuration used in this study are among the most advanced, offering high computational power and efficiency, which are crucial for handling the complexity and scale of modern deep learning models. Converting datasets to the TFRecord format ensures efficient data loading and preprocessing, further enhancing the training performance.

The models were trained using a batch size of 96 per node and an overall batch size of 6144. Training scripts and models were written in Python 3 with Keras and TensorFlow 2.15.0, and the training was done using TensorFlow 2.15.0-pjrt.

Testing metrics

SMILES to IUPAC name translation testing:

All model test results were evaluated for identical predictions for the SMILES to IUPAC translation. The predicted IUPAC name string was compared with the original string to determine whether it matched. If the predicted string was not identical to the original, it was rejected as not identical.

Only in Experiment 3 were all the generated IUPAC names retranslated back into SMILES using OPSIN. OPSIN was chosen because it systematically parses IUPAC names with a defined set of rules. The retranslated SMILES strings were compared with the original SMILES strings to determine the similarity between the original and predicted structures. Identical structures and the Tanimoto similarity indices (using PubChem fingerprints) were evaluated.

The overall average of these values was used for the final assessment of a model's accuracy.

IUPAC name to SMILES translation testing:

The IUPAC to SMILES test results were evaluated by performing a one-to-one string match between the original and predicted SMILES to determine how similar the predicted structures were to the original structures. To assess the structural similarity, a Tanimoto similarity index calculation (using PubChem fingerprints) was performed between the original and predicted SMILES strings.

The overall average of the prediction accuracy and the Tanimoto similarity were considered for evaluating a model's accuracy.

Web App implementation

The web service implementation combines the strengths of Streamlit [35] for the front end and FastAPI for the back end, ensuring an efficient and user-friendly interface paired with a robust API.

FastAPI, chosen for its speed and efficiency, forms the backbone of the API, enabling the creation of advanced, scalable RESTful endpoints and helping to integrate with Python functions seamlessly. This choice facilitates rapid development and deployment of the backend services while ensuring high performance and reliability. The front end and the back end are containerised using Docker. Containerisation with Docker ensures consistent application execution across different environments, simplifying dependency management and streamlining the deployment process. Semantic versioning principles are meticulously applied to track changes in the codebase and toolkit versions, ensuring clear and manageable updates and backward compatibility. This holistic approach results in a seamless, high-performance web service capable of handling complex tasks efficiently, offering a seamless experience for users while maintaining robustness and scalability on the backend.

The web app constitutes functions to convert SMILES to IUPAC names, a Bulk Translate option to translate more than a single SMILES string, and the ability to retranslate them back into SMILES using OPSIN to verify the correct translation. It also offers an option to translate IUPAC names back into SMILES using STOUT or OPSIN and search for IUPAC names in PubChem using pubchempy.

Results and Discussion

The development of STOUT V2 demonstrates the transformer models' capacity and the fact that neural networks can accurately perform the non-trivial task of converting SMILES to IUPAC names. This work was solely possible due to the access of much-established rule-based systems like OpenEye’s Lexichem software. Our primary goal of this work is to make this work available to a wider audience, including those without a programming background, to simplify their IUPAC naming tasks. The development process was centred on creating an accurate, user-friendly tool that leverages the capabilities of the transformer-based models rather than competing with any existing rule-based systems.

Experiment 1 - Impact of Dataset Size and Tokenisation on Model Performance:

In this experiment, a series of models were trained with SMILES as inputs and IUPAC names as outputs, progressively increasing the dataset size.

The IUPAC names were divided into tokens using two separate methods: token-wise splitting and character-wise splitting (see appendix). Three models were trained for each combination based on 1 million, 10 million, and 50 million training data points selected from PubChem.

Table 3: SMILES to IUPAC names translation performance comparison with increasing dataset size.

	IUPAC character-wise			IUPAC token-wise		
Train data	**1 million**	**10 million**	**50 million**	**1 million**	**10 million**	**50 million**
Test data	**265,332**	**972,817**	**1,024,000**	**265,332**	**972,817**	**1,024,000**
SMILES	83.42%	92.38%	94.70%	83.46%	92.04%	94.59%

In this study, all models were tested against the respective test datasets discussed in the dataset section. Each model was compared on its ability to predict IUPAC names based only on one-to-one string matches since we are primarily interested in a model that can produce more accurate IUPAC names overall.

According to Table 3, the performance of SMILES models improves with increasing dataset size, while the increment with unique tokens and maximum length has little effect on performance. This trend can be seen with increasing dataset size, as illustrated in Figure 3.

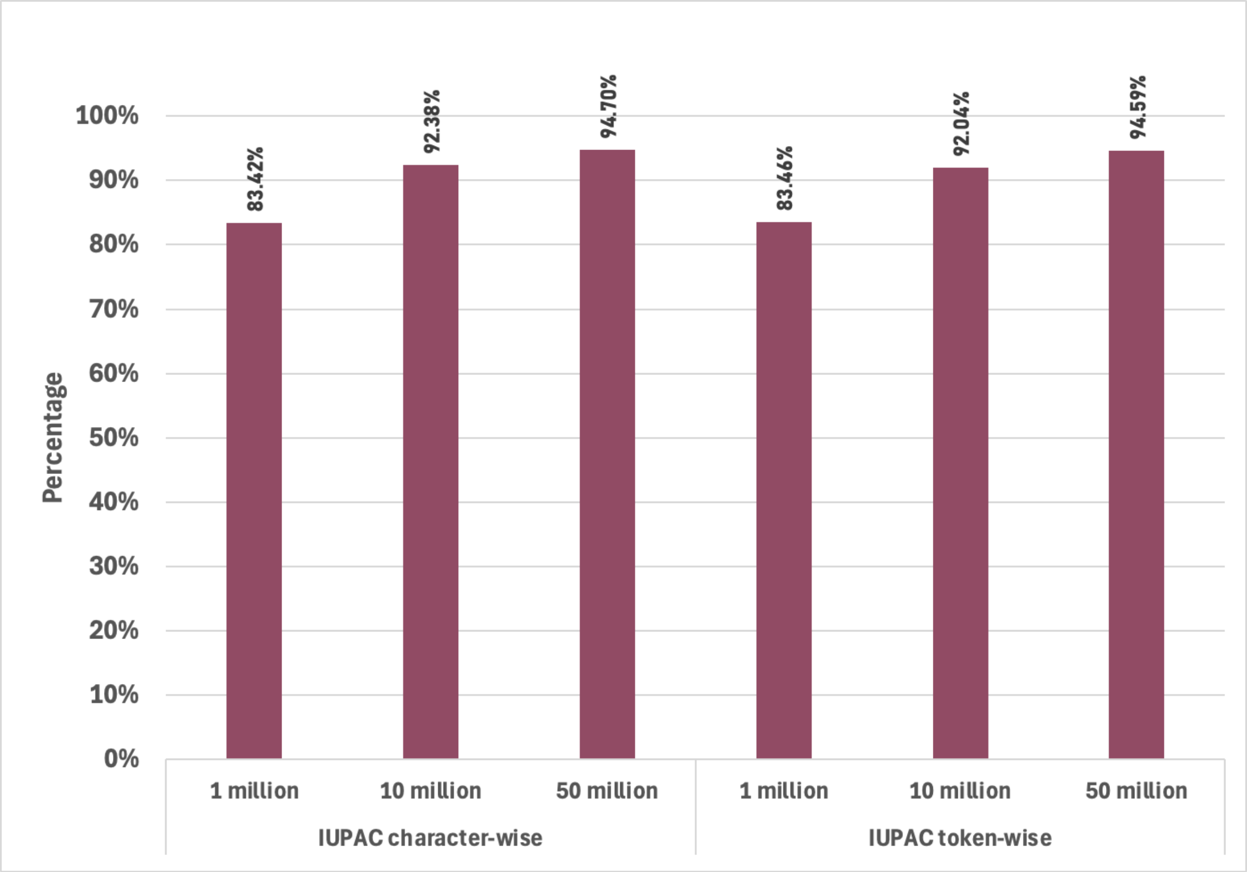


Figure 3: Comparison of SMILES representation performance using IUPAC character-wise and tokenised approaches across different dataset sizes (1 Million, 10 Million, and 50 Million molecules).

When examining IUPAC names tokenised by rules versus tokenided by character, the character-level tokenisation demonstrates better performance in models trained with more than 1 million data points. Overall, the IUPAC split by characters performs better when used as output using SMILES as input.

Experiment 2 - Training with Increased Data and Longer Sequences:

From Experiment 1, it was evident to use a Transformer model with SMILES as Input and IUPAC names character-by-character as output. With this information, a bigger dataset obtained from PubChem was used to train the same model to understand the performance gain and to see whether there is an impact on the performance by doubling the input and output maximum lengths of the strings.

The model was tested on 834,774 data points, and it was observed that it could produce 89.86% of accurate SMILES to IUPAC name translation when checked by a one-to-one string match. This reduction of accuracy was observed due to long complex names that were introduced, and the longer string prediction was also made to create errors.

Using the same dataset, when an IUPAC name-to-SMILES model was trained, the model performed with an accuracy of 94.46%, a Valid SMILES prediction of 99.54%, a Tanimoto 1.0 count of 97.47%, and an average Tanimoto of 0.99, indicating robust performance and high-quality predictions.

Experiment 3 - Large-Scale Training and Generalisation Analysis:

This experiment aimed to investigate the impact of significantly increasing the training dataset size on the performance of a transformer model for IUPAC name translation. Expanding the dataset tenfold provided the model with a substantially larger number of examples, hypothesising that this would lead to improved learning of IUPAC name translation patterns and better overall generalisation.

The model for this study was trained on a dataset of nearly one billion data points, as described in the methods section. The training was carried out on a TPU v4 VM with a 256-node pod slice, with each epoch taking an average of 15 hours and 2 minutes. A second model was also developed to perform the reverse task: IUPAC names were translated back into SMILES strings. This model used the same dataset but with the input and output string representations reversed. The approach was used to explore bidirectional conversion between chemical structures and their standardised names.

The model was then evaluated using the test dataset from Experiment 2, yielding a performance of 83.52% for exact string match in SMILES to IUPAC name translation.

For IUPAC to SMILES performance on the same test set, the model achieved a test accuracy of 90.46% with a Valid SMILES prediction of 99.1%, Tanimoto 1.0 count of 95.11% and a Tanimoto average of 0.99.

This observed decrease in accuracies compared to Experiment 2 could be due to the larger and more diverse dataset. While this approach enhances generalisability, it can also lead to a slight decrease in performance on any single test set. The exposure to a broader range of examples increased the likelihood of generating errors. Adding additional examples from sources other than PubChem may have caused a data distribution shift [36,37]. This shift could have introduced training examples that differ significantly from those in the test dataset, potentially leading to reduced model performance on the original test set.

By increasing the maximum length of IUPAC names in training and predictions, the model can learn from longer complex names, but it also increases prediction errors.

A benchmark study using the ChEBI dataset was conducted to understand whether the model improved overall. This experiment checked the models' ability to generalise and handle previously unseen data, thus providing insights into their real-world applicability.

This test assessed the model's ability to generate correct IUPAC names by comparing the predicted and original names using exact string matching.

To ensure that the predicted IUPAC names represent a valid chemical structure, a two-step verification process was implemented. The predicted IUPAC names were reverse-translated back into SMILES notation using OPSIN, and the resulting SMILES strings were then compared to the original input SMILES strings to verify how similar the original and the retranslated structures are using Tanimoto similarity using PubChem fingerprints incorporated in Chemistry Development Kit (CDK) [38,39]. See Figure 4 for more details.

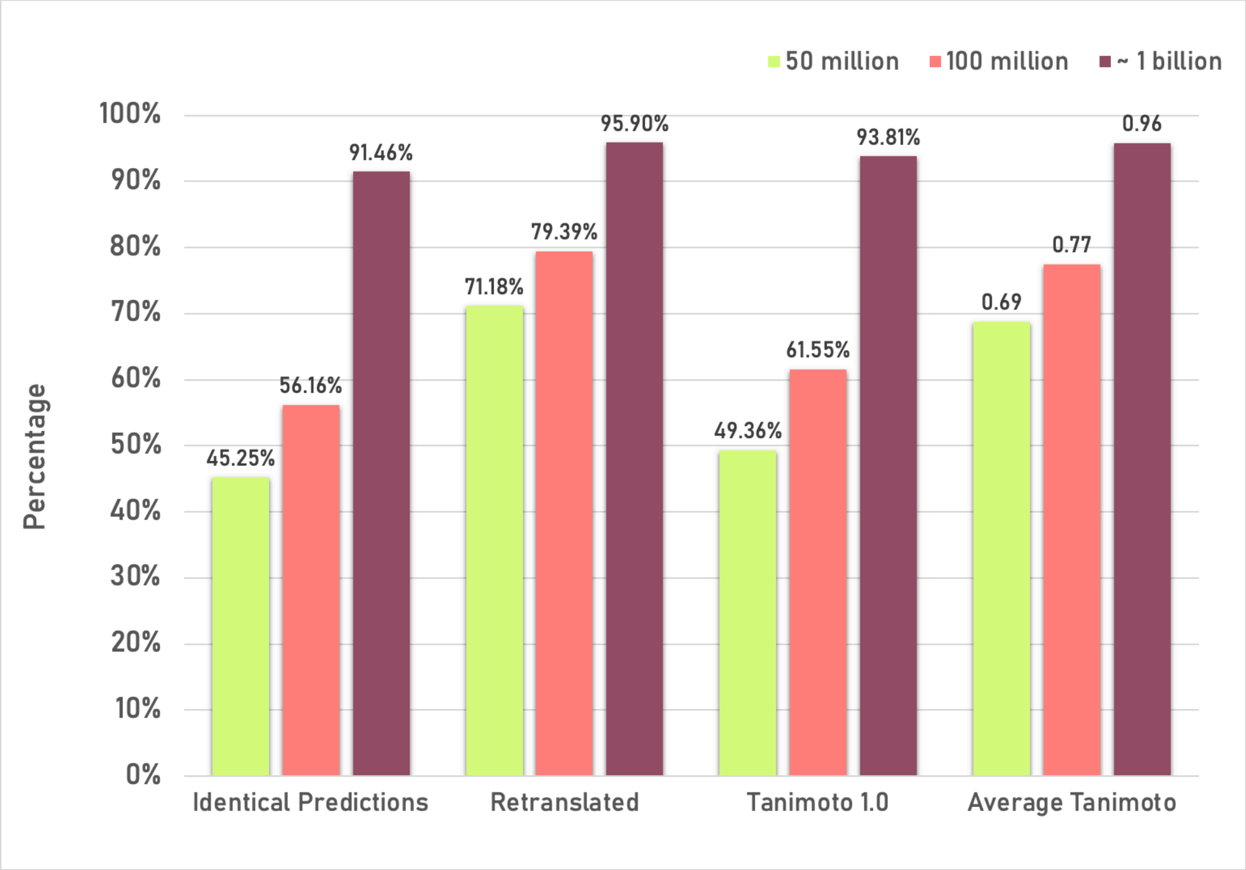


Figure 4: Performance of the SMILES to IUPAC name translation model with increasing dataset size. Comparing the identical predictions, Percentage of IUPAC names retranslation, Tanimoto 1.0 percentage and Average Tanimoto for models.

Looking at the results in Figure 4, the identical predictions and the retranslation accuracy increase with more training data, implying that the model's predicted IUPAC names become more consistent and fluent with more extensive training. Tanimoto 1.0 and Average Tanimoto also improve significantly with more training data, reflecting a higher quality in the model's predictions as it is trained on larger datasets. Overall, this suggests that increasing the training data improves the model's generalisability, and one could get better and more accurate IUPAC names.

To further understand the implications of the length of predicted IUPAC names, the predictions were categorised according to the input SMILES length, and the prediction quality was then analysed in detail.

As shown in Table 4, the prediction quality of the model trained on 100 million data points reduces as the length of the SMILES increases. In contrast, the model trained on 1 billion data points maintains better prediction quality across accuracy, retranslation using OPSIN, and average Tanimoto similarity despite the increased complexity of the IUPAC names.

The retranslation fails for longer IUPAC names generated by the 100 million data points model due to errors such as missing spaces, incorrect group order, formatting issues, incorrect valency, and other OPSIN-related errors encountered during the retranslation of these names. As seen in Figure 5, the poor quality of the retranslation decreases the overall average Tanimoto similarity.

Table 4: Accuracy and Tanimoto Similarities by SMILES String Lengths for Different Data Sizes

		Accuracy		Retranslated		Average Tanimoto	
Group	characters length	100 million	1 billion	100 million	1 billion	100 million	1 billion
1	0-60	92%	100%	99%	99%	0.98	0.99
2	61-120	84%	99%	96%	98%	0.94	0.98
3	121-180	62%	93%	84%	97%	0.82	0.97
4	181-240	47%	89%	72%	94%	0.71	0.94
5	241-300	23%	84%	68%	93%	0.65	0.93
6	301-360	8%	78%	42%	96%	0.39	0.96
7	361-420	26%	59%	32%	97%	0.31	0.96
8	421-480	18%	60%	27%	100%	0.26	1.00
9	481-540	0%	20%	40%	60%	0.40	0.60
10	541-600	0%	100%	0%	100%	0.00	1.00

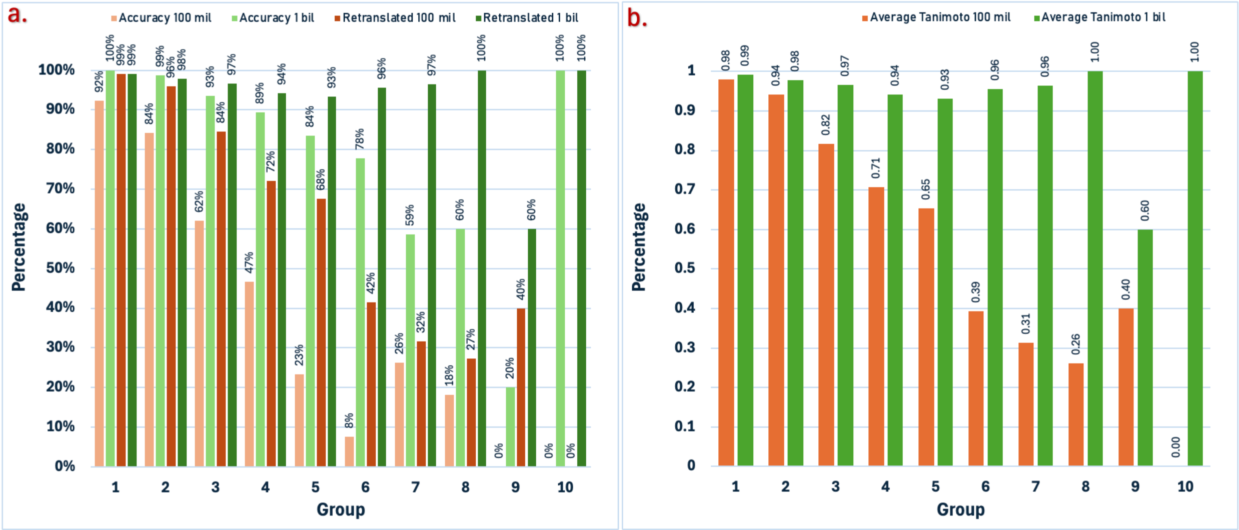


Figure 5: a) IUPAC Name Prediction Accuracy and Retranslation Quality Across SMILES String Length Groups; b) Average Tanimoto Similarities Compared Across Various SMILES String Length Groups

Similarly, the IUPAC to SMILES translation models, trained on 100 million and approximately 1 billion data points, were also compared using the ChEBI dataset. As seen in Table 5 the models trained on 100 million and 1 billion data points perform almost similarly. However, the model trained on 1 billion data points, which included IUPAC names longer than 600 characters, had to learn more complex IUPAC names. This complexity slightly reduced the model's performance compared to the 100 million data point model. In general, the performance of both models is comparable.

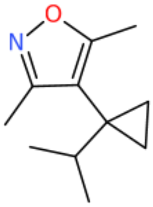
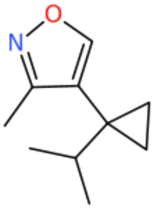
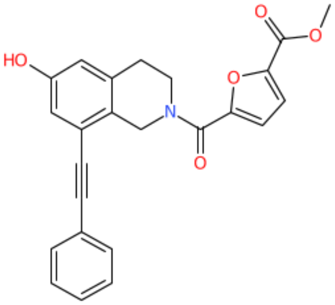
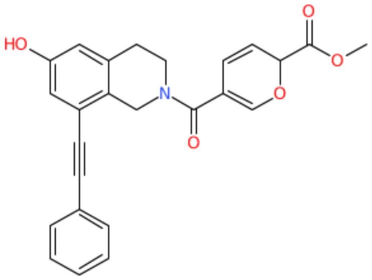
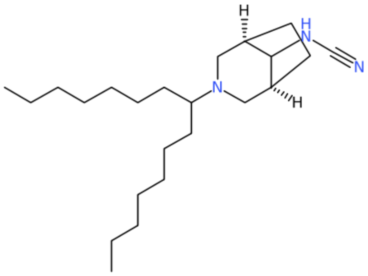
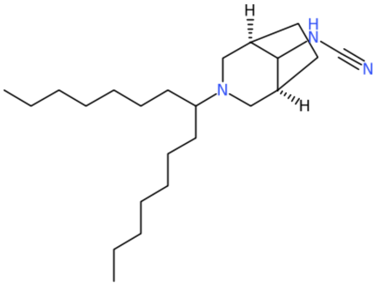
Table 5: Performance of IUPAC to SMILES models trained on increasing training data

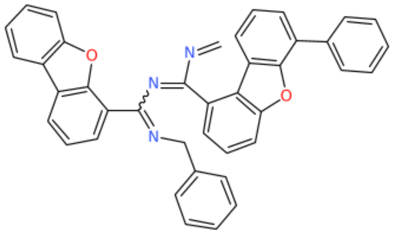
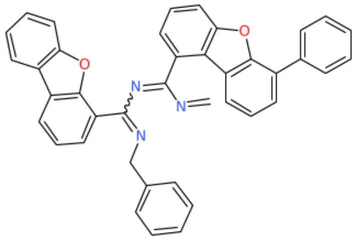
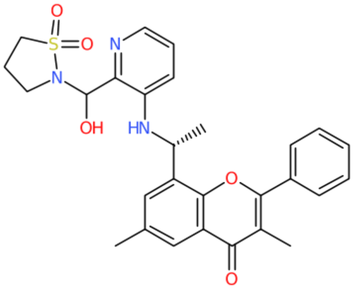
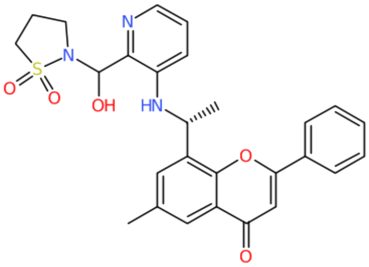
Training data size	Valid SMILES	Identical Predictions	Tanimoto 1.0	Average Tanimoto
100 million	99.80%	96.97%	99.12%	0.99
~ 1 billion	99.60%	96.23%	99.06%	0.99

In this work, the IUPAC names predicted to be different from the original IUPAC names were entirely discarded during the accuracy calculation, as chemists prefer only 100% accurate IUPAC names. Slightly different names are still considered incorrect.

To understand how well the model handles slightly different IUPAC names, names that did not match the original were retranslated using OPSIN, and the resulting structures were compared. Table 6 summarises a few examples:

Table 6: The robustness of the model was evaluated by examining incorrectly predicted IUPAC names that were retranslated into valid chemical structures.

Original Structure	Predicted Structure	Tanimoto Similarity
 3**,5-dimethyl**-4-(1-propan-2-ylcyclopropyl)-1,2-oxazole	 3-**methyl**-4-(1-propan-2-ylcyclopropyl)-1,2-oxazole	0.89
 methyl5-[6-hydroxy-8-(2-phenylethynyl)-3,4-dihydro-1H-isoquinoline-2-carbonyl]**furan-2-**carboxylate	 methyl5-[6-hydroxy-8-(2-phenylethynyl)-3,4-dihydro-1H-isoquinoline-2-carbonyl]**pyran-2-**carboxylate	0.8
 [(**1R,5S**)-3-pentadecan-8-yl-3-	 [(**1S,5R**)-3-pentadecan-8-yl-3-azabicyclo[3.2.1]oc	1.0

azabicyclo[3.2.1]octan-8-yl]cyanamide	tan-8-yl]cyanamide	
 N'**-(N-benzyl-C-dibenzofuran-4-ylcarbonimidoyl)-N-methylidene-6-phenyl**dibenzofuran-**1**-carboximidamide	 N'-benzyl**-N-[(methylideneamino)-(6-phenyldibenzofuran-1-yl)methylidene]**dibenzofuran-**4**-carboximidamide	1.0
 8-[(1R)-1-[[2-[(1,1-dioxo-1,2-thiazolidin-2-yl)-hydroxymethyl]pyridin-3-yl]amino]ethyl]-**3,6-dimethyl**-2-phenylchromen-4-one	 8-[(1R)-1-[[2-[(1,1-dioxo-1,2-thiazolidin-2-yl)-hydroxymethyl]pyridin-3-yl]amino]ethyl]**-6-methyl**-2-phenylchromen-4-one	0.99

As shown in Table 6, although the IUPAC names generated by the model are incorrect compared to the original IUPAC names, the resulting chemical structures are often very similar. In some cases, they are identical. This indicates that the model makes only minor errors in predicting the IUPAC names. Upon visual inspection of the depicted structures, these slight inaccuracies in naming can be readily identified and corrected by a trained chemist. This can significantly improve the process of naming chemical compounds. This model's capability makes it a valuable tool in assisting chemists with the often time-consuming task of compound nomenclature.

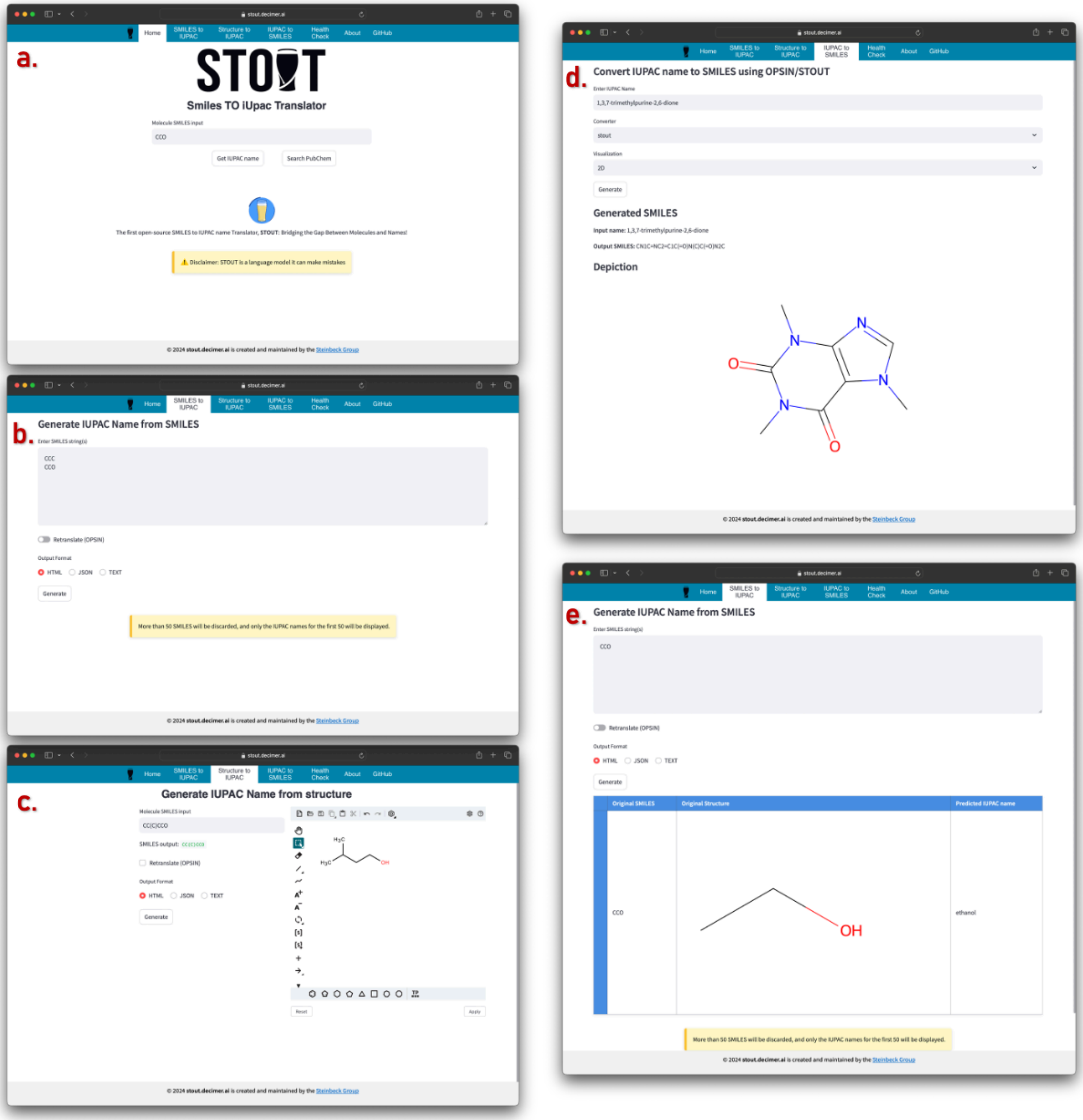
Some names failed to retranslate completely due to errors in the predicted names and limitations within OPSIN. Examples of these errors are listed in Table 7.

Table 7: Examples of OPSIN retranslation errors:

Original IUPAC Name	Predicted IUPAC Name	OPSIN error messages
tert-butyl 12-(2-aminoquinolin-8-yl)-13-fluoro-16-methoxy-8-methyl-9-oxa-2,5,11,15,17-pentazatricyclo[8.7.1.014,18]octadeca-1(17),10,12,14(18),15-pentaene-5-carboxylate	tert-butyl 15-(2-aminoquinolin-8-yl)-14-fluoro-17-methoxy-11-methyl-10-oxa-2,5,8,16,18-pentazatricyclo[7.7.1.013,17]heptadeca-1(16),9(17),13(17),14-tetraene-5-carboxylate	Could not find the atom with locant 18.
tert-butyl (8S)-12-chloro-13-fluoro-16-methoxy-6,8-dimethyl-9-oxa-2,5,11,15,17-pentazatricyclo[8.7.1.014,18]octadeca-1(17),10,12,14(18),15-pentaene-5-carboxylate	tert-butyl (3S)-16-chloro-15-fluoro-13-methoxy-3,5-dimethyl-2-oxa-6,9,11,12,14-pentazatricyclo[8.6.1.015,17]heptadeca-1(16),10,12,14-tetraene-6-carboxylate	Atom is in unphysical valency state! Element: C valency: 5
N'-[[methyl-[1-(6-spiro[1,2-dihydrofluorene-9,9'-6,7-dihydroxanthene]-3'-ylpyridin-3-yl)ethyl]amino]-phenylmethyl]benzenecarboximidamide	N'-[[methyl-[1-(6-spiro[1,2-dihydrotriphenylene-9,9'-6,7-dihydroxanthene]-3'-ylpyridin-3-yl)ethyl]amino]-phenylmethyl]benzenecarboximidamide	Failed to assign all double bonds! (Check that indicated hydrogens have been appropriately specified)
(7S,10S,13S)-N-[(1S)-1-cyclopropylethyl]-10-(2-morpholin-4-ylethyl)-9,12-dioxo-13-(2-oxopyrrolidin-1-yl)-2-oxa-8,11-diazabicyclo[13.3.1]nonadeca-1(18),15(19),16-triene-7-carboxamide	(7S,10S,13S)-N-[(1S)-1-cyclopropylethyl]-13-(2-morpholin-4-ylethyl)-9,12-dioxo-16-(2-oxopyrrolidin-1-yl)-2-oxa-8,11-diazabicyclo[16.3.1]docosa-1(21),18(22),19-triene-7-carboxamide	Could not find atom that: <stereoChemistry locant="10" type="RorS" value="S" stereoGroup="Abs">10S</stereoChemistry> appeared to be referring to
(4R,6R)-6-[(1R)-1-[[2-[amino(methanimidoyl)amino]acetyl]amino]ethyl]-4-methyl-7-oxo-3-[(3S,5S)-5-[3-(1,2,4-triazol-4-yl)azetidine-1-carbonyl]pyrrolidin-3-yl]sulfanyl-1-azabicyclo[3.2.0]hept-2-ene-2-carboxylic acid	4R,6R)-4-[(1R)-1-[[2-[amino(methanimidoyl)amino]acetyl]amino]ethyl]-6-methyl-2-oxo-3-[(3S,5S)-5-[3-(1,2,4-triazol-4-yl)azetidine-1-carbonyl]piperidin-3-yl]sulfanyl-1-azabicyclo[3.2.0]hept-3-ene-8-carboxylic acid	Suffix: imido does not apply to the group it was associated with (type: standardGroup) according to suffixApplicability.xml

Table 7 reveals several issues in automated chemical nomenclature using deep learning. While many predicted names were successfully retranslated, some failed due to errors in the generated names and limitations within OPSIN. The errors mentioned above indicate the complexities of automated IUPAC name generation and interpretation, particularly for intricate organic molecules. Despite these challenges, the model could generate structurally similar names in many cases. However, the results underscore the continued importance of expert verification in deep learning-based chemical nomenclature tasks and point to specific areas for improvement in name-generation models and parsing software like OPSIN.

STOUT Web Application



a.

Smiles TO iUpac Translator

Molecule SMILES input: CCO

The first open-source SMILES to IUPAC name Translator, **STOUT**: Bridging the Gap Between Molecules and Names!

⚠ Disclaimer: STOUT is a language model it can make mistakes

© 2024 **stout.decimer.ai** is created and maintained by the Steinbeck Group

b.

Generate IUPAC Name from SMILES

Enter SMILES string(s):
CCC
CCO

Retranslate (OPSIN)

Output Format: ● HTML ○ JSON ○ TEXT

More than 50 SMILES will be discarded, and only the IUPAC names for the first 50 will be displayed.

© 2024 **stout.decimer.ai** is created and maintained by the Steinbeck Group

c.

Generate IUPAC Name from structure

Molecule SMILES input: CC(C)CCO

SMILES output: CC(C)CCO

□ Retranslate (OPSIN)

Output Format: ● HTML ○ JSON ○ TEXT

© 2024 **stout.decimer.ai** is created and maintained by the Steinbeck Group

d.

Convert IUPAC name to SMILES using OPSIN/STOUT

Enter IUPAC Name: 1,3,7-trimethylpurine-2,6-dione

Converter: stout

Visualization: 2D

Generated SMILES

Input name: 1,3,7-trimethylpurine-2,6-dione

Output SMILES: CN1C=NC2=C1C(=O)N(C)C(=O)N2C

Depiction

© 2024 **stout.decimer.ai** is created and maintained by the Steinbeck Group

e.

Generate IUPAC Name from SMILES

Enter SMILES string(s): CCO

Retranslate (OPSIN)

Output Format: ● HTML ○ JSON ○ TEXT

Original SMILES	Original Structure	Predicted IUPAC name
CCO	OH	ethanol

More than 50 SMILES will be discarded, and only the IUPAC names for the first 50 will be displayed.

© 2024 **stout.decimer.ai** is created and maintained by the Steinbeck Group

Figure 6: The STOUT web application interface, featuring a home page for single SMILES input (a), batch processing for up to 50 SMILES strings (b) example (e), the Ketcher editor for drawing structures (c), and the IUPAC to SMILES translation with structure depiction using CDK (d).

This work also includes a web application designed to support the automated naming of chemical structures for chemists or chemical database curators. The Web App, leveraging the STOUT models, allows users to input SMILES strings of the chemical compounds and receive IUPAC names. It also provides a feature to check whether the compound is already catalogued in PubChem, displaying the corresponding IUPAC name, if available, and retrieved directly from PubChem.

The Web App supports bulk submission of up to 50 SMILES strings, facilitating the generation of IUPAC names for multiple compounds simultaneously. These generated names can be retranslated using OPSIN and visualised using CDK at the backend, enabling users to compare the original and predicted structures. The results can be downloaded in HTML, JSON, or text formats.

It also includes a Ketcher[40] window for drawing chemical compounds and obtaining their IUPAC names. For IUPAC names to SMILES conversion, users can choose to utilise either STOUT or OPSIN, with the Web App depicting the resulting structure for visual analysis of the predictions. This web application is completely containerised, so those who do not want to use the web application available online can spin up their web application locally using the Docker container.

Conclusion

This work presents STOUT V2.0, an improved successor for SMILES to IUPAC name Translator. This work represents a significant advancement in automated deep learning-based IUPAC name generation. It was, however, only possible due to the availability of IUPAC names generated by deterministic naming algorithms, in our case, by OpenEye's Lexichem software. A substantial improvement was achieved by leveraging the transformer-based models for deep learning.

In terms of tokenisation strategy, it was discovered that character-split tokenisation of IUPAC names yielded better overall accuracy compared to word-split tokenisation. These character-split models also demonstrated faster training times and utilised fewer unique tokens.

We conducted large-scale training on a dataset of nearly one billion compounds to improve model generalisability. While we observed a slight decrease in accuracy on specific test datasets, the overall performance on a comprehensive benchmark dataset showed significant improvement. This suggests the increased training data enhanced the model's ability to handle a wider range of chemical structures. Moreover, a bigger model, such as the Large Language Model (LLM), could be used along with bigger training data.

By expanding the input token context window to 700 characters and extending the prediction window to 1000 characters, STOUT V2.0 can now generate longer, more complex IUPAC names. This expansion is crucial for accurately naming larger molecular structures. Furthermore, we leveraged the latest TPU VMs to train the models, significantly reducing training time for these complex models with extensive datasets. These improvements in training hardware accelerate the development process and enable one to use more data and complex model architectures to train and test.

Our user-friendly web application further enhances the accessibility and usability of STOUT V2.0, which is now accessible to users with limited programming knowledge. The application includes integrated visualisation features that enable trained chemists to identify and promptly correct minor errors in generated IUPAC names. This feature not only improves the accuracy of the output but also serves as a valuable educational tool for understanding the nuances of IUPAC nomenclature.

By making the model checkpoints, weights, and fully documented source code available as open-source resources, we aim to promote unrestricted use and encourage further development within the field. This will allow researchers in this field to build upon our work and adapt it to their specific needs.

List of abbreviations

BLEU - BilinguaL Evaluation Understudy
CDK - CHemistry Development Kit
CPU - Central Processing Unit
GRU - Gated Recurrent Units
IUPAC - International Union of Pure and Applied Chemistry
LLM - Large Language Model
NLP - Natural Language Processing
OCSR - Optical Chemical Structure Recognition
OPSIN - Open Parser for Systematic IUPAC Nomenclature
REST - REpresentational State Transfer
RNN - Recurrent Neural Networks
SDF - Structure-Data File
SMILES - Simplified Molecular Input Line Entry Specification
STOUT - Smiles TO iUpac Translator
TFRecord - Tensor Flow Record
TPU - Tensor Processing Units
V - Version
VM - Virtual Machine
XML - Extensible Markup Language

Availability and requirements

- **Project name:** Smiles TO iUpac Translator
- **Project home page:** https://github.com/Kohulan/Smiles-TO-iUpac-Translator
    - Web Application: https://github.com/Kohulan/STOUT_WebApp
    - Models and checkpoints: https://zenodo.org/records/13318286
- **Current version:** v2.0.7
- **DOI of the archived current release:**
- **Operating system(s):** Independent
- **Programming language:** Python 3
- **Requirements:**
    - API calls:
        - Internet connection and command line interface or a web browser
    - Run locally:
        - Docker - To use the STOUT Web app as a Docker container
        - Conda environment - to use STOUT natively without Docker as a Python package
    - Dependencies (managed by Docker/Conda):
        - Python packages: uvicorn>=0.15.0,<0.16.0, fastapi>=0.80.0, fastapi-pagination==0.10.0, fastapi-versioning>=0.10.0, pystow>=0.4.9, unicodedata2==15.0.0, tensorflow==2.15.0-pjrt, httpx>=0.24.1
- **Licence:** MIT
- **Documentation:**
    - Home page: https://stout.decimer.ai
    - API: https://stout.api.decimer.ai/latest/docs
    - Python docs: https://stout-web-application.readthedocs.io/en/latest/index.html
- **Any restrictions to use by non-academics:** None
- **PyPi Package:** https://pypi.org/project/STOUT-pypi/

Declarations

Ethics approval and consent to participate

Not applicable

Consent for publication

The authors have given their consent for the work to be published.

Competing interests

AZ is co-founder of GNWI - Gesellschaft für naturwissenschaftliche Informatik mbH, Dortmund, Germany. The remaining authors declare no financial and non-financial competing interests.

Funding

Open Access funding enabled and organised by Projekt DEAL. KR acknowledges the funding by the German Research Foundation under project number 239748522- SFB 1127 ChemBioSys (Project INF)., and C.S. acknowledge the funding by the Carl-Zeiss-Foundation for this work.

Authors' contributions

KR initiated, designed, tested, applied and validated the software features. KR, AZ, and CS wrote the manuscript. CS and AZ conceived the project and supervised the work. All authors contributed to and approved the manuscript.

Acknowledgements

We greatly acknowledge OpenEye for granting us an academic license to their Lexichem software, without which this work would not have been possible.
We are grateful that Google makes free computing time available to us on its TPU Research Cloud infrastructure. KR acknowledges the support through Cloud TPUs from Google's TPU Research Cloud (TRC). KR acknowledges Google Cloud Research for the research credits provided for this work under the Google Cloud Research Credits program with the award GCP19980904.

References

1. Panico, R.; Powell, W.H.; Richer, J.-C. *A Guide to IUPAC Nomenclature of Organic Compounds: Recommendations 1993 (including Revisions, Published and Hitherto Unpublished, to the 1979 Edition of Nomenclature of Organic Chemistry*; Wiley-Blackwell, 1993; ISBN 9780632034888.
2. *Nomenclature of Organic Chemistry: IUPAC Recommendations and Preferred Names 2013*; Royal Society of Chemistry, 2014; ISBN 9780854041824.
3. International Union of Pure and Applied Chemistry *Nomenclature of Inorganic Chemistry: IUPAC Recommendations 2005*; Royal Society of Chemistry, 2005; ISBN 9780854044382.
4. Inczedy, J.; Lengyel, T.; Ure, A.M.; Gelencsér, A.; Hulanicki, A.; Others Compendium of Analytical Nomenclature. *Hoboken: Blackwell Science* **1998**.
5. Tinley, E.H. *Naming Organic Compounds: A Guide to the Nomenclature Used in Organic Chemistry*; 2013; ISBN 9781258629830.
6. Werd, S. Mnova 15.0.1 Available online: https://mestrelab.com/download_file/mnova-15-0-1/ (accessed on 1 July 2024).
7. Molconvert Available online: https://docs.chemaxon.com/display/lts-lithium/molconvert.md (accessed on 1 July 2024).
8. Convert Chemical Structures and Chemical Names Available online: https://www.eyesopen.com/lexichem-tk (accessed on 1 July 2024).
9. Generate IUPAC Names for Chemical Structures Available online:

https://www.acdlabs.com/products/name/ (accessed on 1 July 2024).

10. Website Available online: ChemAxon - Software Solutions and Services for Chemistry & Biology. https://www.chemaxon.com.
11. Website Available online: OpenEye Toolkits 2023.1. OpenEye, Cadence Molecular Sciences, Santa Fe, NM. http://www.eyesopen.com.
12. Dargan, S.; Kumar, M.; Ayyagari, M.R.; Kumar, G. A Survey of Deep Learning and Its Applications: A New Paradigm to Machine Learning. *Arch. Comput. Methods Eng.* **2020**, *27*, 1071–1092, doi:10.1007/s11831-019-09344-w.
13. Taye, M.M. Understanding of Machine Learning with Deep Learning: Architectures, Workflow, Applications and Future Directions. *Computers* **2023**, *12*, 91, doi:10.3390/computers12050091.
14. Khan, W.; Daud, A.; Khan, K.; Muhammad, S.; Haq, R. Exploring the Frontiers of Deep Learning and Natural Language Processing: A Comprehensive Overview of Key Challenges and Emerging Trends. *Natural Language Processing Journal* **2023**, *4*, 100026, doi:10.1016/j.nlp.2023.100026.
15. Yang, S.; Wang, Y.; Chu, X. A Survey of Deep Learning Techniques for Neural Machine Translation. *arXiv [cs.CL]* 2020.
16. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models Are Few-Shot Learners. *arXiv [cs.CL]* 2020.
17. Chang, E.Y. Examining GPT-4's Capabilities and Enhancement with SocraSynth. In Proceedings of the 2023 International Conference on Computational Science and Computational Intelligence (CSCI); IEEE, December 13 2023; pp. 7–14.
18. 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. *Chem. Sci.* **2018**, *9*, 6091–6098, doi:10.1039/c8sc02339e.
19. Rajan, K.; Brinkhaus, H.O.; Agea, M.I.; Zielesny, A.; Steinbeck, C. DECIMER.ai: An Open Platform for Automated Optical Chemical Structure Identification, Segmentation and Recognition in Scientific Publications. *Nat. Commun.* **2023**, *14*, 5045, doi:10.1038/s41467-023-40782-0.
20. Blanco-González, A.; Cabezón, A.; Seco-González, A.; Conde-Torres, D.; Antelo-Riveiro, P.; Piñeiro, Á.; Garcia-Fandino, R. The Role of AI in Drug Discovery: Challenges, Opportunities, and Strategies. *Pharmaceuticals* **2023**, *16*, doi:10.3390/ph16060891.
21. Ertl, P.; Lewis, R.; Martin, E.; Polyakov, V. In Silico Generation of Novel, Drug-like Chemical Matter Using the LSTM Neural Network. *arXiv [cs.LG]* 2017.
22. Olivecrona, M.; Blaschke, T.; Engkvist, O.; Chen, H. Molecular de-Novo Design through Deep Reinforcement Learning. *J. Cheminform.* **2017**, *9*, 48, doi:10.1186/s13321-017-0235-x.
23. Ivanenkov, Y.A.; Polykovskiy, D.; Bezrukov, D.; Zagribelnyy, B.; Aladinskiy, V.; Kamya, P.; Aliper, A.; Ren, F.; Zhavoronkov, A. Chemistry42: An AI-Driven Platform for Molecular Design and Optimization. *J. Chem. Inf. Model.* **2023**, *63*, 695–701, doi:10.1021/acs.jcim.2c01191.
24. Baum, Z.J.; Yu, X.; Ayala, P.Y.; Zhao, Y.; Watkins, S.P.; Zhou, Q. Artificial Intelligence in Chemistry: Current Trends and Future Directions. *J. Chem. Inf. Model.* **2021**, *61*,

3197–3212, doi:10.1021/acs.jcim.1c00619.

25. Handsel, J.; Matthews, B.; Knight, N.J.; Coles, S.J. Translating the InChI: Adapting Neural Machine Translation to Predict IUPAC Names from a Chemical Identifier. *J. Cheminform.* **2021**, *13*, 79, doi:10.1186/s13321-021-00535-x.
26. Rajan, K.; Zielesny, A.; Steinbeck, C. STOUT: SMILES to IUPAC Names Using Neural Machine Translation. **2020**.
27. Krasnov, L.; Khokhlov, I.; Fedorov, M.V.; Sosnin, S. Transformer-Based Artificial Neural Networks for the Conversion between Chemical Notations. *Sci. Rep.* **2021**, *11*, 14798, doi:10.1038/s41598-021-94082-y.
28. Lowe, D.M.; Corbett, P.T.; Murray-Rust, P.; Glen, R.C. Chemical Name to Structure: OPSIN, an Open Source Solution. *J. Chem. Inf. Model.* **2011**, *51*, 739–753, doi:10.1021/ci100384d.
29. Kim, S.; Chen, J.; Cheng, T.; Gindulyte, A.; He, J.; He, S.; Li, Q.; Shoemaker, B.A.; Thiessen, P.A.; Yu, B.; et al. PubChem 2023 Update. *Nucleic Acids Res.* **2023**, *51*, D1373–D1380, doi:10.1093/nar/gkac956.
30. *Chemaxon*; Benoit, K., Ed.; Dict, 2011; ISBN 9786136645247.
31. Molecular Modeling Software Available online: http://www.eyesopen.com. (accessed on 5 August 2024).
32. Dalke, A. The Chemfp Project. *J. Cheminform.* **2019**, *11*, 76, doi:10.1186/s13321-019-0398-8.
33. Ashton, M.; Barnard, J.; Casset, F.; Charlton, M.; Downs, G.; Gorse, D.; Holliday, J.; Lahana, R.; Willett, P. Identification of Diverse Database Subsets Using Property-Based and Fragment-Based Molecular Descriptions. *Quant. struct.-act. relatsh.* **2002**, *21*, 598–604, doi:10.1002/qsar.200290002.
34. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv [cs.CL]* 2017.
35. Khorasani, M.; Abdou, M.; Fernández, J.H. *Web Application Development with Streamlit*; Apress;.
36. Quinonero-Candela, J.; Sugiyama, M.; Schwaighofer, A.; Lawrence, N.D. *Dataset Shift in Machine Learning*; MIT Press, 2022; ISBN 9780262545877.
37. Pan, S.J.; Yang, Q. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359, doi:10.1109/TKDE.2009.191.
38. Willighagen, E.L.; Mayfield, J.W.; Alvarsson, J.; Berg, A.; Carlsson, L.; Jeliazkova, N.; Kuhn, S.; Pluskal, T.; Rojas-Chertó, M.; Spjuth, O.; et al. The Chemistry Development Kit (CDK) v2.0: Atom Typing, Depiction, Molecular Formulas, and Substructure Searching. *Journal of Cheminformatics* 2017, *9*.
39. Steinbeck, C.; Han, Y.; Kuhn, S.; Horlacher, O.; Luttmann, E.; Willighagen, E. The Chemistry Development Kit (CDK): An Open-Source Java Library for Chemo- and Bioinformatics. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 493–500, doi:10.1021/ci025584y.
40. Karulin, B.; Kozhevnikov, M. Ketcher: Web-Based Chemical Structure Editor. *J. Cheminform.* **2011**, *3*, 1–1, doi:10.1186/1758-2946-3-S1-P3.
41. Chollet, F.; Others Keras Available online: https://keras.io.

Appendix

1. tokenisation:

SMILES tokenisation: The same tokenisation strategy for SMILES strings as outlined in our previous work[19], utilising the Keras[41] tokeniser for generating the token sequences. The SMILES strings were split into meaningful tokens according to the following set of rules:

- Each heavy atom (e.g., "C", "Si", "Au") was treated as a separate token.
- Each open and closed bracket (i.e., "(", ")", "[", "]") was tokenised individually.
- Each bond symbol ("=", "#") was considered as a separate token.
- The characters ".", "-", "+", "", "/", "@", "%", and "*" were split into individual tokens.
- Each single-digit number was tokenised separately.

After splitting the SMILES strings into tokens, a "<start>" token was added at the beginning of each sequence, and an "<end>" token was appended at the end.

DeepSMILES tokenisation: Similar to SMILES tokenisation, the DeepSMILES were also split into meaningful tokens using the same ruleset.

IUPAC Names tokenisation: IUPAC names were split after certain characters like opening brackets ("(", "{", "["), closing brackets (")", "}", "]"), dashes "-", periods ".", and commas ",". Additionally, they were split after specific prefixes and chemical terms like "mono", "di", "tri", "tetra", "penta", "hexa", "hepta", "octa", "nona", "deca", "oxo", "methyl", "hydroxy", "benzene", "oxy", "chloro", "cyclo", "amino", "bromo", "hydro", "fluoro", "methane", "cyano", "amido", "ethene", "phospho", "amide", "butane", "carbono", "sulfane", "sulfino", "iodo", "ethane", "ethyne", "bi", "imino", "nitro", "butan", "idene", "sulfo", "carbon", "propane", "ethen", "acetaldehyde", "benzo", "oxa", "nitroso", "hydra" and "iso". This tokenisation approach aimed to break down the IUPAC name representations into meaningful subunits.

IUPAC Names tokenisation Characterwise: Character-wise tokenisation involved splitting each IUPAC name into tokens after each character, with spaces also treated as characters. For example, "benzene" can be tokenised by splitting into meaningful segments or characters, resulting in different counts of unique tokens. Character-wise splitting would break "benzene" into 7 tokens with 4 unique tokens.

2. Experiment 1 - Impact of Dataset Size and Tokenisation on Model Performance:

The training dataset was obtained from the PubChem database. Of the 110 million compounds downloaded using the chemfp implementation of the MaxMin algorithm, approximately 51 million were selected.

Training and Test Sets:

- From the 51 million subset, a training dataset of 50 million compounds and a testing dataset of 1 million compounds were chosen.
- Additionally, an 11 million compound subset was selected using the MaxMin algorithm, from which a training set of 10 million compounds and a test set of 1 million compounds were created.
- A final subset of 1 million compounds for training and 250,000 for testing was also selected from the 11 million subset.

tokenisation Procedures

SMILES tokenisation followed the methods described in Appendix 1.

IUPAC Names tokenisation: The IUPAC names were split into meaningful tokens using the tokenisation rules from Appendix 1 and were also split into tokens character-wise separately, resulting in three additional training datasets.