VYTAUTAS MAGNUS UNIVERSITY

FACULTY OF NATURAL SCIENCE

DEPARTMENT OF BIOLOGY

Mykhailo Shtopko

Transforming Chemical Languages: Leveraging mT5-based Transformers for Bidirectional Conversion between SMILES and IUPAC Nomenclatures

Bachelor thesis

Biology and Genetics study program, state code 612C10003, Biology field of study

Supervisor: doc. dr. Vykintas Baublys _____
(Degree, name, surname) (signature)

Defended: prof. dr. S. Mickevičius _____
(Dean of the faculty) (signature)

Kaunas, 2024

Work was done: 2023 – 2024 m. at the Vytautas Magnus University, Department of Biology

Reviewer: Dr. Raminta Rodaitė

The work defended: at the meeting of the commission for the defense of bachelor's/master's theses in 2024 at the Vytautas Magnus University, Department of Biology.

Address: University st. 10, Academy 53361

Executor: ……. _____

(Signature)

Scientific supervisor: doc. dr. Vykintas Baublys _____

(Signature)

Department responsible for preparation of bachelor’s thesis: Department of Biology

Department of Biology, head: prof. dr. Jana Radzijevskaja _____

(Signature

TABLE OF CONTENTS

LIST OF ABBREVIATION

ML – Machine Learning;

SMILES – Simplified molecular-input line-entry system;

IUPAC – The International Union of Pure and Applied Chemistry (a set of rules for naming chemical compounds developed by the institution);

ANN – Artificial neural network

NLP – Natural language processing

SUMMARY

Author of term paper: Mykhailo Shtopko
Full title of term paper: Transforming Chemical Languages: Leveraging mT5-based Transformers for Bidirectional Conversion between SMILES and IUPAC Nomenclatures
Term paper supervisor: prof. Vykintas Baublys
Presented at: 2023 – 2024 m. Vytauto Didžiojo Universitete, Biologijos katedroje.
Number of pages: 35
Number of tables: 7
Number of figures: 5

Accurate translation between Simplified Molecular Input System (SMILES) nomenclature and International Union of Pure and Applied Chemistry (IUPAC) nomenclature is critical to building a bridge between computational and human-readable chemical representations. This study presents the application of mT5 transformers to convert between SMILES and IUPAC, aiming to overcome the limitations of rule-based and earlier machine learning approaches. Using a dataset of 120 million molecules from PubChem, we developed and trained mT5-based models to translate chemical names into both nomenclatures. Our models achieved high accuracy and BLEU-4 scores, demonstrating superior performance compared to existing machine learning solutions. Notably, the performance of the models varied depending on the complexity of the chemical structure, with basic models outperforming small models, especially for compounds enriched in certain functional groups or structural features. The study examines the impact of model architecture size and input token length constraints on translation accuracy. The study also compares the developed Python library "Chemical-Converters" with existing tools, showing significant improvements in processing speed and accuracy. This work contributes to chemoinformatics by providing robust tools that enhance the accessibility and usability of chemical data, supporting scientific research and development across multiple sectors.

SANTRAUKA

Baigiamojo darbo autorius: Mykhailo Shtopko
Baigiamojo darbo pilnas pavadinimas: Cheminių kalbų transformacija: mT5 pagrįstų transformerių taikymas dvikrypčiam SMILES ir IUPAC nomenklatūrų konvertavimui
Baigiamojo darbo vadovas: prof. Vykintas Baublys
Prisistatyta: 2023 – 2024 m. Vytauto Didžiojo Universitete, Biologijos katedroje.
Puslapių skaičius: 35
Lentelių skaičius: 7
Paveikslėlių skaičius: 5

Tikslus konvertavimas tarp Paprastųjų molekulių įvedimo sistemos (SMILES) nomenklatūros ir Tarptautinės grynosios ir taikomosios chemijos sąjungos (IUPAC) nomenklatūros yra kritiškai svarbus siekiant sukurti tiltą tarp skaičiavimo ir žmonių skaitymo cheminių atvaizdų. Šis tyrimas pristato mT5 transformatorių taikymą SMILES ir IUPAC konvertavimui, siekiant įveikti ribojimus, kuriuos sukuria taisyklėmis pagrįsti ir ankstesni mašininio mokymosi metodai. Naudodami 120 milijonų molekulių duomenų rinkinį iš PubChem, sukūrėme ir išmokėme mT5 pagrįstus modelius vertimui į abu nomenklatūros. Mūsų modeliai pasiekė aukštą tikslumą ir BLEU-4 rezultatus, demonstruodami pranašesnį veikimą lyginant su esamais mašininio mokymosi sprendimais. Ypač pastebėtas modelių veikimas priklausomai nuo cheminės struktūros sudėtingumo, pagrindiniai modeliai pralenkė mažus modelius, ypač junginiams, turintiems tam tikrų funkcinių grupių arba struktūrinių ypatybių. Tyrimas nagrinėja modelio architektūros dydžio ir įvesties žetonų ilgio apribojimų poveikį vertimo tikslumui. Tyrimas taip pat lygina sukurtą "Chemical-Converters" Python biblioteką su esamais įrankiais, parodant reikšmingus pagerėjimus apdorojimo greičio ir tikslumo atžvilgiu. Šis darbas prisideda prie chemoinformatikos srities, teikdamas patikimus įrankius, kurie padidina cheminių duomenų prieinamumą ir naudojimą, skatindami mokslinius tyrimus ir plėtrą įvairiuose sektoriuose.

INTRODUCTION

Chemoinformatics has transformed the study of chemistry by integrating computational technologies to enhance and accelerate chemical studies and procedures. This has not only improved the precision and effectiveness of chemical research processes, but also expanded its range of uses, influencing several fields such as medicine, material science, and environmental studies (Kumar et al., 2022). The growing complexity of chemical data, which is a result of the emergence of new substances and the expansion of chemical databases, has necessitated the development of more effective computational tools that can effectively manage, analyze, and interpret the data (Irwin et al., 2020).

Chemical nomenclature conversion is a critical process that is essential for the preservation of consistency and clarity in chemical data documentation across global research communities, and it is one of challenges in chemoinformatics (Mahaffy et al., 2008). The International Union of Pure and Applied Chemistry (IUPAC) is the primary organization responsible for the standardization of chemical names, which guarantees that each compound is defined in a unique manner through a systematic naming convention. Nevertheless, the complexity of these nomenclatures presents substantial challenges for computational interpretation, despite their optimization for human use and comprehension. Consequently, the development of sophisticated tools is necessary to bridge the gap between human-readable and machine-readable chemical data formats.

The introduction of notational systems, such as SMILES (Simplified Molecular Input Line Entry System), has been a significant technological advancement. This system facilitates the interaction with digital systems and provides a chemical representation that is machine-readable. SMILES simplifies the manipulation and analysis of computational models by facilitating the efficient encoding and decoding of molecular structures into a line of text (Veselinovic et al., 2015). The inherent complexities and the variability in chemical structure representations make the transition from traditional IUPAC nomenclature to SMILES notation and vice versa a non-trivial task, despite its benefits (Eller, 2006).

This research aims to delve into the development and enhancement of conversion tools that address these challenges, focusing on the accuracy, speed, and usability of these systems in processing and converting chemical names by implementing modified mT5 Transformer ANN. The study seeks to contribute to the broader field of chemoinformatics by providing robust tools that enhance the accessibility and usability of chemical data, ultimately supporting the advancement of scientific research and development across various sectors.

The objectives of the research are the following:

1. To collect a dataset of molecules with SMILES annotations and IUPAC names and to train custom chemical tokenizer on the collected data.
2. To develop and train modified mT5 Transformer on chemical data.
3. To test pre-trained models accuracy and delve deeper into structural patterns and how they affect models accuracy.
4. To develop a Python library Chemical-Converters and compare is to existing similar tool STOUT.

1. LITERATURE ANALYSIS

1.1. Chemical nomenclature

The process of naming chemical compounds is a critical task in science and technology, providing a systematic way to identify and classify different chemical substances (Martin et al., 2017). The International Union of Pure and Applied Chemistry (IUPAC) plays a major role in this process through its development of nomenclature standards, which are comprehensive sets of rules, recommendations, and requirements designed to standardize the naming of chemical compounds, particularly organic compounds (International Union of Pure and Applied Chemistry (IUPAC), 2024; Cahn et al., 2013). This standardized nomenclature facilitates clear communication and spread of information among scientists and researchers by ensuring that each compound's name reflects its chemical structure and properties (Mahaffy et al., 2008).

However, while the IUPAC nomenclature system is designed for human understanding and use (Hepler-Smith et al., 2015), it presents significant challenges when it comes to computer interpretation. The system's complexity and the linguistic nuances involved in chemical names make it difficult for computational tools to parse and interpret them without specialized programs and databases (McEwen, 2020).

To bridge this gap between human-readable chemical nomenclature and machine-readable formats, the simplified molecular-input line-entry system (SMILES) was developed by David Weininger in the late 1980s (Weininger, 1988). SMILES is a notation system that encodes molecular structures into a line of characters, enabling the straightforward representation and manipulation of molecules in digital space (Weininger et al., 1989; Weininger, 1990). Each SMILES string is a compact encoding of a molecule's structure, specifying atoms, bonds, and other features using simple ASCII characters (Putz et al., 2020). This method allows computers to easily process, search, and analyze molecular structures, facilitating tasks such as drug discovery, chemical database management, and various forms of cheminformatics research.

SMILES notation has become a fundamental tool in computational chemistry due to its simplicity and efficiency (Veselinovic et al., 2015). It allows for the representation of complex molecular structures in a manner that is both human-readable and conducive to algorithmic operations, making it an necessary tool for chemists working in the digital domain. Moreover, the widespread adoption of SMILES has led to the development of various software tools and libraries designed to generate, read,

and manipulate SMILES strings, further embedding it as a standard within the chemical informatics field (Korshunova et al., 2021).

1.2. Rules-based approaches for converting between SMILES and IUPAC

In the field of chemoinformatics, researchers commonly employ rules-based approaches such as Chemaxon (Chemaxon, 2024) and ChemDraw (ChemDraw, 2024) to handle the conversion between SMILES representations and IUPAC names. These systems rely on predefined rules and patterns to translate the chemical structure information written in SMILES into the IUPAC nomenclature and vice versa. The primary advantage of using these tools is their accuracy and reliability in handling commonly encountered chemical structures.

However, these methods come with significant drawbacks. One major issue is the high cost associated with acquiring licenses for software, which can be a barrier for some researchers and small institutions. Additionally, the rules-based nature of these systems means they depend heavily on manually defined patterns (Lowe et al., 2011). This rigid structure can lead to inaccuracies or outright errors in cases where chemical structures do not conform to the expected patterns or are novel entities not previously encoded into the system's rule set (Eller, 2006).

1.3. ML-based approaches for converting between SMILES and IUPAC

An alternative to rules-based methodologies for sequence-to-sequence translation is the application of machine learning. The STOUT Model (Rajan et al., 2021) represents an innovative approach in this domain, initially transforming SMILES into Self-Referencing Embedded Strings (SELFIES) (Krenn et al., 2020) before passing the input into the encoder-decoder architecture. This architecture is based on Recurrent Neural Networks (RNNs) equipped with Gated Recurrent Units (GRUs) for both encoder and decoder with a soft attention mechanism. The model was trained with datasets containing 30 million and 60 million examples, yielding mean BLEU-4 scores (Papineni et al., 2002) of 0.86 and 0.92, respectively, for the SMILES to IUPAC task, and 0.89 and 0.94 for the IUPAC to SMILES task. Rajan et al. developed a Python library "STOUT-pypi" which was employed to compare with our models in the tests.

On the other hand, the International Chemical Identifier (InChI) could be utilized instead of SMILES for converting line representation to IUPAC with a transformer model (Handsel et al., 2021).

The model reached an accuracy of 91% on the validation dataset containing molecules with InChI names shorter than 200 characters and IUPAC names shorter than 150 characters.

The transformer-based model, developed by Krasnov et al. showed an accuracy of 98.9% with a beam size 5 for SMILES or structure to IUPAC task and 99.1% for reverse, but only for sequences of length 50 and shorter (Krasnov et al., 2021). The unavailability of the Krasnov`s model was confirmed by our team and noticed by Handsel et al. Based on this fact, Handsel et al. focused on sequences with lengths 50 and longer. It is critical to note a potential oversight in this approach, given that Handsel's tokenizer encompasses merely 66 tokens for InChI and 70 for IUPAC. Although the exact tokenization schema of Krasnov's model remains undisclosed, it is obviously that their approach employs larger than single-character segments of chemical names for tokenization. Therefore, the direct comparison of lengths in terms of tokens may not be entirely accurate, necessitating a recalibration based on the average molecular lengths across the dataset to ensure a proper alignment of gained results.

1.4. Transformer ANN architecture

The original transformer architecture was introduced in the paper "Attention is All You Need" by Vaswani et al. in 2017. It represents a groundbreaking shift in how sequence-to-sequence tasks are solved in the field of natural language processing (NLP) and beyond (Vaswani et al., 2017; Nieminen, 2023). This architecture replaces the recurrent layers commonly employed in sequence modeling with layers of self-attention mechanisms.

Transformers use a self-attention mechanism to calculate the representation of a sequence by counting relationships between different positions within a single input sequence (Hao et al., 2021; Vaswani et al., 2017; Shaw et al., 2018; Ghojogh et al., 2020). The main breakthrough of the Transformer is its ability to manage relationships between input and output at various positions through parallelizable matrix computations, without the need for recurrence (Vaswani et al., 2017, Tay et al., 2022). The architecture includes an encoder and decoder, both consisting of identical layers that utilize multi-head self-attention and fully connected layers for final predictions (Voita et al., 2019).

The original paper (Vaswani et al., 2017) demonstrated that Transformers could achieve state-of-the-art performance on most NLP benchmarks, including the challenging task of machine translation. Following its introduction, the Transformer architecture quickly became the standard in NLP (Myers et al., 2024).

After the introduction of the basic Transformer model, research expanded into various adaptations and branches:

1. BERT (Bidirectional Encoder Representations from Transformers): Was introduced by Devlin et al. in 2018, BERT changed the way contextual information is processed by pre-trained language models by enabling the model to condition on left and context in the all layers (Devlin et al, 2018).
2. GPT (Generative Pre-trained Transformer): By Radford et al., GPT extended the Transformer architecture to a purely generative model formatted to predict the next token in a sequence, demonstrating powerful capabilities in text generation (Radford et al., 2019).
3. Transformer-XL: The model introduced by Dai et al. was specifically designed to address the limitations of learning dependency beyond a fixed length while maintaining temporal coherence (Dai et al., 2019).
4. T5 (Text-to-Text Transfer Transformer): To simplify the process of applying models to a diverse array of NLP tasks, T5 transformed all NLP tasks into a unified text-to-text format (Raffel et al, 2020).

The flexibility and efficiency of the Transformer architecture have led to its adoption across a wide range of tasks, including:

1. Image processing: Vision Transformers (ViTs) apply the Transformer architecture to image classification tasks, demonstrating that self-attention can effectively handle pixel-level dependencies (Khan el al., 2022; Liu et al., 2023; Naseer et al., 2021; Conde et al., 2021).
2. Audio processing: Transformers have been adapted for speech recognition and audio synthesis, illustrating their versatility across different forms of data (Verma et al., 2021; Singla et al., 2022).
3. Multimodal tasks: Models like ViLBERT (Lu et al., 2019) and VideoBERT (Sun et al., 2019) have demonstrated how Transformers can be effectively used for tasks that combine visual and textual data, providing enhanced multimodal understanding (Xu et al., 2023).

Despite their success, Transformers are not without their challenges. They are notably resource-intensive, requiring significant amounts of data and computational power to train effectively (Fournier et al., 2023). This has led to concerns regarding their environmental impact and accessibility to researchers with limited resources (Rillig et al., 2023). Additionally, the black-box nature of these models can complicate efforts to interpret their decisions and outputs (Rudin et al., 2019).

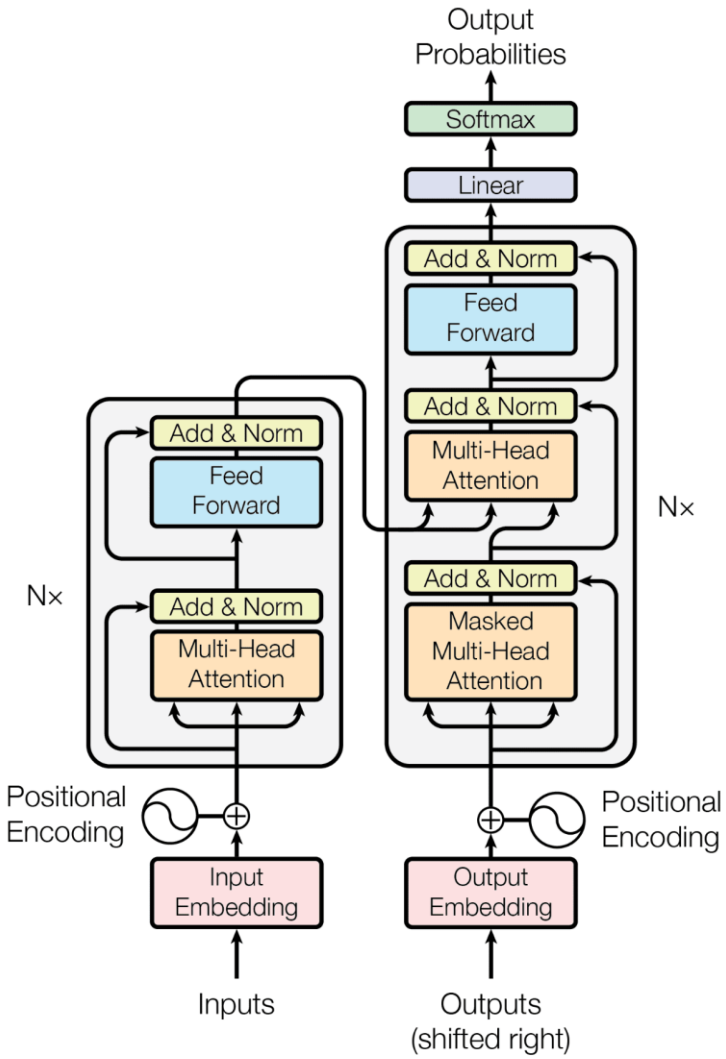

Fig. 1. Scheme of Transformer with attention mechanism (source: Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.)

1.5. mT5 (T5) Transformer Architecture

The mT5 Transformer model (Xue et al., 2020) enhanced the architecture of the T5 (Text-to-Text Transfer Transformer) (Raffel et al., 2020) by incorporating a multilingual capacity, making it adept for handling diverse language inputs. This model is built on the fundamental principles of the Transformer architecture designed by Vaswani et al., designed to process and generate text by leveraging a unified model framework across different languages.

At its core, mT5 retains the encoder-decoder structure characteristic of standard Transformers. This structure is crucial for sequence-to-sequence tasks, where the encoder maps an input sequence of text into a continuous representation that holds all the necessary semantic information of the input, which decoder uses for autoregressive generating the output sequence (Rongali et al., 2019).

Both the encoder and decoder in mT5 utilize the self-attention mechanism, which allows the model to identify the value of different tokens in the input sequence relative to each other (Hao et al., 2021; Vaswani et al., 2017; Shaw et al., 2018; Ghojogh et al., 2020). Unlike traditional attention mechanisms that might focus on a single context vector at a time, self-attention allows the model to analyze different parts of the sequence at the same time, improving the context awareness of the model and parallelizing the matrix computations (Hahn, 2020). This is particularly useful in handling the nuances and contextual variances across multiple languages (Shen et al., 2018).

To maintain a sense of word order, which is not inherently captured by the self-attention mechanism, mT5 uses positional encodings (Kazemnejad et al., 2024). The model is provided with information regarding the relative or absolute position of the tokens in the sequence by these encodings. The positional encoding in mT5 can be either learned or fixed. It is incorporated into the input embeddings at the input of the encoder stack, thereby preserving the order of the input as it passes through the self-attention layers (Kossen et al., 2021).

A critical enhancement within the mT5's architecture is the implementation of multi-head attention (Cordonnier et al., 2020). This property enables the model to parallel analyze input data from various representation subspaces at different locations (Liu et al., 2021). By doing so, mT5 can capture a richer representation of the input, a necessary feature for understanding complex patterns and dependencies in multilingual text.

A feed-forward network (FFN) is present in all layers of both the encoder and the decoder in mT5, which is composed of two linear transformations and a ReLU activation in between. The FFN can

be considered as additional layers of abstraction that are learned from the data, which can capture complex representations beyond what the attention mechanism can do alone (Xue et al., 2020).

mT5 incorporates layer normalization and residual connections, which are standard in preventing the vanishing gradient problem in deep networks. Layer normalization should be applied before every sub-layer (self-attention and FFN) and before the output layer of the decoder. This helps stabilize the learning process by normalizing the activations of the previous layer. Residual connections, or skip connections, allow the gradients to flow directly through the network, enabling deeper models by mitigating the risk of gradient dispersion during training (Xue et al., 2020; Neill et al., 2020).

2. MATERIALS AND METHODS

2.1. Data

The PubChem FTP service (PubChem-FTP, 2024) was used to collect a dataset of 120 million molecules in SDF format. A custom SDF parser was designed to extract relevant fields from each record, as detailed in Table 1. The initial dataset was filtered to exclude molecules whose IUPAC names included isotopic notations, with the regular expression pattern "\(\d+[A-Z]+\)|deuter|triti|". In preparation for training the canonical models, modifications were made to the IUPAC names to eliminate notations about isomers, using the regex pattern "\((?:\d*[RSEZ](?:,\d*[RSEZ])*)\)-?." Each molecule's record was then augmented by replicating it threefold and prefixing the SMILES sequence with one of three IUPAC style tokens: "<BASE>", "'<TRAD>", or "<SYST>". This operation tripled the dataset's size. The "<TRAD>" token is associated with the "PUB- CHEM_IUPAC_TRADITIONAL_NAME" field, often incorporating trivial names, whereas the "<SYST>" token corresponds to the "PUBCHEM_IUPAC_SYSTEMATIC_NAME" field, which doesn't contain trivial names. The "<BASE>" token, linked to the "PUBCHEM_IUPAC_NAME" field, could include trivial names, serving as an intermedi- ate between the systematic and traditional styles.

The average token length of SMILES sequences was determined to be 52.2, with a standard deviation of 23.82, for those in canonical format. For isomeric SMILES sequences, the average token length was slightly higher, at 54.75, with a standard deviation of 28.65. In contrast, the mean token length for IUPAC names was found to be 48.05, with a standard deviation of 26.58, when excluding information on isomerism. For IUPAC names that include isomeric details, the average length in tokens increased marginally to 50.45, with a standard deviation of 31.30, as depicted in Figure 2.

Table 1: SDF data parsed fields

Field	Style Token	Description
PUBCHEM_IUPAC_NAME	<BASE>	A compromise between the <SYST> and <TRAD>styles
PUBCHEM_IUPAC_SYSTEMATIC_NAME	<SYST>	Doesn't contain trivial names
PUBCHEM_IUPAC_TRADITIONAL_NAME	<TRAD>	Contains trivial names
PUBCHEM_OPENEYE_ISO_SMILES		SMILES strings containing information about isomery
PUBCHEM_OPENEYE_CANONICAL_SMILES		SMILES strings containing only structural in-formation without isomery

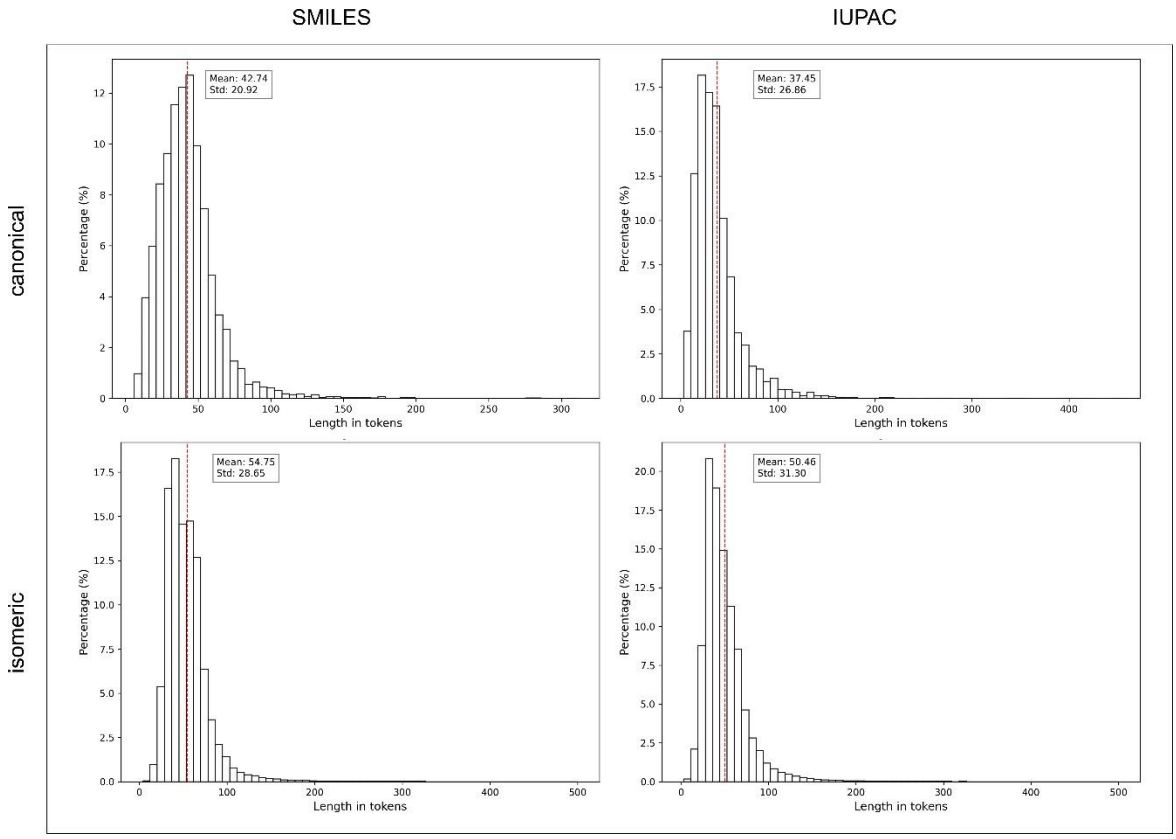

Figure 2: Histograms of SMILES and IUPAC names length in tokens distribution within the dataset. Plots represent only sequences with a length of less than 500, excluding rare long sequences.

2.2. Tokenizers

Custom tokenizers were developed based on the PreTrainedTokenizerFast (PreTrainedTokenizerFast, 2024) class from the Transformers Python library (Transformers, 2024), employing a WordLevel tokenizer (Tokenizers, 2024) combined with a BPEDecoder (Decoders, 2024) for decoding and split methods for the pre-tokenization process. The regular expression patterns designated for the split methods are accessible within the official repository hosted on Hugging Face, available at SMILES-FAST-TOKENIZER (https://huggingface.co/knowledgator/SMILES-FAST-TOKENIZER) and IUPAC-FAST-TOKENIZER (https://huggingface.co/knowledgator/IUPAC-FAST-TOKENIZER). In the tokenization of SMILES sequences, a character-by-character and element-by-element approach was adopted, whereas, for IUPAC names, a more granulated strategy was implemented, wherein numbers, characters, and segments of names were tokenized distinctively as represented in Table 2.

2.3. Models architecture

All models employed in the study were built on the MT5 transformer architecture designed by Google (Xue et al., 2020), which is built on T5 transformer architecture (Raffel et al., 2020). The model was modified to accept different tokenizers for the encoder and decoder, especially SMILES and IUPAC tokenizers. Detailed parameters are presented in Table 3.

Table 2: Examples of tokenization

Type	Details
SMILES Canonical Input	<BASE>CC(=O)OC(CC(=O)[O-])C[N+](C)(C)C
SMILES Canonical Output	['<BASE>', 'C', 'C', '(', '=', 'O', ')', 'O', 'C', '(', 'C', 'C', '(', '=', 'O', ')', '[', 'O', '-', ']', ')', 'C', '[', 'N', '+', ']', '(', 'C', ')', '(', 'C', ')', 'C']
IUPAC Canonical Input	3-acetyloxy-4-(trimethylazaniumyl)butanoate
IUPAC Canonical Output	['3', '-', 'acet', 'yl', 'ox', 'y', '-', '4', '-', '(', 'tri', 'meth', 'yl', 'az', 'anium', 'yl', ')', 'but', 'an', 'oate']
SMILES Isomeric Input	<BASE>C1=CC(=CC=C1C[C@H](C(=O)O)NC(=O)CN)O
SMILES Isomeric Output	['<BASE>', 'C', '1', '=', 'C', 'C', '(', '=', 'C', 'C', '=', 'C', '1', 'C', '[', 'C', '@', 'H', ']', '(', 'C', '(', '=', 'O', ')', 'O', ')', 'N', 'C', '(', '=', 'O', ')', 'C', 'N', ')', 'O']
IUPAC Isomeric Input	(2R)-2-[(2-aminoacetyl)amino]-3-(4-hydroxyphenyl)propanoic acid
IUPAC Isomeric Output	['(', '2', 'R', ')', '-', '2', '-', '[', '(', '2', '-', 'amino', 'acet', 'yl', ')', 'amino', ']', '-', '3', '-', '(', '4', '-', 'hydr', 'ox', 'y', 'phen', 'yl', ')', 'prop', 'an', 'oic', ' ', 'acid']

Table 3: Parameters of the models

Model	d_ff	d_kv	d_model	num_layers	num_decoder_layers	num_heads
Small models	512	64	256	4	4	3
Base models	1024	64	512	8	8	6

2.4. Models training

All models were trained on a corpus of 100 M examples of molecules from a shuffled dataset, utilizing the Dataset class from the Transformers library, with an initialization seed set to 42. The training procedures were conducted on Azure Standard NC24ads A100 v4 instance, equipped with 24 virtual CPUs and 220 GiB of RAM, for two epochs. Throughout the training phase, metrics were collected with the WandB library. For the training of the small models, a learning rate of 3e-4 and a batch size of 1024 were employed, the base models were trained with a learning rate of 1e-4 and a reduced batch size of 512.

2.5. Evaluation

The performance of trained models was evaluated using a dataset comprising 100K test examples randomly selected from the 12M test dataset. The primary metric for evaluating model efficacy was the BLEU-4 score (4 n-grams), which measures the correspondence between the model's predictions and the target sequences on a scale of 0 to 1. For this purpose, both the predicted and target sequences were first tokenized with the described above tokenizers. Subsequently, the BLEU-4 scores were calculated using the sentence_bleu method provided by the NLTK library (NLTK, 2024) and applied to the lists of tokens.

The absolute accuracy of the models was determined by calculating the ratio of correctly predicted examples to the total number of examples. Additionally, the Pearson correlation method (Cohen et al., 2009) was employed to examine the relationship between the BLEU-4 score and the input length in tokens. This analysis was complemented by the generation of plots to visually represent this correlation.

Metrics were also specifically calculated for inputs with lengths less than the maximum input length —128 tokens for SMILES inputs and 156 tokens for IUPAC inputs. Furthermore, we computed the accuracy and BLEU-4 scores for inputs with an average length of 52 tokens for SMILES inputs and 48 tokens for IUPAC inputs.

To delve into the complex chemical patterns represented in our models, we analyzed incorrectly predicted molecules by clustering their Morgan fingerprints (Pattanaik et al., 2020), with a radius of 1 and a bit size of 2048, into 50 clusters using the k-means method (Ahmed et al., 2020) from the Sklearn library (Sklearn, 2024). We have trained a single k-means model for canonical subsets. Subsequently, 50 most frequent SMILES and IUPAC tokens were plotted across the clusters for better understanding the structural patterns inherent for particular clusters. This approach could potentially help to understand the models' performance concerning intricate chemical structures.

3. RESULTS AND DISCUSSION

3.1. Tokenizers training

Tokenizers were successfully trained on the entire dataset. The total vocabulary size of the SMILES tokenizer reached 138 and 822 for the IUPAC tokenizer. Tokenizers were able to tokenize every sequence in the dataset, therefore, were determined as suitable to be employed for the models.

3.2. Models benchmarking

The pre-trained models examined in the study were evaluated using a test dataset and demonstrated remarkable accuracy and BLEU-4 scores as presented in Tables 4 and 5. Among these, the "SMILES2IUPAC-canonical-base" and "IUPAC2SMILES-canonical-base" models stood out due to their exceptional performance. These models achieved mean BLEU-4 scores of 0.9690 and 0.9800, respectively, indicating a near-perfect translation between SMILES notation system and IUPAC nomenclature.

Further analysis of performance metrics shows that both models maintain robustness across various input sizes. Notably, the "IUPAC2SMILES-canonical-base" model recorded a BLEU-4 score as high as 0.9977 for average-sized inputs, as depicted in Figure 3. A notable aspect of these results is the negative Pearson correlation between BLEU-4 scores and input size, indicating that as the input size increases, the BLEU-4 score tends to decrease, with the strongest negative correlation observed in the "IUPAC2SMILES-canonical-base" model (-0.8291). Also, as the architecture increases, the model tends to become less accurate as the size of the input sequence increases, while nevertheless remaining significantly more accurate than smaller models.

Table 4: Models‘ accuracy

Model	**Mean accuracy**	**Accuracy for input<=max input size**	**Accuracy for input = average input size**
SMILES2IUPAC-canonical-small	0.7674	0.7927	0.8008
SMILES2IUPAC-canonical-base	0.8842	0.9117	0.9158
IUPAC2SMILES-canonical-small	0.9054	0.9262	0.9612
IUPAC2SMILES-canonical-base	0.9488	0.9705	0.9886

Table 5: Models‘ BLEU-4 score

Model	Mean BLEU-4 score	BLEU-4 score for input <= max input size	BLEU-4 score for input = average input size	Pearson correlation of the BLEU-4 score with the input size
SMILES2IUPAC-canonical-small	0.9395	0.9556	0.9608	-0.5824
SMILES2IUPAC-canonical-base	0.9690	0.9852	0.9871	-0.7324
IUPAC2SMILES-canonical-small	0.9723	0.9872	0.9899	-0.7848
IUPAC2SMILES-canonical-base	0.9800	0.9949	0.9977	-0.8291

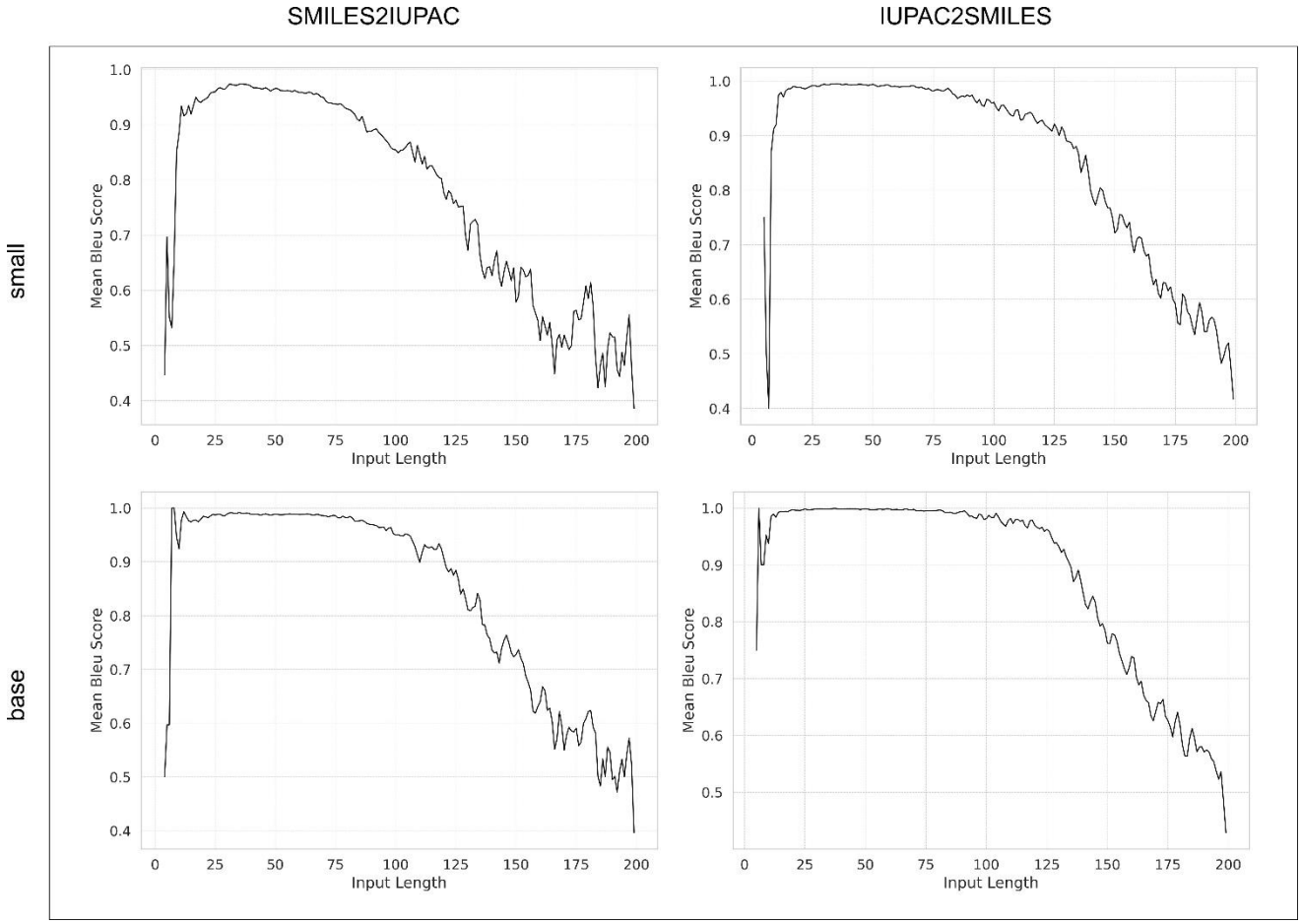

Figure 3: Mean BLEU-4 score per input length in tokens. Plots represent only sequences with a length of less than 200, excluding rare long sequences.

3.3. Clustering analysis

The comprehensive evaluation across 50 clusters as shown in Table 6 reveals insights into the performance of the SMILES2IUPAC and IUPAC2SMILES models, underscoring the influence of model architecture size and the complexity inherent in chemical structure representation. For better understanding the structural patterns, we have plotted 50 most frequent SMILES (Figure 4) and IUPAC (Figure 5) clusters. The clusters, delineated based on structural characteristics and the presence of specific chemical groups, span a broad spectrum of BLEU-4 scores, indicating the variable difficulty of accurately converting between SMILES and IUPAC nomenclature.

A distinct pattern emerges from the analysis: clusters enriched with specific chemical features, such as double bonds and nitrogen (Cluster 2), or those containing sulfur (Clusters 8 and 24) or halogens (Clusters 11, 39) exhibit higher BLEU-4 scores, suggesting that these features are better captured and translated by the models. This is particularly evident in the performance of the SMILES2IUPAC model, where clusters with enriched nitrogen and oxygen content demonstrate exceptionally high accuracy, with BLEU-4 scores reaching up to 0.99 and 1.0 in both small and base model sizes.

Clusters that feature intricate structural characteristics, particularly those with branched configurations and a reduction in double bonds coupled with an increased presence of oxygen and amino groups (such as Clusters 12 and 21), pose significant challenges to computational models. This complexity in molecular structure often results in these models achieving lower BLEU-4 scores, indicating a struggle to accurately predict or interpret these complex entities compared to simpler molecular structures.

The model architecture size plays a significant role in performance, with the base model consistently achieving higher scores across the majority of clusters. This suggests that the increased capacity of the base model enables more effective learning and representation of complex chemical structures. However, the performance gap between the small and base models narrows in clusters where the chemical structures are less complex, indicating that for simpler conversions, the additional capacity of the base model does not significantly enhance performance.

The imposed limitations on input token lengths, with a maximum of 128 tokens for SMILES2IUPAC and 156 for IUPAC2SMILES, constrain the models' ability to accurately represent and convert longer chemical structures. Clusters exhibiting lower performance, such as Cluster 29 with a mean length of 192.04 SMILES tokens, underscore the challenges posed by these limitations, particularly for the conversion of highly complex molecules.

Table 6: BLEU-4 Score Across Clusters for SMILES2IUPAC and IUPAC2SMILES Models

Cluster	Features	Count	Mean Length	SMILES2IUPAC		IUPAC2SMILES	
			(SMILES Tokens)	small	base	small	base
0	-	1732	51.85	0.96	0.98	0.99	0.99
1	-	2454	46.35	0.91	0.97	0.97	0.99
2	Enriched with double-bonds and nitrogen	2622	52.25	0.98	0.99	1	1
3	Branched structure	1742	75.93	0.93	0.95	0.95	0.96
4	Enriched with nitrogen and oxygen-depleted	1705	32.94	0.98	0.99	0.99	1
5	Enriched with double-bonds and carbon	1485	51.55	0.91	0.97	0.97	0.98
6	Branched and structure and double-bonds-depleted	1651	37.03	0.95	0.97	0.98	0.99
7	-	2088	67.85	0.92	0.95	0.96	0.98
8	Sulfur-enriched	2488	52.54	0.97	0.99	0.99	0.99
9	Branched structure and double-bonds-depleted	879	33.02	0.95	0.98	0.98	0.99
10	Nitrogen-depleted	1920	60.15	0.96	0.97	0.98	0.98
11	Contains fluorine	2418	64.26	0.96	0.97	0.98	0.98
12	Branched structure, contains amino-groups	1050	102.26	0.79	0.84	0.9	0.9
13	Enriched with nitrogen and oxygen-depleted	2597	44.46	0.96	0.98	0.98	0.99
14	-	3407	52.3	0.97	0.98	0.99	0.99
15	Double-bonds-depleted	2034	28.58	0.92	0.96	0.97	0.99
16	Branched structure and double-bonds-depleted	1690	40.27	0.96	0.97	0.98	0.99
17	-	1640	40.68	0.95	0.98	0.99	0.99
18	Enriched with oxygen	1342	55.73	0.95	0.98	0.98	0.99
19	Enriched with nitrogen and oxygen-depleted	1898	48.23	0.96	0.99	0.98	0.99

Continued on next page

Table 6: Continued from previous page

Cluster	Features	Count	Mean Length (SMILES Tokens)	SMILES2IUPAC		IUPAC2SMILES	
				small	base	small	base
20	Enriched with double-bonds	1548	61.43	0.96	0.98	0.98	0.98
21	Branched structure oxygen-enriched and double-bonds-depleted	1128	101.45	0.84	0.86	0.89	0.9
22	Nitrogen- and oxygen-depleted	2231	49.51	0.86	0.96	0.94	0.97
23	-	3255	55.11	0.98	0.99	0.99	0.99
24	Sulfur-enriched	777	66.2	0.98	0.99	0.99	0.99
25	Enriched with nitrogen and oxygen-depleted	2224	52.26	0.93	0.97	0.96	0.97
26	-	2351	46.24	0.96	0.99	0.98	0.99
27	Enriched with carbon and oxygen	1084	89.03	0.78	0.96	0.96	0.98
28	Enriched with fluorine	2747	69.61	0.96	0.97	0.97	0.98
29	-	1609	192.04	0.66	0.77	0.72	0.75
30	-	1958	54.89	0.96	0.98	0.98	0.99
31	Nitrogen- and oxygen-depleted	2248	42.78	0.94	0.97	0.98	0.99
32	Contains triple-bonds	2350	57.18	0.95	0.97	0.97	0.98
33	Contains fluorine	2006	43.06	0.97	0.99	0.99	0.99
34	-	2597	47.85	0.96	0.98	0.99	1
35	Branched structure oxygen-enriched and double-bonds-depleted	2145	37.86	0.9	0.97	0.97	0.98
36	-	1971	58.78	0.97	0.98	0.99	0.99
37	Contains ions	3256	61.61	0.97	0.98	0.98	0.99
38	-	2526	53.73	0.97	0.98	0.99	0.99
39	Contains chlorine	2605	65.01	0.96	0.97	0.98	0.98
40	-	2220	73.27	0.92	0.93	0.96	0.96
41	Enriched with chlorine and oxygen-depleted	2027	44.09	0.97	0.99	0.99	0.99
42	Contains chlorine	2769	52.95	0.98	0.99	0.99	1

Continued on next page

Table 6: Continued from previous page

Cluster	Features	Count	Mean Length (SMILES Tokens)	SMILES2IUPAC		IUPAC2SMILES	
				small	base	small	base
43	Contains nitrogen	2167	49.03	0.98	0.99	0.99	0.99
44	Enriched with nitrogen and double-bonds-depleted	1326	75.34	0.78	0.87	0.9	0.95
45	Enriched with nitrogen and oxygen-depleted	1146	45.14	0.99	1	1	1
46	-	2067	64.96	0.96	0.97	0.98	0.98
47	-	2636	43.35	0.95	0.99	0.98	0.99
48	Contains ions, fluorine, double-bonds-depleted	1746	30.76	0.84	0.96	0.94	0.96
49	-	406	73.8	0.99	1	1	1

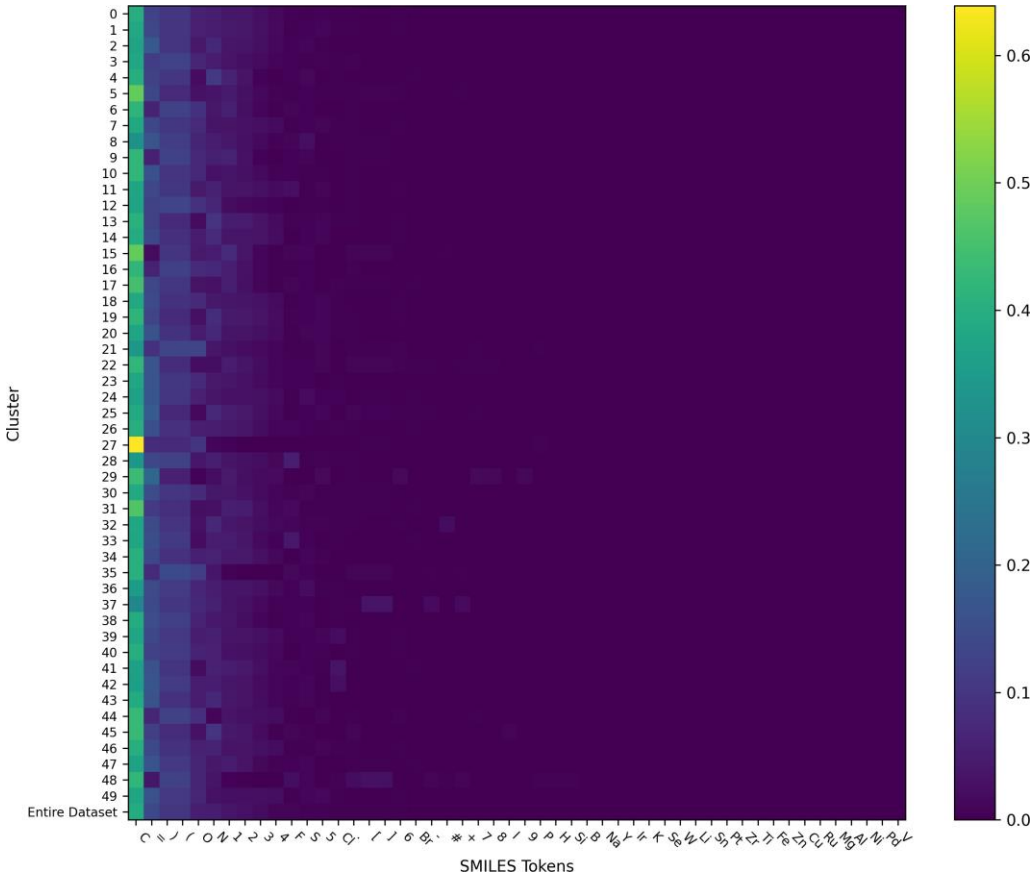

Figure 4: Canonical SMILES 50 most frequent SMILES tokens for 50 clusters in the testing dataset 100k subset.

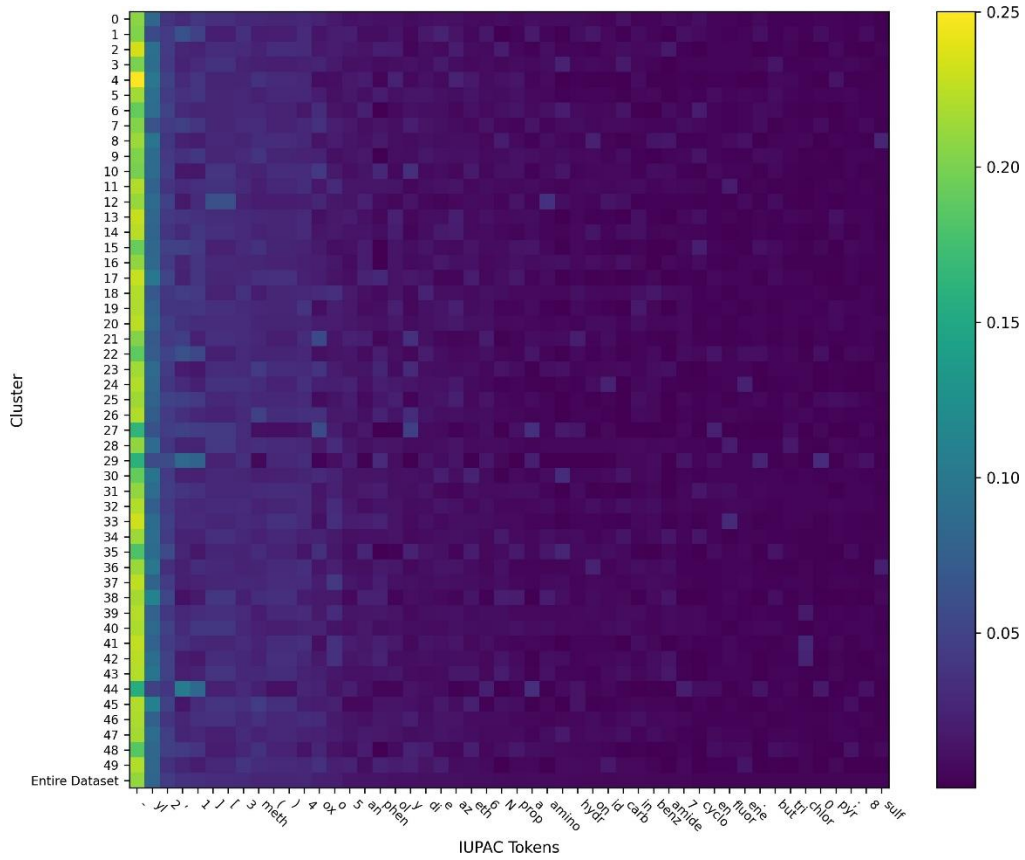

Figure 5: Canonical SMILES 50 most frequent IUPAC tokens for 50 clusters in the testing dataset 100k subset.

3.4. Chemical Converters and STOUT comparison

We have successfully developed and analyzed the Chemical-Converters (Chemical-Converters, 2024), a Python library designed for efficient chemical naming conversion, based on our pre-trained models, benchmarking it against the STOUT model proposed by Rajan et al., 2021. The Chemical-Converters significantly outperformed STOUT in terms of processing speed, handling 66,000 molecules per hour compared to STOUT's 462. This was further enhanced by our implementation of batch processing, a feature absent in STOUT, which significantly contributes to the computing efficiency of our models. Accuracy assessments revealed that our models achieve a mean accuracy of approximately 0.8842 for smaller configurations and over 0.94 for medium-sized models, surpassing STOUT's accuracy of 0.78688.

Additionally, the model weights of "Chemical Converters" range from 20 MB for smaller versions to 180 MB for medium versions, which are considerably lighter compared to STOUT's 216 MB, potentially offering easier integration and less computational overhead. The models are built on Pytorch and are widely accessible through HuggingFace and GitHub, unlike STOUT, which relies on a custom cloud bucket for distribution. Notably, our models have also seen substantial community engagement,

with over 1.2 million downloads in the last 3 months, dwarfing the 1,500 downloads recorded for STOUT.

This comparative analysis underlines the enhanced performance and accessibility of the "Chemical Converters," highlighting its potential as a superior tool in the field of chemoinformatics for researchers and practitioners seeking efficient and accurate chemical data processing solutions.

Table 7: Comparison of our models with STOUT (Rajan et al., 2021)

Fearure	**Chemical-Converters**	**STOUT**
Speed (number of processed molecules per hour)	66000	462
Implemented batch-processing	+	-
Mean accuracy	0.8842 for small models and 0.94+ for medium	0.78688
Model weight	from 20 for small models and to 180 MB for medium	216 MB
Core framework	Pytorch	TensorFlow
Model availability	HuggingFace, GitHub	Custom bucket
Model downloads in last 3 month	1.2 M +	1 k+

4. CONCLUSIONS

1. The project successfully gathered a vast dataset of molecules annotated with SMILES and IUPAC names, upon which a custom tokenizer was developed and trained. The tokenizers achieved comprehensive coverage with a total vocabulary size reaching 138 for SMILES and 822 for IUPAC, demonstrating their robustness and suitability for handling diverse chemical data

2. The modified mT5 Transformer was adeptly developed and trained on the chemical data, which led to the creation of models that demonstrated high proficiency in translating between SMILES and IUPAC notations. The models' remarkable performance, with BLEU-4 scores nearing perfection (up to 0.9800 for IUPAC2SMILES and 0.9690 for SMILES2IUPAC), highlights the effectiveness of the Transformer architecture in capturing the complex relationships within chemical structures and names.

3. The detailed evaluation of the models provided insights into how structural patterns within molecules influence translation accuracy. Models performed exceptionally well with molecules characterized by simpler chemical features, while complex molecules with intricate structures like branched configurations or those enriched with multiple chemical groups posed challenges, as seen in the lower BLEU-4 scores for such clusters.

4. Developed as a part of the project Python library, Chemical-Converters, significantly outperformed the existing STOUT tool in several key areas, including processing speed, where it handled 66,000 molecules per hour compared to STOUT’s 462. This stark improvement in efficiency, combined with the library’s higher accuracy and accessibility, highlights its potential as a superior tool in chemical data processing.

5. ACKNOWLEDGMENTS

I would like to sincerely thank Knowledgator Engineering LTD for supplying the necessary resources for training the neural networks utilized in this study. I express my gratitude to Ihor Stepanov for his invaluable contributions in designing the architecture of the models, which greatly enhanced our research. Their assistance was pivotal in the accomplishment of this project, and I sincerely value their contributions to the progression of scientific research in the realm of bioinformatics.

6. REFERENCES

1. Kumar, R., Chaudhary, M. P., & Chauhan, N. (2022). Recent advances and current strategies of cheminformatics with artificial intelligence for development of molecular chemistry simulations. Journal of Molecular Chemistry, 2(2), 440-440.
2. Irwin, J. J., Tang, K. G., Young, J., Dandarchuluun, C., Wong, B. R., Khurelbaatar, M., ... & Sayle, R. A. (2020). ZINC20—a free ultralarge-scale chemical database for ligand discovery. Journal of chemical information and modeling, 60(12), 6065-6073.
3. Mahaffy, P., Ashmore, A., Bucat, B., Do, C., & Rosborough, M. (2008). Chemists and" the public": IUPAC's role in achieving mutual understanding (IUPAC Technical Report). *Pure and Applied Chemistry*, *80*(1), 161-174.
4. M Veselinovic, A., B Veselinovic, J., V Zivkovic, J., & M Nikolic, G. (2015). Application of SMILES notation based optimal descriptors in drug discovery and design. *Current topics in medicinal chemistry*, *15*(18), 1768-1779.
5. Eller, G. A. (2006). Improving the quality of published chemical names with nomenclature software. *Molecules*, *11*(11), 915-928.
6. Martin, K., Obdulia, R., Anália, L., Julen, O., & Alfonso, V. (2017). Information Retrieval and Text Mining Technologies for Chemistry.
7. International Union of Pure and Applied Chemistry (IUPAC). https://iupac.org/, Mar 2024. Accessed: March 2024.
8. Cahn, R. S., & Dermer, O. C. (2013). *Introduction to chemical nomenclature*. Butterworth-Heinemann.
9. Hepler-Smith, E. (2015). Systematic Flexibility and the History of the IUPAC Nomenclature of Organic Chemistry. *Chemistry International*, *37*(2), 10-14.
10. McEwen, L. R. (2020). Towards a Digital IUPAC. *Chemistry International*, *42*(2), 15-17.
11. Weininger, D. (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.
12. Weininger, D., Weininger, A., & Weininger, J. L. (1989). SMILES. 2. Algorithm for generation of unique SMILES notation. *Journal of chemical information and computer sciences*, *29*(2), 97-101.

13. Weininger, D. (1990). SMILES. 3. DEPICT. Graphical depiction of chemical structures. *Journal of chemical information and computer sciences*, *30*(3), 237-243.
14. Putz, M. V., & Dudaş, N. A. (2020). Smiles. In *New Frontiers in Nanochemistry: Concepts, Theories, and Trends* (pp. 387-402). Apple Academic Press.
15. Korshunova, M., Ginsburg, B., Tropsha, A., & Isayev, O. (2021). OpenChem: a deep learning toolkit for computational chemistry and drug design. *Journal of Chemical Information and Modeling*, *61*(1), 7-13.
16. Chemaxon - a cheminformatics and bioinformatics software. https://chemaxon.com/, Mar 2024. Accessed: March 2024.
17. ChemDraw - a molecule editor. https://revvitysignals.com/, Mar 2024. Accessed: March 2024.
18. Lowe, D. M., Corbett, P. T., Murray-Rust, P., & Glen, R. C. (2011). Chemical name to structure: OPSIN, an open source solution.
19. Rajan, K., Zielesny, A., & Steinbeck, C. (2021). STOUT: SMILES to IUPAC names using neural machine translation. *Journal of Cheminformatics*, *13*(1), 34.
20. Krenn, M., Häse, F., Nigam, A., Friederich, P., & Aspuru-Guzik, A. (2020). Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, *1*(4), 045024.
21. Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002, July). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics* (pp. 311-318).
22. Handsel, J., Matthews, B., Knight, N. J., & Coles, S. J. (2021). Translating the InChI: adapting neural machine translation to predict IUPAC names from a chemical identifier. *Journal of cheminformatics*, *13*, 1-11.
23. Krasnov, L., Khokhlov, I., Fedorov, M. V., & Sosnin, S. (2021). Transformer-based artificial neural networks for the conversion between chemical notations. *Scientific Reports*, *11*(1), 14798.
24. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, *21*(140), 1-67.
25. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

26. Nieminen, M. (2023). The Transformer Model and Its Impact on the Field of Natural Language Processing.
27. Hao, Y., Dong, L., Wei, F., & Xu, K. (2021, May). Self-attention attribution: Interpreting information interactions inside transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 35, No. 14, pp. 12963-12971).
28. Shaw, P., Uszkoreit, J., & Vaswani, A. (2018). Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*.
29. Ghojogh, B., & Ghodsi, A. (2020). Attention mechanism, transformers, BERT, and GPT: tutorial and survey.
30. Tay, Y., Dehghani, M., Bahri, D., & Metzler, D. (2022). Efficient transformers: A survey. *ACM Computing Surveys*, *55*(6), 1-28.
31. Voita, E., Talbot, D., Moiseev, F., Sennrich, R., & Titov, I. (2019). Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*.
32. Myers, D., Mohawesh, R., Chellaboina, V. I., Sathvik, A. L., Venkatesh, P., Ho, Y. H., ... & Jararweh, Y. (2024). Foundation and large language models: fundamentals, challenges, opportunities, and social impacts. *Cluster Computing*, *27*(1), 1-26.
33. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
34. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), 9.
35. Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q. V., & Salakhutdinov, R. (2019). Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.
36. Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., ... & Raffel, C. (2020). mT5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.
37. Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., & Shah, M. (2022). Transformers in vision: A survey. *ACM computing surveys (CSUR)*, *54*(10s), 1-41.
38. Liu, Y., Zhang, Y., Wang, Y., Hou, F., Yuan, J., Tian, J., ... & He, Z. (2023). A survey of visual transformers. *IEEE Transactions on Neural Networks and Learning Systems*.

39. Naseer, M. M., Ranasinghe, K., Khan, S. H., Hayat, M., Shahbaz Khan, F., & Yang, M. H. (2021). Intriguing properties of vision transformers. *Advances in Neural Information Processing Systems*, *34*, 23296-23308.
40. Conde, M. V., & Turgutlu, K. (2021). CLIP-Art: Contrastive pre-training for fine-grained art classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3956-3960).
41. Verma, P., & Berger, J. (2021). Audio transformers: Transformer architectures for large scale audio understanding. adieu convolutions. *arXiv preprint arXiv:2105.00335*.
42. Singla, Y. K., Shah, J., Chen, C., & Shah, R. R. (2022, November). What do audio transformers hear? probing their representations for language delivery & structure. In *2022 IEEE International Conference on Data Mining Workshops (ICDMW)* (pp. 910-925). IEEE.
43. Lu, J., Batra, D., Parikh, D., & Lee, S. (2019). Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, *32*.
44. Sun, C., Myers, A., Vondrick, C., Murphy, K., & Schmid, C. (2019). Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 7464-7473).
45. Xu, P., Zhu, X., & Clifton, D. A. (2023). Multimodal learning with transformers: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
46. Fournier, Q., Caron, G. M., & Aloise, D. (2023). A practical survey on faster and lighter transformers. *ACM Computing Surveys*, *55*(14s), 1-40.
47. Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, *1*(5), 206-215.
48. Rillig, M. C., Ågerstrand, M., Bi, M., Gould, K. A., & Sauerland, U. (2023). Risks and benefits of large language models for the environment. *Environmental Science & Technology*, *57*(9), 3464-3466.
49. Rongali, S., Soldaini, L., Monti, E., & Hamza, W. (2020, April). Don't parse, generate! a sequence to sequence architecture for task-oriented semantic parsing. In *Proceedings of the web conference 2020* (pp. 2962-2968).
50. Hahn, M. (2020). Theoretical limitations of self-attention in neural sequence models. *Transactions of the Association for Computational Linguistics*, *8*, 156-171.

51. Shen, T., Zhou, T., Long, G., Jiang, J., & Zhang, C. (2018). Tensorized self-attention: Efficiently modeling pairwise and global dependencies together. *arXiv preprint arXiv:1805.00912*.
52. Kazemnejad, A., Padhi, I., Natesan Ramamurthy, K., Das, P., & Reddy, S. (2024). The impact of positional encoding on length generalization in transformers. *Advances in Neural Information Processing Systems*, *36*.
53. Kossen, J., Band, N., Lyle, C., Gomez, A. N., Rainforth, T., & Gal, Y. (2021). Self-attention between datapoints: Going beyond individual input-output pairs in deep learning. *Advances in Neural Information Processing Systems*, *34*, 28742-28756.
54. Cordonnier, J. B., Loukas, A., & Jaggi, M. (2020). Multi-head attention: Collaborate instead of concatenate. *arXiv preprint arXiv:2006.16362*.
55. Liu, L., Liu, J., & Han, J. (2021). Multi-head or single-head? an empirical comparison for transformer training. *arXiv preprint arXiv:2106.09650*.
56. Neill, J. O. (2020). An overview of neural network compression. *arXiv preprint arXiv:2006.03669*.
57. PubChem-FTP - An open repository of chemical data.https://ftp.ncbi.nlm.nih.gov/pubchem/RDF/, Mar 2024. Accessed: March 2024.
58. PreTrainedTokenizerFast - Transformers tokenizer class. https://huggingface.co/docs/transformers/en/main_classes/tokenizer, Mar 2024. Accessed: March 2024.
59. Transformers - An open-source library for building transformer models. https://huggingface.co/docs/transformers/en/index, Mar 2024. Accessed: March 2024.
60. Tokenizers - Classes from the Python library Transformers. https://huggingface.co/docs/transformers/main/en/main_classes/tokenizer, Mar 2024. Accessed: March 2024.
61. Decoders - Classes from the Python library Transformers. https://huggingface.co/docs/tokenizers/en/api/decoders, Mar 2024. Accessed: March 2024.
62. NLTK - Natural Language Toolkit for Python. https://www.nltk.org/, Mar 2024. Accessed: March 2024.
63. Cohen, I., Huang, Y., Chen, J., Benesty, J., Benesty, J., Chen, J., ... & Cohen, I. (2009). Pearson correlation coefficient. *Noise reduction in speech processing*, 1-4.

64. Pattanaik, L., & Coley, C. W. (2020). Molecular representation: going long on fingerprints. *Chem*, *6*(6), 1204-1207.
65. Ahmed, M., Seraj, R., & Islam, S. M. S. (2020). The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics*, *9*(8), 1295.
66. Sklearn - A free and open-source machine learning library for the Python programming language. https://scikit-learn.org/, Mar 2024. Accessed: March 2024.
67. Chemical-Converters – AI-based library for translating between SMILES and IUPAC. https://github.com/Knowledgator/chemical-converters, Mar 2024. Accessed: March 2024.