Struct2IUPAC – Transformer-based artificial neural network for the conversion between chemical notations

Lev Krasnov,[†,‡,¶] Ivan Khokhlov,[¶] Maxim V. Fedorov,[†,¶] and Sergey Sosnin[*,†,¶]

†*Center for Computational and Data-Intensive Science and Engineering, Skolkovo Institute of Science and Technology, Skolkovo Innovation Center, Moscow 143026, Russia*

‡*Department of Chemistry, Lomonosov Moscow State University, 119991 Moscow GSP-1, 1-3 Leninskiye Gory, Russia*

¶*Syntelly LLC, Skolkovo Innovation Center, Moscow 143026, Russia*

E-mail: sergey.sosnin@skoltech.ru

Abstract

Providing IUPAC chemical names is necessary for chemical information exchange. We developed a Transformer-based artificial neural architecture to translate between SMILES and IUPAC chemical notations: *Struct2IUPAC* and *IUPAC2Struct*. Our models demonstrated the performance that is comparable to rule-based solutions. We proved that both accuracy, speed of computations, and the model's robustness allow us to use it in production. Our showcase demonstrates that a neural-based solution can encourage rapid development keeping the same performance. We believe that our findings will inspire other developers to reduce development costs by replacing complex rule-based solutions with neural-based ones. The demonstration of *Struct2IUPAC* model is available online on *Syntelly* platform https://app.syntelly.com/smiles2iupac

Introduction

Before the Information Age chemical names were the universal language for the description of chemical structures. In the infancy of organic chemistry, there were no common rules for the naming of chemical compounds. However, the extensive growth of explored chemical space in the XIX century motivated chemists to harmonize chemical naming globally. In 1919 International Union of Pure and Applied Chemistry (IUPAC) was founded, and this non-commercial organization manages the development of chemical nomenclature. IUPAC publishes the Nomenclature of Organic Chemistry, commonly known as the "Blue Book." [1] The "Blue Book" provides guidelines to give unambiguous names for chemical compounds.

Nowadays there are several alternative representations for organic structures. For example, Simplified Molecular Input Line Entry System (SMILES) was designed to be handy both for humans and computational processing. However, IUPAC names still play an important role in organic chemistry. They are obligatory for processing chemicals in many regulated protocols, for example, for REACH registration in the EU, patent application submission in many countries, regulatory submission to FDA in the US. Most chemical journals require IUPAC names for organic structures too. Chemists are simply used to them. That is obvious that IUPAC names do not sink into oblivion in the nearest future.

In the past, chemists created IUPAC names manually. This process was error-prone because it requires deep knowledge of the nomenclature and mind concentration. [2] It is hard for humans to perform the naming process accurately because this process is rather algorithmic by its nature. Moreover, chemists are biased towards trivial names and do not want to discard the names that they are used to utilize. Computers alleviate this problem. Now chemists use software for the name generation widely.

The history of names generators begins from the pioneering Garfield's work. [3] However, the first "everyday use" software for chemists was created and popularised only at the end of the XX century. Now, there are several commercial programs for generating IUPAC names: ACD/Labs, ChemDraw, Marvin, IMnova IUPAC Name, etc. Also, there is a framework Lexi-

Chem TK that provides an application programming interface (API) for some programming languages.[4] Nevertheless, there is no an open-source tool for the structure-to-name translation. Licensing agreements with the existing solutions, like ChemDraw JS and LexiChem TK, do not allow embedding to other platforms.

Amazingly, there is an open-source tool for the name-to-structure translation: OPSIN developed by Daniel Lowe.[5] But, as we mentioned above, there is no one for the reverse problem: structure-to-name conversion.

Our work was motivated by a very practical request – we had needed a tool for structure-to-name translation. We estimated the development costs of such a tool "from scratch" as unacceptable. Instead, we built a Transformer-based neural network that can convert molecules from SMILES representations to IUPAC names and back. In this paper, we describe our solutions, discuss its advantages and disadvantages, and show that Transformer can provide something that resembles human's chemical intuition.

Materials and Methods

Database

Deep learning requires large amount of data. PubChem is a large freely-available collection of chemical compounds with annotations.[6] We used chemical structures and corresponding IUPAC names from this database. It had 94 726 085 structures in total. The processing and training on the full PubChem database is time expensive, and about 50M samples seems more than enough, so we splitted the database into two parts and used one half for training and the other one for testing. Structures that can not be processed by rdkit were removed resulting in 47 312 235 structures in the training set and 47 413 850 in the test set.

IUPAC and SMILES tokenizers

Tokenization – is a process of splitting sequences into pieces and demarking such pieces (tokens). It is a common preprocessing stage for language models. We use simple character-based SMILES tokenization and implemented a rules-based IUPAC tokenizer (*Fig. 1*). Our IUPAC tokenizer was manually designed and curated. We collected all suffixes (-one, -al, -ol, etc.), prefixes (-oxy, -hydroxy, -di, -tri, -tetra, etc.), trivial names (naphthalene, pyrazole, pyran, adamantane, etc.) and special symbols, numbers, stereochemical designations ((,), [,], -, N, R(S), E(Z), λ, etc.). Our tokenizer was able to correctly process more than 99% of molecules from PubChem[1].

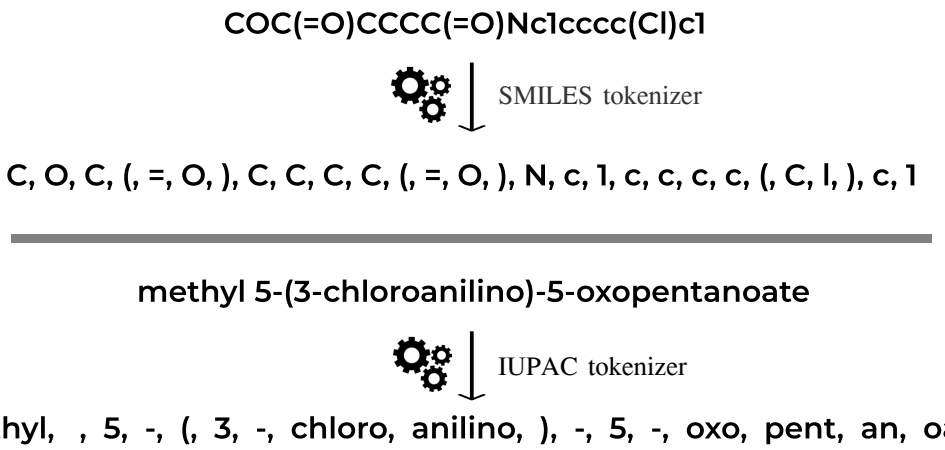


Figure 1: Demonstration of SMILES tokenization (top) and IUPAC names tokenization (bottom)

Transformer model

Transformer is a modern neural architecture designed by Google team especially to boost the quality of machine translation.[7] Since the origin, Transformer based networks notably boosted the performance on NLP problems leading to newsworthy GPT models.[8] Transformer has been successfully applied to chemical-related problem: the prediction of outcomes

[1]Due to a technical mistake, we did not include some tokens in the IUPAC tokenizer (for example, "selena"). This makes it impossible to process molecules containing these tokens. We excluded from the train and test sets all molecules that cannot be correctly tokenized. Since retraining Transformer is computationally intensive, we have not fixed it yet, however we are going to fix it in the next release.

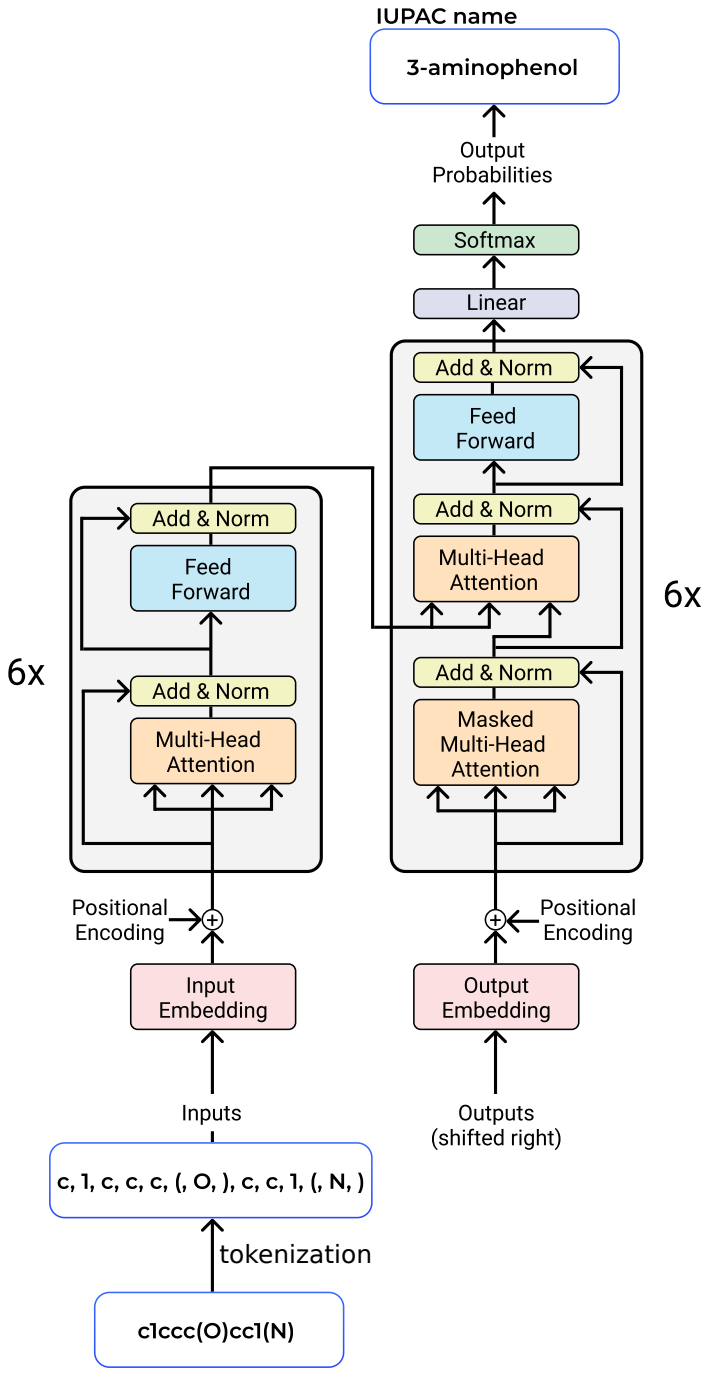


Figure 2: A scheme of *Struc2IUPAC* Transformer. Adopted from[7]

of organic reactions[9] , QSAR modelling[10] and creation of molecular representations.[11] We used standard Transformer architecture with 6 encoder and decoder layers, 8 attention heads. The attention dimension was 512 and the dimension of the feed-forward layer was 2048. We trained two models: *Struc2IUPAC* that converse SMILES strings to IUPAC names and *IUPAC2Srtuc* – that performs reverse operation. There was no ultimate practical use for *IUPAC2Srtuc* model because an open-source OPSIN can be successfully used instead. However, to study the performance of reverse converter and following the aesthetic symmetry principle we created these two models. The schema of our *Struc2IUPAC* model is given on *Fig. 2*

Production mode

Our scheme involves artificial neural networks, that means the solution is probabilistic by nature anyway. However – the generation of a chemical name is a precise task: a name can be either correct or wrong. We believe that the denial of translation is better than false conversion. Transformer can generate several versions of a sequence using beam search. By using OPSIN we can validate generated chemical names to guarantee that this name corresponds to the correct structure. So we can detect failures of our generator and do not display the wrong name.

Results and discussion

To validate the quality of our models we sampled randomly 100 000 molecules from the test set and calculated the percentage of correct predictions with different beam size. Our SMILES to IUPAC names converter, running in production mode, achieved absolute validity on a subset of 100 000 random molecules from the test set (excluding molecules that were not processed by the tokenizer). These results are impressive because we obtained a precise solution by a neural network. Transformer demonstrates the ability for the precise solution

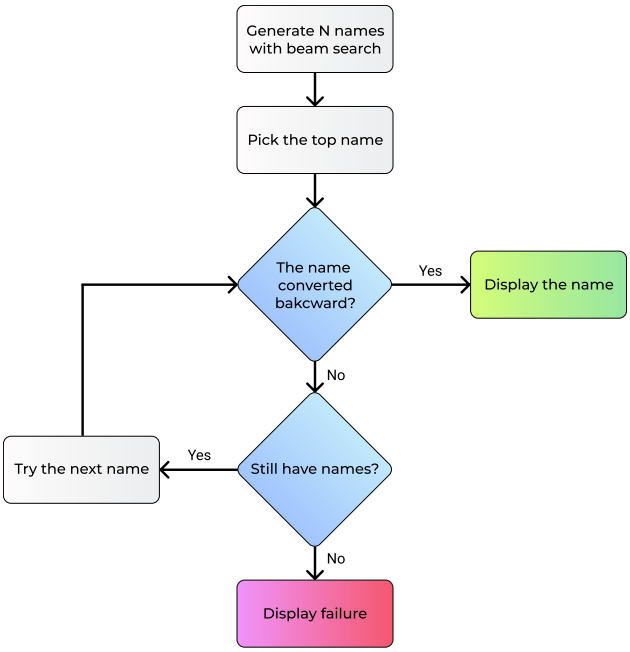


Figure 3: A scheme of "Production mode"

of an algorithmic problem, and that fact raises a new paradigm in software development. Before that there was a consensus opinion that ANN should not be used if a precise (strict algorithmic) solution is possible. Meanwhile, our approach is built on top of typical neural architecture and does not require algorithmic coding and chemical rules collection. The implementation of our system required about one and a half employee months for the whole pipeline. It is hard to estimate the resources that one needs to develop an algorithmic-based generator with competitive performance. Our preliminary estimation about the development of IUPAC names generator "from scratch," even using the source of OPSIN, would take more than a year for a small team. Anyway, we did not quantify our potential expenses, so we prefer to leave this question for a reader's opinion.

We mentioned absolute accuracy because we obtained strictly 100% of correct names at the 100k test subset. That does not mean that our model can guarantee 100% accuracy for any compound. It is known that Transformer fails at large sequences. Moreover, a common issue related to Transformer is a weakness in the processing of extra short sequences. To alleviate this problem we increased the weights of short sequences during training. For ex-

ample, testing the model manually, we found a problematic molecule: methane. To estimate the applicability domain of our model we took 1000 examples for each length from 2 to 10 with a step of 1 and from 10 to 300 with a step of 10. As a result, we found that our model achieves 100% performance in the interval from 10 to 75 SMILES tokens. The result of the experiment is given on *Fig. 4*.

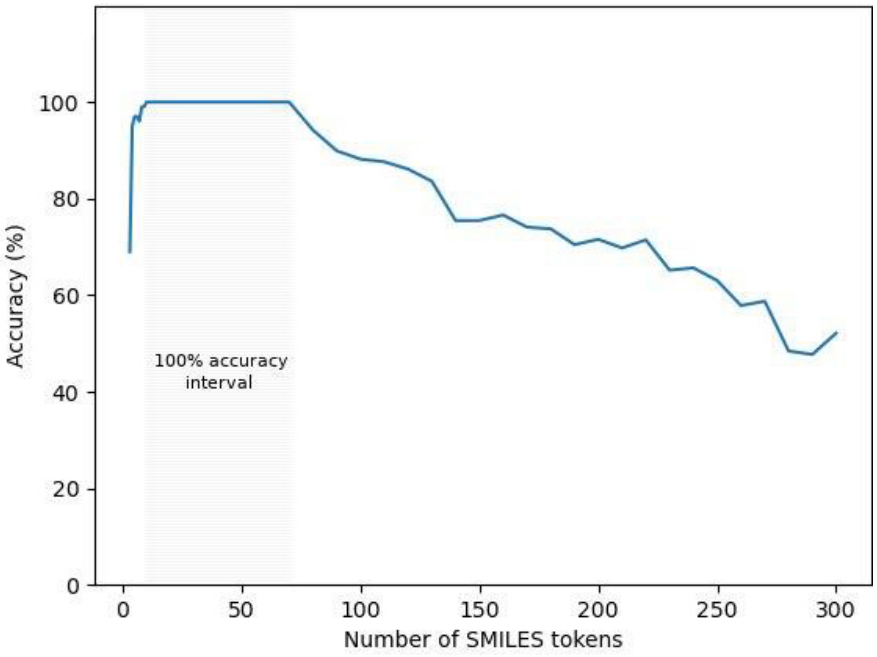


Figure 4: The dependence between model accuracy and the length of SMILES

We also depicted the distribution of sequence lengths on the test set (*Fig. 5*). The mean value of SMILES length is 33,6 tokens and IUPAC length is 27,6 tokens. So the majority of the PubChem molecules is within the applicability domain of our model.

We compared our IUPAC to SMILES Transformer model (*IUPAC2Struct*) with the rules-based tool OPSIN on the test set. Our converter achieved 99.99983% accuracy (17 mistakes per 100 000 molecules) and OPSIN performed 99.99985% (15 mistakes per 100 000 molecules).

Transformer is a heavy neural architecture. The application of Transformer can be slower than for algorithmic-based solutions. To understand this model's practical applicability in terms of execution time, we estimated the speed of name generation both on CPU and GPU.

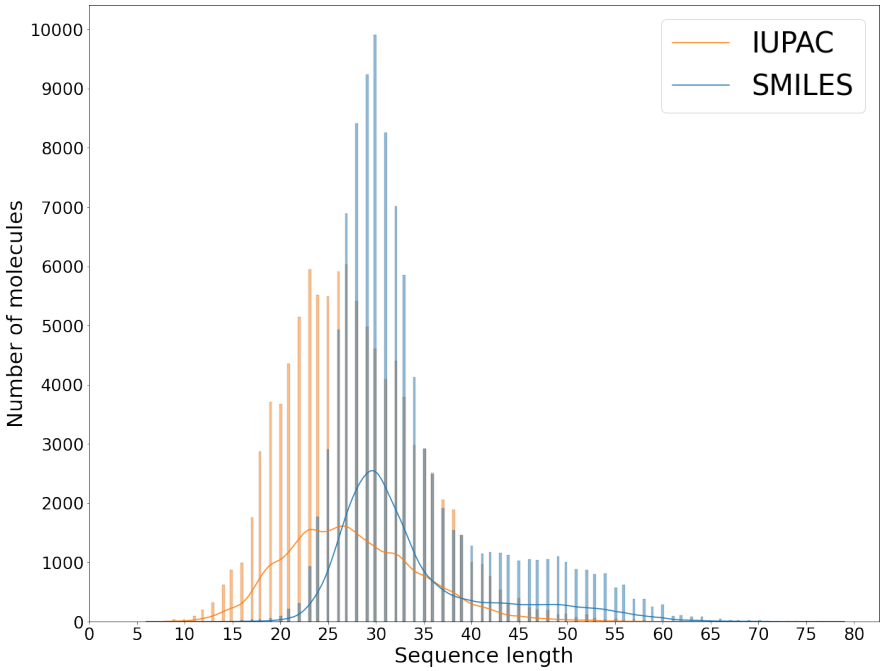


Figure 5: The distribution of the lengths of SMILES and IUPAC on the test set

We measured the dependence between the number of output IUPAC tokens and the time required for several Transformer runs without beam search and validation feedback loop. The result of the experiment is given on *Fig. 6*. Transformer consists of two parts: encoder and decoder. Encoder runs only once to read SMILES input, whereas decoder processes each output token. For this reason, only the output sequence length influences the time of execution. One can see that GPU is notably faster than CPU. GPU application requires less than 0.5 seconds even for chemical names with maximal length. This time-frame is acceptable for the practical usage.

Our soultion has a drawback – it requires prominent computational resources to train the model. Our production model has been trained for ten days on a machine with 4 Tesla V100 GPUs and 36 CPUs, which were also fully loaded. Still, nowadays human work hours are becoming more valuable, and machine hours more cheap, so our approach meets the modern paradigm.

The most intriguing ability of Transformer to operate with IUPAC nomenclature in a chemically reasonable way. One can see that the model can infer some chemical knowledge.

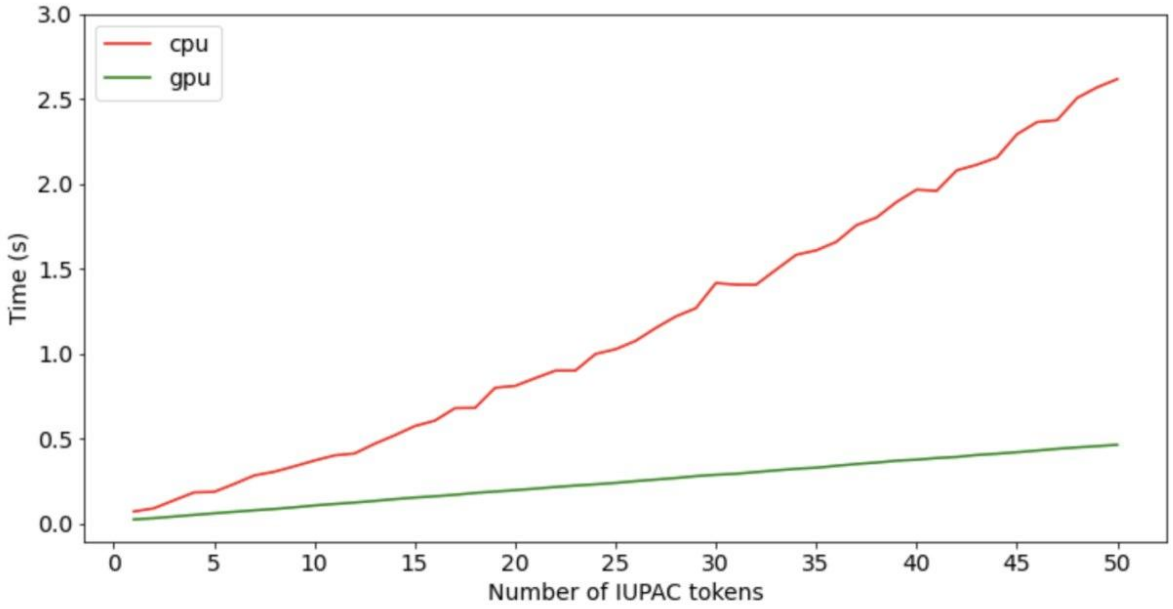


Figure 6: The correlation of mean time and output sequence length.

For example for a molecule on *Fig. 7* it generates two chemically valid names (OPSIN converts these names to the same structures) :

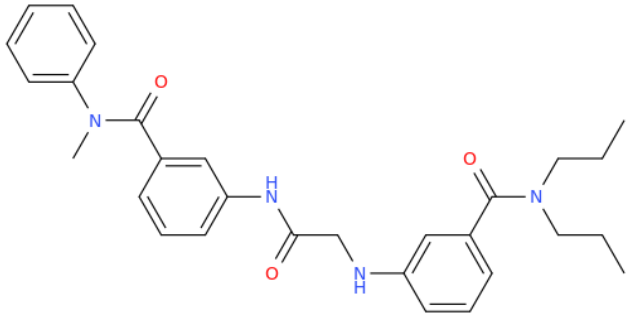


Figure 7: An example of a molecule with two correctly generated chemical names

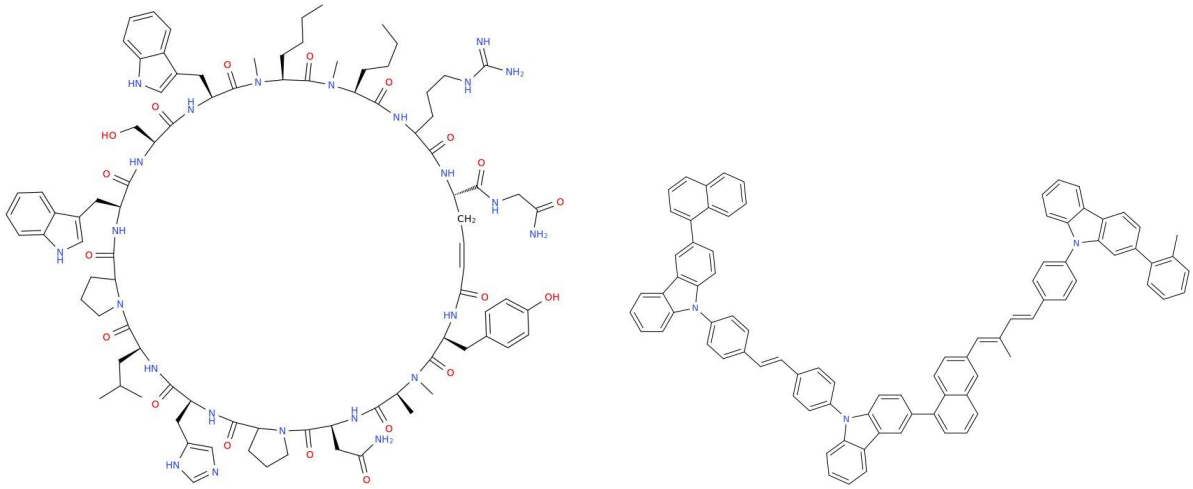
- *3-[[2-[3-(dipropylcarbamoyl)anilino]acetyl]amino]-N-methyl-N-phenylbenzamide*
- *N,N-dipropyl-3-[[2-[3-[methyl(phenyl)carbamoyl]anilino]-2-oxoethyl]amino]benzamide*

The name from PubChem is *3-[[2-[3-[methyl(phenyl)carbamoyl]anilino]-2-oxoethyl]amino]-N,N-dipropylbenzamide*. Both names generated by Transformers represents the correct structures and have chemical sense. It is hard to derive what structure represents a correct chemical name, because ChemDraw online generates for the same structure *3-(2-((3-(*

dipropylcarbamoyl)phenyl) amino)acetamido)-N-methyl-N-phenylbenzamide which is another variation of the name.

It is interesting to follow the behavior of Transformer outside the applicability domain. Our observations revealed that the performance of Transformer drops down with huge molecules. In the range from 200 to 300 tokens there are two common types of mistakes. The first one is the situation when the model loses the opening or closing squared bracket. It fails the whole structure due to a lexical mistake. That means that the model is undertrained on such an extra-size dataset. This behavior was expected because there were small amount very large molecules in the training set. The second typical case is losing a part of a large molecule. In this case, Transformer generates a chemically valid molecule, albeit that is shorter than the original. However, Transformer-based models are known for ability to work with thousands-long sequences, and we believe, that with enough large samples in a dataset, Transformer can achieve good performance on extra large molecules too.

Despite the fact that the accuracy of the model on the largest examples does not exceed 50%, we found the interesting examples of complex molecules for which the IUPAC name was correctly generated (*Fig. 8*).



(a) A cyclic peptide

(b) A carbazole derivative

Figure 8: Two examples of challenging molecules for which Transformer generates correct names

Conclusions

In this paper, we explore the Transformer-based solution for the generation of IUPAC chemical names using SMILES string. Our *Struc2IUPAC* model reached 100% accuracy within 10 to 75 SMILES length. Our reversed model reached accuracy close to 100%, comparable to open-source OPSIN software. We demonstrated that the computation time is generally applicable for production usage. We showed that our model operates well within a wide range of molecules length. Our research opens doors for a new paradigm in software development. We demonstrated that one could even replace a complex rule-based solution with modern "heavy" neural architectures. We believe that neural networks can now solve a wide range of so-called "exact" problems (problems for which an exact algorithm or solution exists) with compatible performance. We ask researchers to validate this idea for other algorithmic-based challenges. Our model is available for the community on *Syntelly* platform: (https://app.syntelly.com/smiles2iupac – text interface only for testing SMILES to IUPAC model, https://app.syntelly.com/individual – graphical interface for prediction of properties of organic compounds and IUPAC names).

Conflict of Interest

Maxim Fedorov and Sergey Sosnin are co-founders of Syntelly LLC. Lev Krasnov and Ivan Khokhlov are employees of Syntelly LLC

Acknowledgement

The authors acknowledge the use of Skoltech's Zhores GPU cluster[12] for obtaining the results presented in this paper.

Supporting Information Available

The Supporting Information is located on Zenodo (https://doi.org/10.5281/zenodo.4280815). It contains a subset of 100 000 chemical compounds that were used for testing Transformer, a subset of compounds on which OPSIN fails and compounds on which our *IUPAC2Smiles* model fails.

References

(1) *Nomenclature of Organic Chemistry: IUPAC Recommendations and Preferred Names 2013*; Royal Society of Chemistry.

(2) Eller, G. A. Improving the quality of published chemical names with nomenclature software. *Molecules 11*, 915–928.

(3) Garfield, E. Chemico-Linguistics: Computer Translation of Chemical Nomenclature. *Nature 192*, 192–192.

(4) Cannon, E. O. New Benchmark for Chemical Nomenclature Software. *J. Chem. Inf. Model. 52*, 1124–1131.

(5) 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. 51*, 739–753.

(6) Kim, S.; Chen, J.; Cheng, T.; Gindulyte, A.; He, J.; He, S.; Li, Q.; Shoemaker, B. A.; Thiessen, P. A.; Yu, B.; Zaslavsky, L.; Zhang, J.; Bolton, E. E. PubChem 2019 update: improved access to chemical data. *Nucleic Acids Research 47*, D1102–D1109.

(7) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv:1706.03762 [cs]*

(8) Brown, T. B. et al. Language Models are Few-Shot Learners. *arXiv:2005.14165 [cs]*

(9) Schwaller, P.; Laino, T.; Gaudin, T.; Bolgar, P.; Hunter, C. A.; Bekas, C.; Lee, A. A. Molecular Transformer: A Model for Uncertainty-Calibrated Chemical Reaction Prediction. *ACS Cent. Sci. 5*, 1572–1583, Publisher: American Chemical Society.

(10) Karpov, P.; Godin, G.; Tetko, I. V. Transformer-CNN: Swiss knife for QSAR modeling and interpretation. *J Cheminform 12*, 17.

(11) Chithrananda, S.; Grand, G.; Ramsundar, B. ChemBERTa: Large-Scale Self-Supervised Pretraining for Molecular Property Prediction. *arXiv:2010.09885 [physics, q-bio]*

(12) Zacharov, I.; Arslanov, R.; Gunin, M.; Stefonishin, D.; Bykov, A.; Pavlov, S.; Panarin, O.; Maliutin, A.; Rykovanov, S.; Fedorov, M. "Zhores" — Petaflops supercomputer for data-driven modeling, machine learning and artificial intelligence installed in Skolkovo Institute of Science and Technology. *Open Engineering 9*, 512–520.