

Translating the molecules: adapting neural machine translation to predict IUPAC names from a chemical identifier

Jennifer Handsel^{*,a}, Brian Matthews^{‡,a}, Nicola J. Knight^{‡,b}, Simon J. Coles^{‡,b}

^aScientific Computing Department, Science and Technology Facilities Council, Didcot, OX11 0FA, UK.

^bSchool of Chemistry, Faculty of Engineering and Physical Sciences, University of Southampton, Southampton, SO17 1BJ, UK.

KEYWORDS

seq2seq, InChI, IUPAC, transformer, attention, GPU

ABSTRACT

We present a sequence-to-sequence machine learning model for predicting the IUPAC name of a chemical from its standard International Chemical Identifier (InChI). The model uses two stacks of transformers in an encoder-decoder architecture, a setup similar to the neural networks used in state-of-the-art machine translation. Unlike neural machine translation, which usually tokenizes input and output into words or sub-words, our model processes the InChI and predicts the

IUPAC name character by character. The model was trained on a dataset of 10 million InChI/IUPAC name pairs freely downloaded from the National Library of Medicine's online PubChem service. Training took five days on a Tesla K80 GPU, and the model achieved test-set accuracies of 95% (character-level) and 91% (whole name). The model performed particularly well on organics, with the exception of macrocycles. The predictions were less accurate for inorganic compounds, with a character-level accuracy of 71%. This can be explained by inherent limitations in InChI for representing inorganics, as well as low coverage (1.4 %) of the training data.

INTRODUCTION

The International Union of Pure and Applied Chemistry (IUPAC) define nomenclature for both organic chemistry² and inorganic chemistry.³ Their rules are comprehensive, but are difficult to apply to complicated molecules. Although there are numerous commercial software packages that can generate the IUPAC name from a chemical structure, these are all closed source and their methodology is unknown to the general public. Correctly generating IUPAC names is therefore an open problem, and in particular is an issue faced by synthetic chemists who want to give a standard name to a new compound. Although canonical SMILES and InChI can be used for this purpose, a correct IUPAC name can be more human-readable.

Two of the most common formats used to represent molecular structure are InChI⁴ and SMILES.⁵ InChI explicitly records chemical formula, connectivity, and isomerism in a single string, although it has the disadvantage of being difficult to interpret by a human. SMILES are often easier to work with, as they explicitly show each atom in the compound, and use a system

of numbers and brackets to show connectivity. The disadvantage is that SMILES are not unique, although various canonicalization schemes are available.

Neural networks excel at making general predictions from a large set of training data. They have shown great success in natural language processing, and have been deployed by Google on their online translation service.¹ Compared to earlier efforts that needed human-designed linguistic features, modern machine translation learns these features directly from matched sentence pairs in the source and target language. This is done with a sequence-to-sequence (seq2seq) neural network, made up of an encoder, which projects the input sentence into a latent state, and a decoder, which predicts the correct translation from the latent state.

This paper presents a seq2seq neural network trained to predict the IUPAC name of a chemical from its unique InChI identifier.

Data Collection

A dataset of 100 million SMILES-IUPAC pairs was obtained from PubChem,⁶ and the SMILES were converted to InChI with Open Babel.⁷ The average character length of the InChI identifiers was 134 ± 60 , and 103 ± 43 for the IUPAC names. To simplify training, compounds were removed from the dataset if their InChI was longer than 200 characters, or their IUPAC name was longer than 150 characters. The resulting dataset of 94 million compounds was split into training data (90% of the data), with the remainder reserved for the validation and test sets. As IUPAC names of small molecules are usually easy to generate from procedural rules, validation and test sets were limited to compounds with an InChI length of 50 characters or greater. Due to the large volume of data available, the training set was reduced to a random sample of 10 million

compounds. For the same reason, 10,000 samples were chosen for the validation set, and 200,000 were chosen for the test set.

Experimental Setup

All experiments were carried out with the PyTorch version of OpenNMT 2.0.0rc2.⁸ The neural network had a transformer encoder-decoder architecture,⁹ with six layers in both the encoder and decoder (Figure 1). Each attention sub-layer had eight heads, and the feed-forward sub-layers had a hidden state size of 2048. Model weights were initialized with Glorot's method.¹⁰

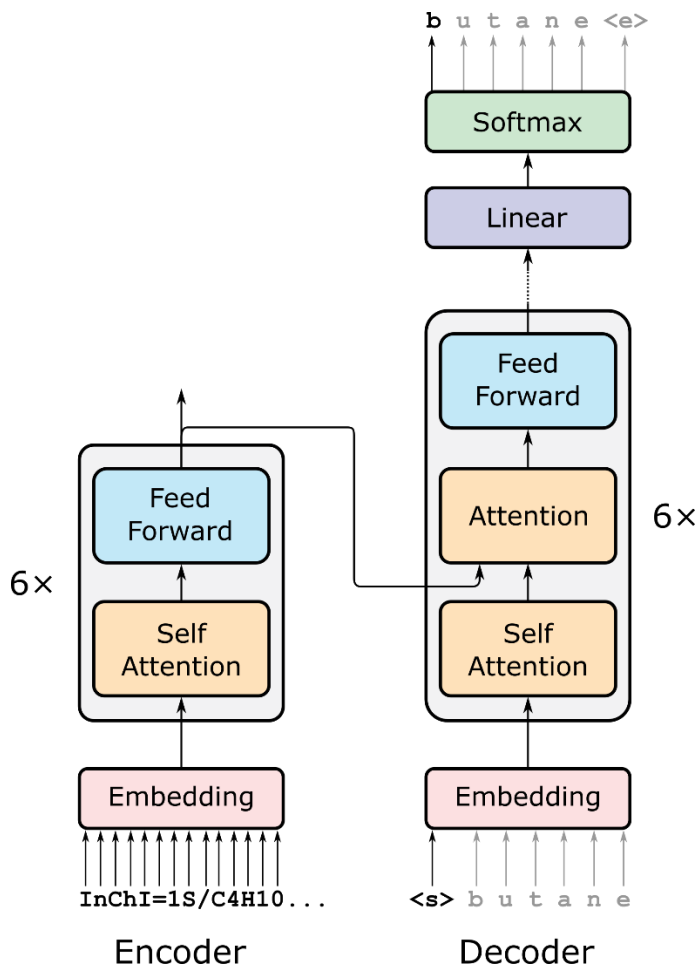


Figure 1. The encoder passes a numerical representation of the InChI to the decoder. The decoder is seeded with a start token, and its output is recursively re-input until it predicts an end token.

The input (InChI) and target (IUPAC name) were tokenized into characters on-the-fly with OpenNMT's pyonmttok module, with each character represented by a trainable embedding vector of length 512. Spaces were treated as separate tokens to enable detokenization of predicted names. The word vectors were augmented with positional encoding, to indicate the position of each character in the word. Character vocabulary was generated separately for InChI (66 characters) and IUPAC name (70 characters), using the whole training set.

The batch size was optimized for throughput: the optimal batch size was 4096 tokens which is equivalent to an average batch size of 30 compounds. Differing sample lengths within a batch were addressed by padding samples to a uniform length, and ignoring pad tokens when calculating model loss.

The model was regularized with a dropout rate of 0.1 applied to both dense layers and attentional layers.¹¹ The decoder output was regularized with label smoothing with magnitude 0.1.¹² The model was optimized with the ADAM variant¹³ of stochastic gradient descent, with beta_1 = 0.9 and beta_2 = 0.998. The loss function to be minimized was the standard cross-entropy loss averaged over all tokens in the batch, defined as

$$\ell = \frac{1}{N} \sum_{c \in \text{batch}} \sum_i p(c_i) \log \left(\frac{1}{q(c_i)} \right) \quad (1)$$

where N is the number of tokens in the batch, $p(c_i)$ is the ground-truth probability that token c is the i^{th} character in the alphabet (regularized with label smoothing as described above), and $q(c_i)$ is the corresponding probability predicted by the model. We report this as perplexity, defined as

$$\wp = e^{\ell} \quad (2)$$

which can be interpreted as the model’s token distribution being, on average, as unreliable as a uniform distribution with \wp branches. We also report token accuracy, defined as the proportion of correctly predicted characters in the IUPAC names, and whole-name accuracy, which is the proportion of IUPAC names predicted without error.

The learning rate was increased linearly to 0.0005 over 8000 warmup steps, then decayed with the reciprocal square root of the iteration number.⁹ Gradients were accumulated over 4 batches before updating parameters.

During training, the model was validated every 3200 batches on a validation set of 10,000 samples, as this size was found to be large enough to be representative. All models were trained until the validation accuracy stalled for three consecutive periods. Both training and validation used teacher forcing to improve convergence: rather than feeding predictions recursively into the decoder, each output character was predicted based on the ground truth from previous timesteps.¹⁴ Training took seven days on a Tesla K80 GPU, with throughputs of 6000 tokens/second (InChI) and 3800 tokens/second (IUPAC). The model was evaluated with a test set of 200,000 samples. The most probable IUPAC name was found using a beam search (width 10) and a length regularizer of strength 1.0.¹

RESULTS

We performed limited training on a subset of 1 million samples to determine appropriate model parameters, and trialed an LSTM architecture¹⁴ before settling on the transformer architecture described above. Training on 10 million samples converged with a validation token accuracy of 99.7 %, and a perplexity of 1.09 (Figure 2). The test set accuracies were 95.2 % (token-level) and 90.7 % (whole name level). The model was sensitive to dropout probability: increasing this parameter above 0.1 reduced the test accuracy by ten percentage points, and training without dropout reduced the accuracy by one percentage point.

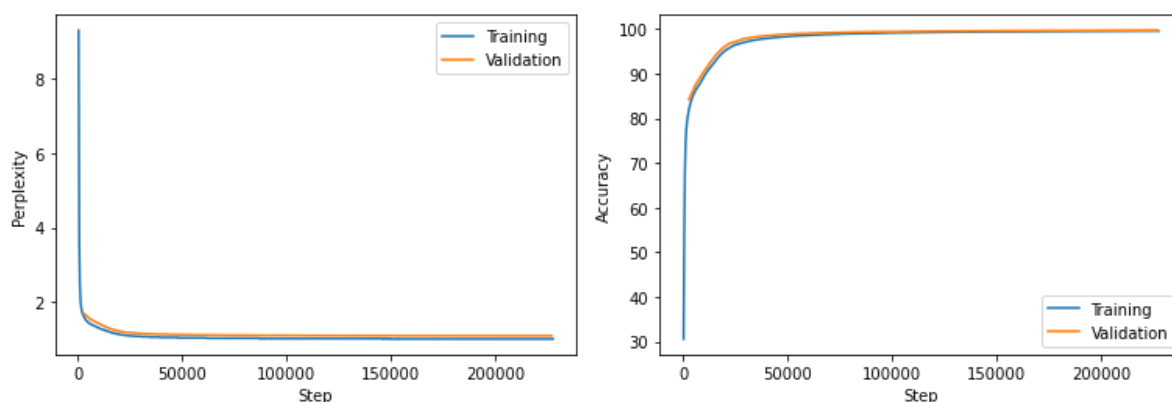


Figure 2. Perplexity and token accuracy during training of the InChI to IUPAC model.

The model can accurately predict the IUPAC name of a wide variety of organic molecules, with the exception of macrocycles. The model did not perform as well on inorganic compounds. Although we cannot provide a comprehensive list here, examples of correct predictions are shown in Table 5. Common molecules whose names our model predicts correctly. Table 5.

DISCUSSION

The encoder-decoder architecture works by projecting the input InChI into a latent vector, and then predicting each character in the IUPAC name sequentially (conditioned on the previous

predictions), until it predicts a stop token (Figure 1). The attentional layer in the decoder essentially calculates a similarity between characters in the input to characters in the predicted IUPAC name. Visualizing these correlations shows which parts of the input were important for predicting the output.

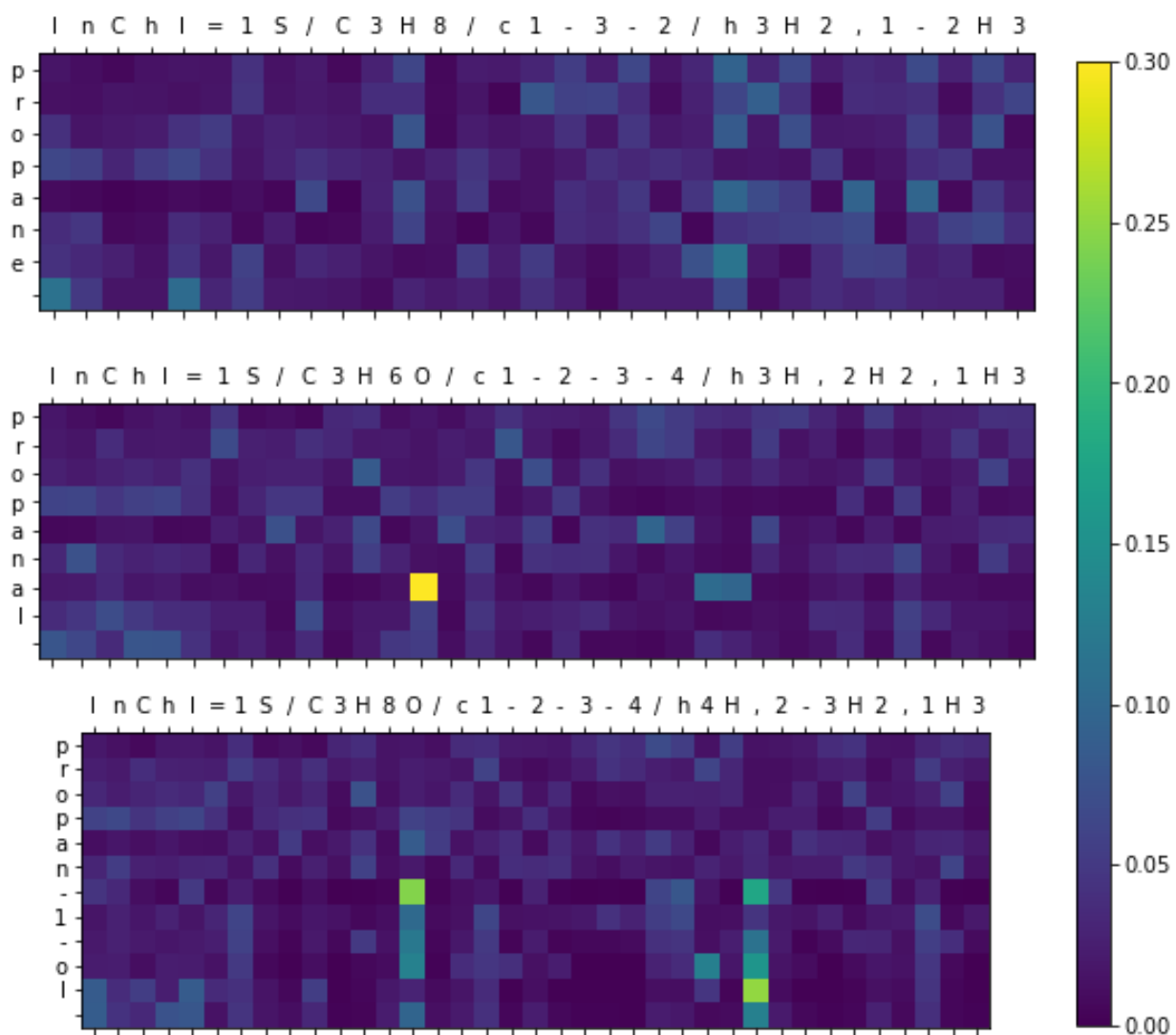


Figure 3. Attention coefficients from second to last layer of decoder, averaged over all heads.

All InChIs have a main layer with the chemical formula, connectivity, and hydrogen positions (in that order). When predicting the IUPAC name ‘propane’, no particular part of its InChI stands

out (Figure 3), but when predicting the suffix in ‘propanal’, the model pays attention to the oxygen element in the formula layer. Similarly, when predicting the ‘-1-ol’ suffix in propan-1-ol, the model pays particular attention to the oxygen atom (in the formula layer), and the fact that atom 4 (oxygen) has only one hydrogen (in the hydrogen layer).

We can probe the model further by selectively setting characters in the InChI to an out-of-vocabulary token. As one might expect, mutating the ‘O’ in propan-1-ol changes the predicted IUPAC name to propane. But the model makes a correct prediction when all of the formula apart from the ‘O’ is mutated, presumably because the connectivity and hydrogen layers still make ‘propan-1-ol’ the most likely candidate (Table 1).

InChI	Predicted IUPAC name
InChI=1S/C3H8O/c1-2-3-4/h4H,2-3H2,1H3	propan-1-ol
InChI=1S/C3H8#/c1-2-3-4/h4H,2-3H2,1H3	propane
InChI=1S/#####O/c1-2-3-4/h4H,2-3H2,1H3	propan-1-ol
InChI=1S/C3H8O/c1-2-3-4/#####2-3H2,1H3	propan-1-one
InChI=1S/C3H8O/c1-2-3-4/h4H,#####	prop-1-en-1-ol
#####C3H8O/c1-2-3-4/h4H,2-3H2,1H3	propan-1-ol

Table 1. Model predictions when mutating the InChI of propan-1-ol with an out-of-vocabulary token.

Isomers

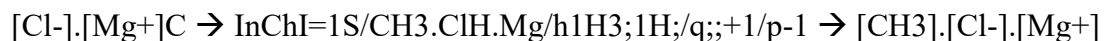
Stereoisomers are specified by an extra layer in the InChI representation. The inchi2iupac model can successfully label enantiomers and diastereomers, even when their InChI differs by a single character (Table 6).

There are issues with predicting isomerism that are related to limitations in the InChI standard. InChI does not recognize optical activity in molecules with Nitrogen in a bridgehead position in a polycyclic system (<https://www.inchi-trust.org/download/104/inchi-faq.pdf>), such as Tröger's base, and as such the model cannot assign isomerism in these cases. Tröger's base highlights another issue with the predicted IUPAC name: the model is unable to distinguish superscripts, so these are reproduced as ordinary characters. This is wholly because the training data did not contain superscript markers.

Inorganic and Organometallic Compounds

Token accuracy dropped to 71 % for compounds with an inorganic atom. This is partly due to only 1.4 % of the training set fitting into this category, which is a small proportion given the wide variety of inorganic compounds. Another reason is that InChI is inherently limited in representing complexes and organometallics. The standard representation encodes the metal as a separate component, and does not encode connectivity.¹⁵ While it is possible to add a 'reconnected' layer to InChI to represent coordinate or organometallic bonds (<https://www.inchi-trust.org/download/104/inchi-faq.pdf>), the latter is not part of the standard specification, and is rarely used in chemical databases. The result is that the model fails to predict the correct IUPAC name of many simple inorganic and organometallic compounds (Table 2).

To demonstrate the problem, converting a Grignard reagent from SMILES to InChI and back results in the magnesium being separated from the carbon:



For such molecules, the IUPAC prediction model would have to base its predictions on standardized SMILES, or another format able to denote connectivity.

There is also a question of the accuracy of the IUPAC names in the training set. Many of the IUPAC names of inorganic and organometallic compounds in the PubChem database are inaccurate, perhaps because they too were generated from InChI. An improved InChI to IUPAC model would need a better data source.

Common Name	IUPAC Name	Predicted Name
ferrocene	bis(η^5 -cyclopentadienyl)iron	cyclopenta-1,3-diene;iron(2+)
ferrocene (with reconnected layer)	bis(η^5 -cyclopentadienyl)iron	cyclopenta-1,3-diene;1,2,3,4-tetrafluorocyclopenta[b]pyrrol-4-ide;iron(2+)
hexaamminecobalt(III) chloride	hexaamminecobalt(III) chloride	azane;trichlorocobalt
methylmagnesium bromide	bromo(methyl)magnesium	magnesium;carbanide;bromide
n-butyllithium	butyllithium	lithium;butane

Table 2. Prediction of the IUPAC name of inorganic and organometallic compounds.

Charges, Radicals and Isotopes

Although the test set accuracy for charged molecules was only 77 %, the model is still able to predict the names of common charged species. However, due to low training set coverage, the model performs poorly when predicting the names of molecules with isotopic substitutions (Table 3). As InChI encodes point isotopic substitutions with an extra layer at the end, the model tends to ignore this information and predict the name of the non-substituted compound.

Common Name	IUPAC Name	Predicted Name
phenolate	phenolate	phenolate
ammonium	azanium	azanium
trimethylammonium	trimethylazanium	trimethylazanium
naphthalen-1-ylazanium	naphthalen-1-ylazanium	naphthalen-1-ylazanium
methyl carbene radical	methylene	methane
phenyl radical	phenyl	cyclohexatriene
phenoxy radical	phenyloxidanyl	cyclohexa-2,4-dien-1-one
heavy water	(² H ₂)Water	deuteriooxydiazene
tritiated water	(³ H ₂)Water	tritiooxytin
deuterated benzene	1,2,3,4,5,6-hexadeuteriobenzene	1,2,3,4,5,6-hexadeuteriobenzene
3-chloroalanine-Cl37	(³⁷ Cl)2-amino-2-chloroacetic acid	2-amino-2-chloroacetic acid

Table 3. Prediction of the IUPAC name of charged species, radicals, and molecules with isotopic substitutions.

Tautomers

Standard InChI can recognize certain tautomers,¹⁵ but when it does so, it encodes a general representation. This is powerful, but it does mean that information on the specific tautomer can be lost when converting to the InChI format. We found that InChI does not standardize keto-enol tautomers or enamine-imine tautomers, and that our model can correctly predict the IUPAC name of specific tautomers (Table 4).

However, for simple proton shifts InChI encodes the structure in the general form. For γ -lactam / γ -lactim tautomers, our model predicted the lactam form. A similar effect can be seen with resonance forms of the five-membered ring in guanine. While it is possible to specify the

resonance form with a non-standard fixed-H layer, there were no such examples in our training set and our model still predicts the standard IUPAC name for guanine even when an alternative tautomer is specified. The same can be seen on charged species with a mobile proton: the oxazolium ion can have a protonated oxygen or nitrogen, but standard InChI does not specify the charge center and standardizes the location of the proton.

Overall, our model performed well on the limited range of tautomers we tested, considering the limitations of standard InChI.

Common Names	IUPAC Name	Predicted Name
cyanamide (enamine-imine)	cyanamide / methanediimine	cyanamide / methanediimine
glucic acid (keto-enol)	2-hydroxypropanedial / 2,3-dihydroxyprop-2-enal	2-hydroxypropanedial / 2,3-dihydroxyprop-2-enal
γ -Lactam (lactam-lactim)	pyrrolidin-2-one / 3,4-dihydro-2 <i>H</i> -pyrrol-5-ol	pyrrolidin-2-one
guanine	2-amino-1,7-dihydropurin-6-one	2-amino-1,7-dihydropurin-6-one
guanine (resonance specified with fixed H layer)	N/A	2-amino-1,7-dihydropurin-6-one
Oxazolium (mobile proton)	4,5-dihydro-1,3-oxazol-3-ium	4,5-dihydro-1,3-oxazol-3-ium

Table 4. Prediction of the IUPAC name of tautomers.

Alternative Models

In machine translation, there are several alternatives to the character-level approach used in the current paper. Byte-pair encoding¹⁶ and unigram language models¹⁷ attempt to tokenize the input into common clusters of characters in the training data, and have been very successful in machine translation. We performed a limited number of experiments with these encoding methods, but could not match the accuracy of the character-level approach.

The transformer network can be trained with SMILES instead of InChI, and achieve a similar accuracy. However, these models did not generalize to alternative SMILES representations that were not present in the training data. It may be possible to regularize a SMILES to IUPAC model with a function that randomly permutes the possible representations of each training sample, but such an implementation is not trivial and may take far longer to train. An attempt to do so with two alternative SMILES representations per compound did not generalize.

Alternative IUPAC Names

To our knowledge, there are six different commercial software packages that can generate an IUPAC name from a structure. There's a certain amount of variability in the predicted name between the different packages, and some allow the user to specify different IUPAC standards. This variability is reflected in the IUPAC names found in PubChem⁶ and ChemSpider,¹⁸ presumably because they use different software to generate the names. For example, Codeine has the names:

(4R,4aR,7S,7aR,12bS)-9-methoxy-3-methyl-2,4,4a,7,7a,13-hexahydro-1H-4,12-methanobenzofuro[3,2-e]isoquinolin-7-ol

(5 β ,6 β ,9 α ,13 α ,14 α)-3-Methoxy-17-methyl-7,8-didehydro-4,5-epoxymorphinan-6-ol

As our model was trained on PubChem data, its predictions may differ from the IUPAC name found on other online services.

Conclusions and Future Work

Our InChI to IUPAC model works very well for organics, but has some clear shortfalls, mainly due to known issues with InChI and the composition of our training data. It is suitably robust to be deployed as a service, as long as the service is constrained to the types of molecule on which the model performs well. Our model will be integrated into the United Kingdom's Physical Sciences Data-science Service, as part of a wider physico-chemical property prediction platform.


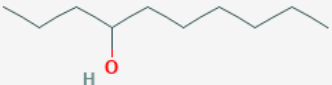
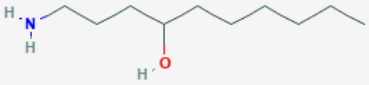
This work could be improved by retraining the model (or a separate model) with better data for inorganics and organometallics. Due to the inherent issues with InChI, this model would need to be trained on SMILES or directly on the chemical graph. The former would either need to rely on a canonical form of SMILES, or randomly permute equivalent SMILES strings at each iteration, which could result in slow convergence. The latter could be achieved using a hybrid model with a graph neural network for the encoder, and a transformer for the decoder.

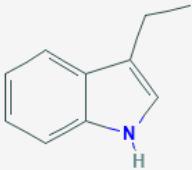
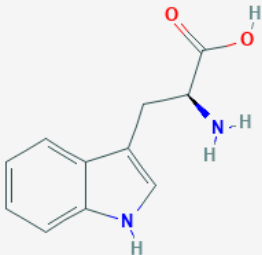
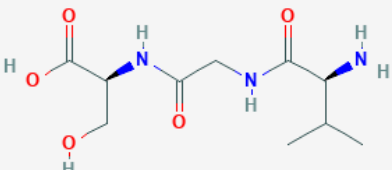
Data and Software Availability

The dataset of 100 million compounds was obtained from PubChem's⁶ public ftp server (<ftp://ftp.ncbi.nlm.nih.gov/pubchem/Compound/Extras/>) in two separate files (CID-SMILES.gz and CID-IUPAC.gz). The two files were combined by merging on the CID column (the internal identifier used by PubChem) with GNU join (<http://www.gnu.org/software/coreutils/>). The integrity of the data was verified with Open Babel,⁷ which is freely available under the GNU General Public License (<https://github.com/openbabel/openbabel>). This was done by reading in

each SMILES string and excluding 132,421 structures that could not be parsed. The integrity of the IUPAC column was verified by excluding names with unbalanced parentheses, using a simple regular expression. The SMILES from PubChem were converted to InChI and canonical SMILES using Open Babel's pybel module. The neural machine translation software, OpenNMT,⁸ is freely available under the MIT license (<https://github.com/OpenNMT/OpenNMT-py>).

TABLES

	<p>Common Name: decane</p> <p>Name: decane</p> <p>Prediction: decane</p>
	<p>Common Name: 4-decanol</p> <p>Name: decan-4-ol</p> <p>Prediction: decan-4-ol</p>
	<p>Name: 1-aminodecan-4-ol</p> <p>Prediction: 1-aminodecan-4-ol</p>

	<p>Name: 3-ethyl-1<i>H</i>-indole</p> <p>Prediction: 3-ethyl-1<i>H</i>-indole</p>
	<p>Common Name: L-Tryptophan</p> <p>Name: (2<i>S</i>)-2-amino-3-(1<i>H</i>-indol-3-yl)propanoic acid</p> <p>Prediction: (2<i>S</i>)-2-amino-3-(1<i>H</i>-indol-3-yl)propanoic acid</p>
	<p>Common Name: Val-Gly-Ser Peptide</p> <p>Name: (2<i>S</i>)-2-[[2-[[[(2<i>S</i>)-2-amino-3-methylbutanoyl]amino]acetyl]amino]-3-hydroxypropanoic acid</p> <p>Prediction: (2<i>S</i>)-2-[[2-[[[(2<i>S</i>)-2-amino-3-methylbutanoyl]amino]acetyl]amino]-3-hydroxypropanoic acid</p>

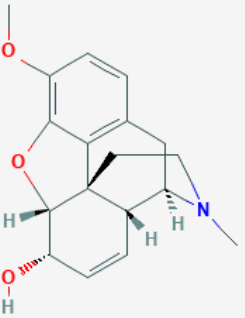
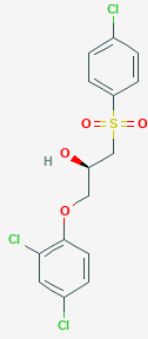
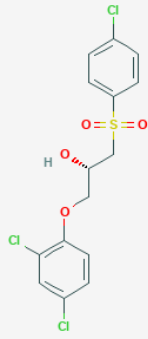
	<p>Common Name: Codeine</p> <p>Name: (4<i>R</i>,4<i>aR</i>,7<i>S</i>,7<i>aR</i>,12<i>bS</i>)-9-methoxy-3-methyl-2,4,4<i>a</i>,7,7<i>a</i>,13-hexahydro-1<i>H</i>-4,12-methanobenzofuro[3,2-<i>e</i>]isoquinolin-7-ol</p> <p>Prediction: (4<i>R</i>,4<i>aR</i>,7<i>S</i>,7<i>aR</i>,12<i>bS</i>)-9-methoxy-3-methyl-2,4,4<i>a</i>,7,7<i>a</i>,13-hexahydro-1<i>H</i>-4,12-methanobenzofuro[3,2-<i>e</i>]isoquinolin-7-ol</p>
---	--

Table 5. Common molecules whose names our model predicts correctly.

	<p>InChI: InChI=1S/C15H13Cl3O4S/c16-10-1-4-13(5-2-10)23(20,21)9-12(19)8-22-15-6-3-11(17)7-14(15)18/h1-7,12,19H,8-9H2/t12-/m1/s1</p> <p>Name: (2<i>R</i>)-1-(4-chlorophenyl)sulfonyl-3-(2,4-dichlorophenoxy)propan-2-ol</p> <p>Prediction: (2<i>R</i>)-1-(4-chlorophenyl)sulfonyl-3-(2,4-dichlorophenoxy)propan-2-ol</p>
	<p>InChI: InChI=1S/C15H13Cl3O4S/c16-10-1-4-13(5-2-10)23(20,21)9-12(19)8-22-15-6-3-11(17)7-14(15)18/h1-7,12,19H,8-9H2/t12-/m0/s1</p> <p>Name: (2<i>S</i>)-1-(4-chlorophenyl)sulfonyl-3-(2,4-dichlorophenoxy)propan-2-ol</p> <p>Prediction: (2<i>S</i>)-1-(4-chlorophenyl)sulfonyl-3-(2,4-dichlorophenoxy)propan-2-ol</p>

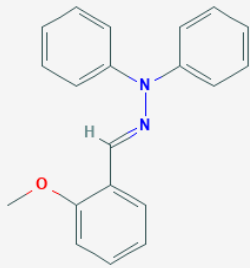
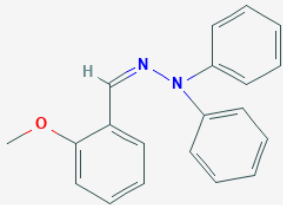
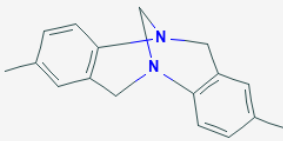
		<p>InChI: InChI=1S/C20H18N2O/c1-23-20-15-9-8-10-17(20)16-21-22(18-11-4-2-5-12-18)19-13-6-3-7-14-19/h2-16H,1H3/b21-16+</p> <p>Name: <i>N</i>-[(E)-(2-methoxyphenyl)methylideneamino]-<i>N</i>-phenylaniline</p> <p>Prediction: <i>N</i>-[(E)-(2-methoxyphenyl)methylideneamino]-<i>N</i>-phenylaniline</p>
		<p>InChI: InChI=1S/C20H18N2O/c1-23-20-15-9-8-10-17(20)16-21-22(18-11-4-2-5-12-18)19-13-6-3-7-14-19/h2-16H,1H3/b21-16-</p> <p>Name: <i>N</i>-[(Z)-(2-methoxyphenyl)methylideneamino]-<i>N</i>-phenylaniline</p> <p>Prediction: <i>N</i>-[(Z)-(2-methoxyphenyl)methylideneamino]-<i>N</i>-phenylaniline</p>
		<p>InChI: InChI=1S/C17H18N2/c1-12-3-5-16-14(7-12)9-18-11-19(16)10-15-8-13(2)4-6-17(15)18/h3-8H,9-11H2,1-2H3</p> <p>Names: (±)-Tröger's base</p> <p>(1<i>S</i>,9<i>S</i>)- / (1<i>R</i>,9<i>R</i>)-5,13-dimethyl-1,9-diazatetracyclo[7.7.1.0^{2,7}.0^{10,15}]heptadeca-2(7),3,5,10(15),11,13-hexaene</p> <p>Prediction: 5,13-dimethyl-1,9-diazatetracyclo[7.7.1.02,7.010,15]heptadeca-2(7),3,5,10(15),11,13-hexaene</p>

Table 6. Prediction of the IUPAC name of isomers not present in the training set.

ASSOCIATED CONTENT

The following files are available free of charge.

SMILES of all molecules presented in the manuscript (smiles.csv)

AUTHOR INFORMATION

Corresponding Author

*Email: jennifer.handsel@stfc.ac.uk

Author Contributions

The manuscript was written through contributions of all authors. All authors have given approval to the final version of the manuscript. ‡These authors contributed equally.

Funding Sources

This work was funded by the Physical Sciences Data-science Service under EPSRC grant number EP/S020357/1.

ACKNOWLEDGMENT

We thank the SCARF team in Scientific Computing for providing access to high performance computing clusters. J. H. thanks Keith Butler for helping with batch scripts for the computing cluster.

ABBREVIATIONS

LSTM, long short-term memory; seq2seq, sequence to sequence; InChI, International Chemical Identifier; SMILES, Simplified molecular-input line-entry system; IUPAC, International Union of Pure and Applied Chemistry

REFERENCES

(1) Wu, Y.; Schuster, M.; Chen, Z.; Le, Q. V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; Klingner, J.; Shah, A.; Johnson, M.; Liu, X.; Kaiser, L.; Gouws, S.; Kato, Y.; Kudo, T.; Kazawa, H.; Stevens, K.; Kurian, G.; Patil, N.; Wang, W.; Young, C.; Smith, J.; Riesa, J.; Rudnick, A.; Vinyals, O.; Corrado, G.; Hughes, M.; Dean, J. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *CoRR* **2016**, *abs/1609.08144*.

(2) *Nomenclature of Organic Chemistry*; Royal Society of Chemistry: Cambridge, 2013. <https://doi.org/10.1039/9781849733069>.

(3) Hartshorn, R. M.; Hellwich, K.-H.; Yerin, A.; Damhus, T.; Hutton, A. T. Brief Guide to the Nomenclature of Inorganic Chemistry. *Pure Appl. Chem.* **2015**, *87* (9–10). <https://doi.org/10.1515/pac-2014-0718>.

(4) Heller, S. R.; McNaught, A.; Pletnev, I.; Stein, S.; Tchekhovskoi, D. InChI, the IUPAC International Chemical Identifier. *J. Cheminform.* **2015**, *7* (1). <https://doi.org/10.1186/s13321-015-0068-4>.

(5) James, C. A. OpenSMILES specification. <http://opensmiles.org/opensmiles.html>.

(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 in 2021: New Data Content and Improved Web Interfaces. *Nucleic Acids Res.* **2021**, *49* (D1). <https://doi.org/10.1093/nar/gkaa971>.

- (7) O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. Open Babel: An Open Chemical Toolbox. *J. Cheminform.* **2011**, *3* (1). <https://doi.org/10.1186/1758-2946-3-33>.
- (8) Klein, G.; Hernandez, F.; Nguyen, V.; Senellart, J. The OpenNMT Neural Machine Translation Toolkit: 2020 Edition. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*; Association for Machine Translation in the Americas: Virtual, 2020; pp 102–109.
- (9) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł. ukasz; Polosukhin, I. Attention Is All You Need. In *Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U. V, Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc., 2017; Vol. 30, pp 5998–6008.
- (10) Glorot, X.; Bengio, Y. Xavier Initialization. *J. Mach. Learn. Res.* *2010b*. ISSN **2010**, *15324435*.
- (11) Hinton, G. E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors. *CoRR* **2012**, *abs/1207.0580*.
- (12) Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. *CoRR* **2015**, *abs/1512.00567*.
- (13) Kingma, D. P.; Ba, J. Adam: A Method for Stochastic Optimization. 2017.
- (14) Luong, M.-T.; Pham, H.; Manning, C. D. Effective Approaches to Attention-Based Neural Machine Translation. *CoRR* **2015**, *abs/1508.04025*.

- (15) Warr, W. A. Many InChIs and Quite Some Feat. *J. Comput. Aided. Mol. Des.* **2015**, 29 (8). <https://doi.org/10.1007/s10822-015-9854-3>.
- (16) Sennrich, R.; Haddow, B.; Birch, A. Neural Machine Translation of Rare Words with Subword Units. *CoRR* **2015**, *abs/1508.07909*.
- (17) Kudo, T.; Richardson, J. SentencePiece: A Simple and Language Independent Subword tokenizer and Detokenizer for Neural Text Processing. *CoRR* **2018**, *abs/1808.06226*.
- (18) Pence, H. E.; Williams, A. ChemSpider: An Online Chemical Information Resource. *J. Chem. Educ.* **2010**, 87 (11). <https://doi.org/10.1021/ed100697w>.