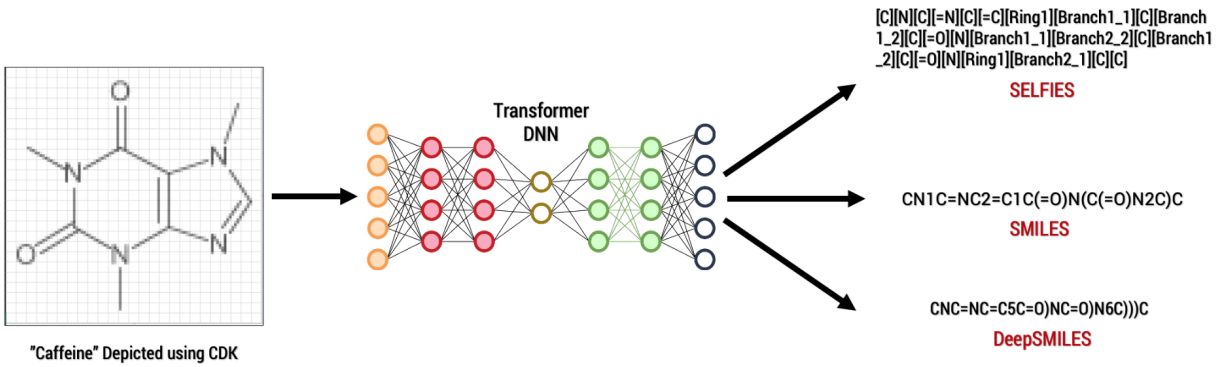
Performance of chemical structure string representations for chemical image recognition using transformers

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

[1] Institute for Inorganic and Analytical Chemistry, Friedrich-Schiller-University Jena, Lessingstr. 8, 07743 Jena, Germany
[2] Institute for Bioinformatics and Chemoinformatics, Westphalian University of Applied Sciences, August-Schmidt-Ring 10, D-45665 Recklinghausen, Germany
*Corresponding author email: christoph.steinbeck@uni-jena.de

Abstract

The use of molecular string representations for deep learning in chemistry has been steadily increasing in recent years. The complexity of existing string representations, and the difficulty in creating meaningful tokens from them, lead to the development of new string representations for chemical structures. In this study, the translation of chemical structure depictions in the form of bitmap images to corresponding molecular string representations was examined. An analysis of the recently developed DeepSMILES and SELFIES representations in comparison with the most commonly used SMILES representation is presented where the ability to translate image features into string representations with transformer models was specifically tested. The SMILES representation exhibits the best overall performance whereas SELFIES guarantee valid chemical structures. DeepSMILES performs in between SMILES and SELFIES, InChIs are not appropriate for the learning task. All investigations were carried out with publicly available datasets and the code used to train and evaluate the models has been made available to the public.



Keywords

Chemical data extraction, Deep learning, Neural networks, Optical chemical structure recognition, chemical string representations, DeepSMILES, SMILES, SELFIES

Introduction

Deep Learning in chemistry with large and complex neural networks requires appropriate representations of the chemical structure to encode the molecular information in question. Often, a textual representation of chemical structures - a character string - is utilized for this purpose. A number of new textual representations for chemical structures have recently been developed due to apparent inefficiencies of the already existing and widely used SMILES [1] format.

SMILES has a number of issues in its application for deep learning [2]. For example, branches are introduced with an opening bracket "(" and closed at a subsequent string position with a closing bracket ")". The same holds for ring openings and closures which are marked by a number where a ring opens or closes. The precise placement of such markers at potentially distant places in the textual string poses problems for many deep neural networks.

New representations like DeepSMILES [2] and SELFIES [3] were developed to overcome these problems, e.g. by grouping ring-opening and closures.

In a recent study [4], we encountered a similar problem with SMILES representations which eventually led to an implementation based on SELFIES. Interestingly, by utilizing so-called transformer networks the situation changed: Transformers turned out to work best with SMILES. Here we report our findings of a case study for the chemical image to chemical structure translation.

Methods

Data

In this study, all data were taken from the ChEMBL [5] and PubChem [6] databases. The data was originally downloaded in SDF format. Using the Chemistry Development Kit (CDK) [7] the chemical structures were converted into SMILES strings with and without stereochemistry information. After the SMILES conversion, the DECIMER filtering rules [4] were applied to obtain a balanced dataset. Then two datasets were created, one containing SMILES without stereochemistry and one with stereochemistry information.

The filtering rules for the datasets without stereochemistry included the following,

- have a molecular weight of fewer than 1500 Daltons,
- not possess counter ions,
- only contain the elements C, H, O, N, P, S, F, Cl, Br, I, Se and B,
- not contain isotopes of Hydrogens (D, T),
- have 3 - 40 bonds,
- only contain implicit hydrogens, except in functional groups,
- have less than 40 SMILES tokens,
- no stereochemistry was allowed.

After filtering, a total of 1,655,225 molecules were obtained from ChEMBL. With the RDKit [8] MaxMin algorithm, equally diverse training and test subsets were created so that both sets cover a similar chemical space.

A set of 3 million molecules from PubChem was used to investigate whether the network performs better with more data. Here, the dataset was twice as large as the ChEMBL dataset. The PubChem dataset was filtered using the same rules as above, and the RDKit MaxMin algorithm was again applied to create the test set.

For the datasets with stereochemistry, a total of 1,653,833 molecules were obtained from ChEMBL and 3 million molecules from PubChem. Again, the RDKit MaxMin algorithm was used to select diverse training and test subsets. Table 1 provides an overview of the datasets.

The dataset with stereochemistry obtained from ChEMBL was a little smaller than the corresponding dataset without stereochemistry since stereochemistry adds new characters to SMILES, thereby lowering the number of available molecules due to the applied ruleset. With PubChem, however, the dataset size can be managed, since PubChem is much larger than ChEMBL.

Table 1: Overview of the datasets used in this study.

Database name	**ChEMBL**		**PubChem**	
Dataset name	Dataset 1	Dataset 2	Dataset 3	Dataset 4
Dataset description	Without stereochemistry	With stereochemistry	Without stereochemistry	Without stereochemistry
Train dataset size	1536000	1536000	3072000	3072000

Test dataset size	119225	117833	250000	250000

To visualize the training and test dataset diversity, Morgan fingerprints [9] were generated using RDKit and a Principal Component Analysis (PCA) [10] was performed on the generated fingerprints, see Figure 1.

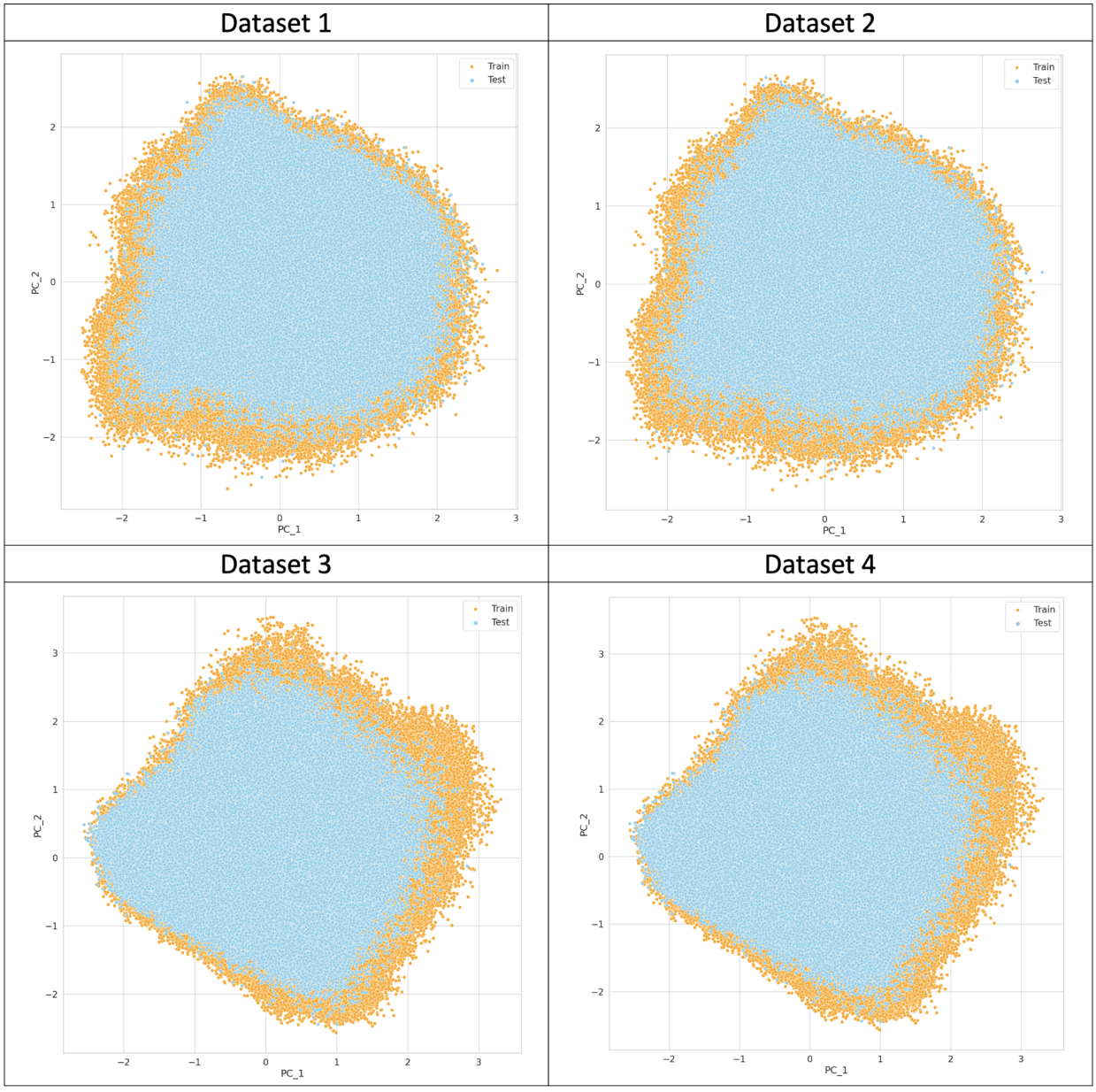


Figure 1: PCA plots visualising similar diversity of training and test datasets listed in Table 1.

Textual Data

The generated molecule sets were then converted into different textual representations of the chemical structures: SMILES, DeepSMILES, SELFIES and InChIs [11] and then split into tokens. For SELFIES this was a straightforward process since they already inherit a token-like word representation. Thus, SELFIES were split into tokens by using a space between the squared brackets “] [”.

For splitting SMILES, DeepSMILES and InChIs into tokens another set of rules had to be applied. They were split after,

- every heavy atom,
- every open bracket and close bracket “(”,”)”,
- every bond symbol “=”,”#”,
- every single-digit number and
- all the characters inside the squared brackets were retained as-is.

The "InChI=1S/" token was kept as one single token. As it is common in all InChIs, it was not used as a token during training but was later added to the predicted strings during post-processing to evaluate the results.

In addition to the token count, the maximum string length found in the datasets was calculated. This refers to the length of the longest string available in each dataset and also plays a role during training and testing. During training, the input vocabulary size which the network can handle was determined by comparing the number of tokens with the maximum length. In cases where the maximum length found in a dataset was smaller than the number of tokens available in the dataset, the input vocabulary size would be the number of tokens, otherwise, it would be the maximum length. During testing, the maximum length was used to determine when to stop predicting a structure if the end token is not met. Table 2 summarizes the number of tokens and the maximum string length found in each dataset. Datasets with stereochemistry information contain more tokens than datasets without. SELFIES representation led to more tokens than SMILES or DeepSMILES representation. InChIs had the lowest number of tokens but the largest maximum length of the longest string. With datasets 1 and 2, it became clear that InChIs perform significantly worse than the other string representations, so they were omitted in training and testing datasets 3 and 4.

Table 2: Overview of the token count and the maximum length.

Database name	ChEMBL				PubChem			
	Dataset 1 (Without stereochemistry)		Dataset 2 (With stereochemistry)		Dataset 3 (Without stereochemistry)		Dataset 4 (With stereochemistry)	
	Number of tokens	Maximum Length of String	Number of tokens	Maximum Length of String	Number of tokens	Maximum Length of String	Number of tokens	Maximum Length of String
SMILES	52	81	104	81	73	87	125	83
SELFIES	69	80	187	88	98	84	205	90
DeepSMILES	76	93	127	101	97	93	148	96
InChI	32	236	41	273	--	--	--	--

Image Data

A production-quality bitmap image of each molecule was generated with the CDK Structure Diagram Generator (SDG) at a resolution of 300x300 pixels. The generated images were saved in 8-bit PNG format. Each image contains a single structure only.

The features from these images were extracted as vectors by using the pre-trained weights of the 'noisy student' [12] trained EfficientNet-B3 [13] model. The extracted image features were then saved into NumPy arrays [14]. These topics were discussed in detail in our previous publication [15].

The extracted image features combined with the tokenized textual data were then converted into TFRecords [16]. TFRecords are binary records that can be used to train a model faster using Cloud Tensor Processing Units (TPUs) [17] on the Google Cloud Platform (GCP).

For training purposes, each TFRecord contains 128 data points consisting of 128 image feature vectors accompanied by 128 tokenized string representations. The TFRecords were generated on an in-house server and then moved into a Google Cloud Storage bucket.

Each dataset contains the same image data but different string representations.

Network, Training and Testing.

In this work, we use the same network as in [15], a transformer-based network model similar to the "Base model" as explained in Google's publication, *Attention Is All You Need* [18]. The only difference between the original implementation and ours is that we used four encoder-decoder layers instead of six. The network was coded with Python 3 using TensorFlow 2.3 [19] on the backend.

Throughout the training process, all models were trained on TPU v3-8 devices in the Google cloud. When comparing the training speed and network performance, a batch size of 1024 was found to be an adequate choice. The models were trained until the training loss had converged. In total, we trained eight models on datasets 1 and 2, and six models on datasets 3 and 4.

Once the models were fully converged, they were tested on an in-house server equipped with a GPU. To determine how many of the predictions were identical, the predictions were compared to the original strings. After the identical prediction calculations, all the predictions were converted to SMILES.

An analysis of the Tanimoto [20] similarity index was conducted between the original and predicted SMILES using PubChem fingerprints available in the CDK. The Tanimoto similarity indices helped to understand how well the network was able to learn chemical string representations since occasionally the predictions were not identical but only similar to the original structures and even for isomorphic structures, there may be many different SMILES.

Results and Discussion

The purpose of this study was to examine different chemical string representations that are available for deep learning in chemistry by their performance on chemical image to string translation using transformer networks. Predictions were valid if the images could be translated into structures correctly.

All the test results were assessed as following,

- Valid DeepSMILES/SELFIES/InChI: The predicted DeepSMILES, SELFIES and InChIs that could decode back into SMILES strings. The rest were deemed invalid.
- Valid SMILES: Predicted SMILES and decoded SMILES which could be parsed to calculate the Tanimoto similarity calculations. The rest were classified as invalid SMILES.
- Identical Predictions: This calculation identified how many predictions matched the original string representations. This was accomplished by using a one-to-one character string match. If a single character was wrong in the predicted string, it was considered as a wrong prediction.
- Average Tanimoto: The Tanimoto similarity between the original and predicted SMILES was calculated from the valid SMILES and the average Tanimoto similarity index was calculated against the entire test dataset.

- Tanimoto 1.0 Percentage: The number of Tanimoto 1.0 counts on the calculated Tanimoto similarity indices of the valid SMILES and the percentage against the total test dataset.

Results for the ChEMBL dataset

From ChEMBL two datasets were obtained to train and test, one with stereochemistry (Dataset 1) and one without stereochemistry (Dataset 2). Table 3 summarizes the test results obtained with training on images from Dataset 1.

Table 3: Test results on dataset 1 (without stereochemistry)

	SMILES	**DeepSMILES**	**SELFIES**	**InChI**
Test dataset size	119225	119225	119225	119225
Invalid DeepSMILES/SELFIES/InChI	0.00%	0.07%	0.00%	30.79%
Valid DeepSMILES/SELFIES/InChI	100.00%	99.93%	100.00%	69.21%
Invalid SMILES	0.35%	0.10%	0.00%	0.00%
Valid SMILES	99.65%	99.83%	100.00%	69.21%
Identical Predictions (String match)	80.87%	78.67%	68.85%	64.28%
Tanimoto 1.0 Percentage (Not Identical)	86.30%	84.11%	73.88%	65.53%
Average Tanimoto	0.97	0.97	0.95	0.69

SMILES performed best in comparison to the other representations. Comparing the identical predictions and the Tanimoto 1.0 count, SMILES based models were more accurate. This could be due to fewer tokens in the SMILES language space. Additionally, the maximum SMILES string was shorter than the rest. As a result, the model learns the representations better. Even though the InChIs have fewer tokens compared to the other representations, having a lesser number of tokens increases the maximum length of each string compared to the other representations, which ultimately creates more errors for learning and predicting. In addition, valid InChI predictions were predominantly identical to the original string.

Even though SELFIES has the most valid structures, the overall predictivity of the SELFIES-based model was lower than that of SMILES and DeepSMILES. Overall, SMILES were simpler to learn - but for guaranteed valid structures, SELFIES were the best option.

To estimate the impact of stereochemistry, the same procedure was repeated with Dataset 2 where the models were trained from scratch. The results are summarized in Table 4.

Table 4: Test results on Dataset 2 (with stereochemistry)

	SMILES	**DeepSMILES**	**SELFIES**	**InChI**
Test dataset size	117833	117833	117833	117833
Invalid DeepSMILES/SELFIES/InChI	0.00%	0.11%	0.00%	32.99%
Valid DeepSMILES/SELFIES/InChI	100.00%	99.89%	100.00%	67.01%
Invalid SMILES	0.81%	0.64%	0.08%	0.00%
Valid SMILES	99.19%	99.25%	99.92%	67.01%
Identical Predictions (String match)	78.16%	77.07%	66.59%	59.10%
Tanimoto 1.0 Percentage (Not Identical)	85.02%	83.89%	72.07%	63.49%
Average Tanimoto	0.97	0.97	0.94	0.66

The inclusion of stereochemistry information led to a lowered accuracy. For DeepSMILES and InChIs, the number of invalid predictions increased. Additionally, the fraction of invalid SMILES increased for all representations except InChIs. After parsing all InChIs, there were only valid SMILES.

SMILES with stereochemistry reduced the overall predictability and accuracy due to the new artefacts added to the images. In addition, one should consider that the overall token count in these datasets increased due to stereochemistry with additional tokens being introduced.

SMILES were overall best to get the most accurate predictions. Since InChIs showed a significantly inferior performance, it was decided to restrict further investigations to SMILES, DeepSMILES and SELFIES.

Results for the PubChem dataset

In order to determine model improvement with increasing data size, the training and test data were doubled by utilizing data from PubChem. As pointed out above, InChIs were omitted in subsequent testing.
The number of molecules available in PubChem is currently 110 million. For this work, 3 million molecules for training and 250,000 molecules for testing were obtained by random selection: The resulting tokens were carefully compared to those in the ChEMBL dataset to ensure a similar token set. Using the PubChem derived datasets with (Datasets 3) and without

stereochemistry (Datasets 4), the same training and testing procedures were repeated and the same evaluation procedure was used as before. For the dataset, without stereochemistry (Dataset 3) the results are summarized in Table 5.

Table 5: Test results on Dataset 3 (without stereochemistry)

	SMILES	DeepSMILES	SELFIES
Test dataset size	250000	250000	250000
Invalid DeepSMILES/SELFIES/InChI	0.00%	0.08%	0.00%
Valid DeepSMILES/SELFIES/InChI	100.00%	99.92%	100.00%
Invalid SMILES	0.22%	0.08%	0.00%
Valid SMILES	99.78%	99.84%	100.00%
Identical Predictions (String match)	88.62%	87.52%	82.96%
Tanimoto 1.0 Percentage (Not Identical)	92.19%	91.08%	86.42%
Average Tanimoto	0.98	0.98	0.97

By comparison of Table 5 with Table 3, it can be concluded that the data increase improved the model's performance in general. Again, SMILES show the best accuracy on the test results and SELFIES still retain 100% valid structures.

DeepSMILES falls somewhere between these two. Although DeepSMILES has more valid structures than SMILES, when considering overall accuracy, the DeepSMILES format falls behind: comparing DeepSMILES to SELFIES, DeepSMILES has a better accuracy because of its SMILES like representation, but its overall number of valid structures lags behind SELFIES (see figure 2).

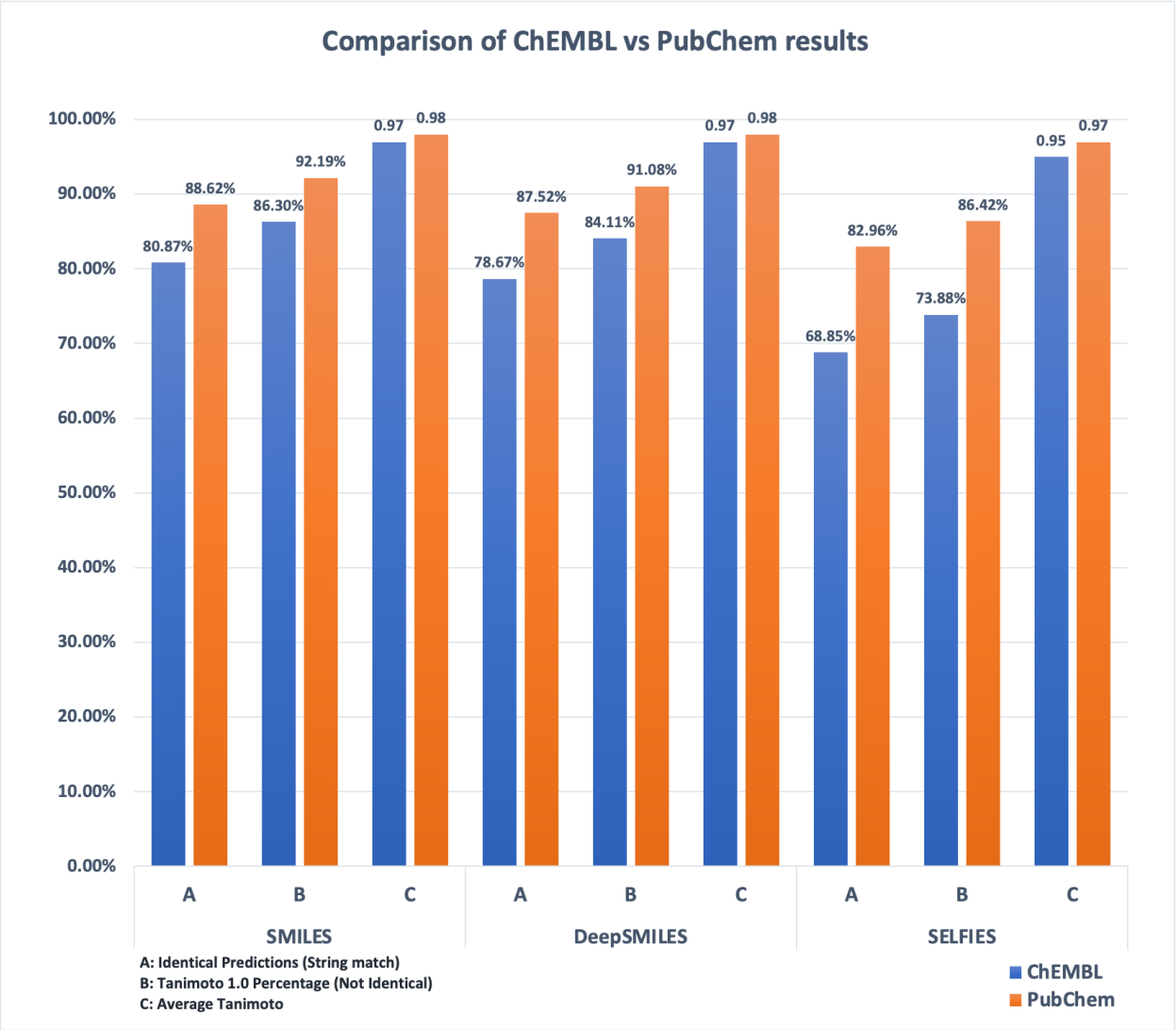


Figure 2: Comparison of identical predictions, Tanimoto 1.0 count and average Tanimoto of ChEMBL vs PubChem datasets (without stereochemistry).

A summary of the results for Dataset 4 with stereochemistry information can be found in table 6.

Table 6: Test results on Dataset 4 (with stereochemistry)

	SMILES	DeepSMILES	SELFIES
Test dataset size	250000	250000	250000
Invalid DeepSMILES/SELFIES/InChI	0.00%	0.06%	0.00%
Valid DeepSMILES/SELFIES/InChI	100.00%	99.94%	100.00%
Invalid SMILES	0.34%	0.05%	0.00%
Valid SMILES	99.66%	99.88%	100.00%
Identical Predictions (String match)	85.80%	83.80%	79.73%
Tanimoto 1.0 Percentage (Not Identical)	91.69%	90.60%	86.00%
Average Tanimoto	0.98	0.98	0.97

Compared to table 4, the results in table 6 showed that increasing the dataset size also increased the overall accuracy. Datasets with stereochemistry did not perform as well as datasets without. However, the overall accuracy did increase compared to the datasets derived from ChEMBL. In addition, all of the SELFIES predictions which were decoded back into SMILES were valid, providing 100% valid structures in comparison with Table 4. SMILES again performed best in terms of predictability and accuracy, see Figure 3.

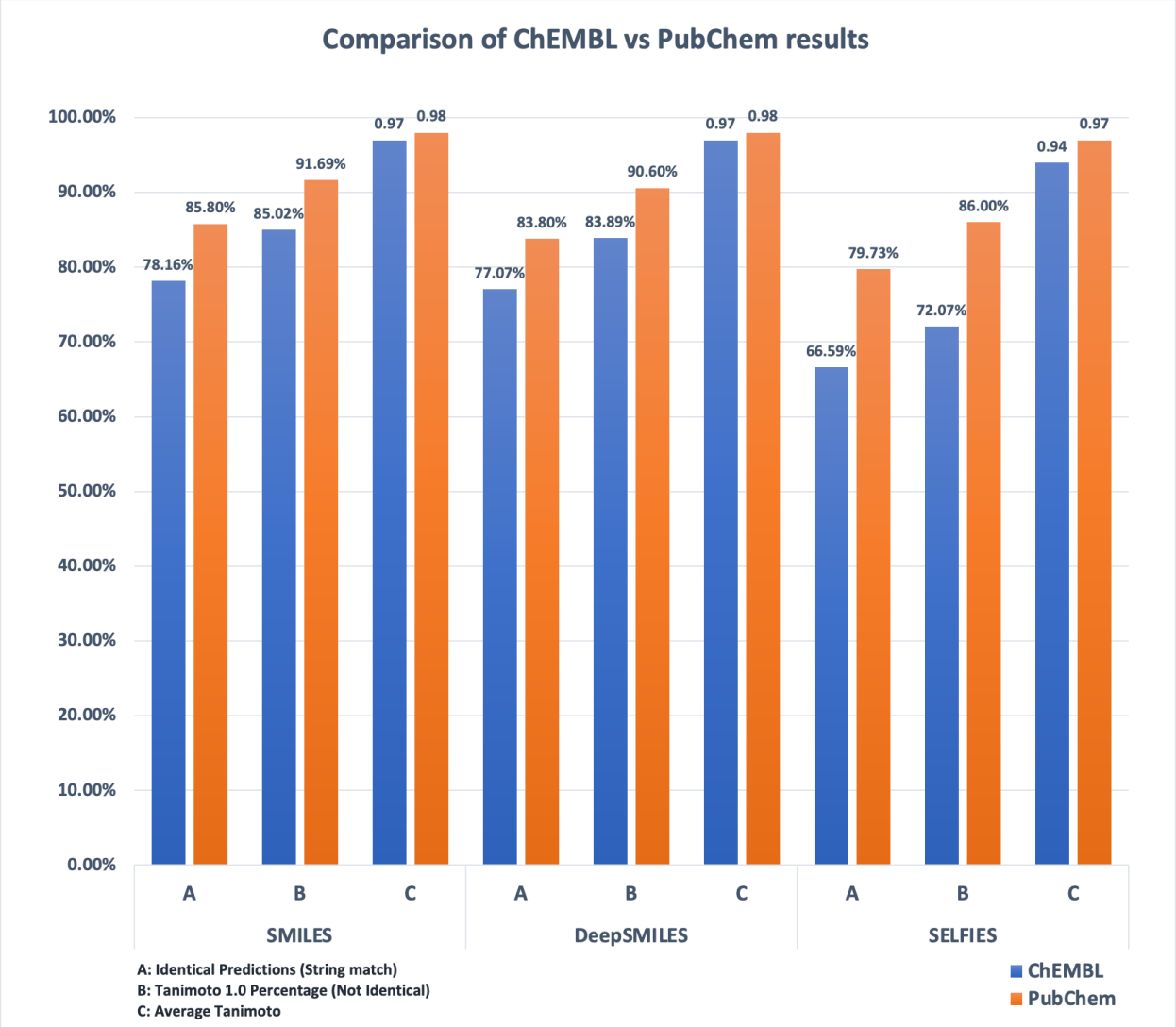


Figure 3: Comparison of identical predictions, Tanimoto 1.0 count and average Tanimoto of ChEMBL vs. PubChem datasets (with stereochemistry).

Conclusion

The performance of different textual chemical structure representations for the chemical image to structure translation using transformers was investigated. The most accurate models were obtained by using the SMILES representation. Using SELFIES, however, we were able to produce models that led to predictions with fewer invalid structures. DeepSMILES models always fell between SMILES and SELFIES. To ensure that the models improve similarly with more data, the datasets were scaled up: The results showed the same comparative performance. For most accurate predictions, models should be trained using SMILES, for maximizing valid structures SELFIES should be used.

The valid structures generated after decoding from SELFIES and DeepSMILES showed that the SELFIES decoding was superior to DeepSMILES decoding. SMILES and DeepSMILES should always be used with a set of rules on how to split them into meaningful tokens. SELFIES does not require this. There were fewer tokens in DeepSMILES than in SELFIES because the representation was similar to that in SMILES.

Since SELFIES encoding is a promising endeavor under active development, improved SELFIES variants could reach or even surpass the SMILES predictivity with the additional advantage of a 100% structural validity.

Availability of data and materials

The scripts are available at: https://github.com/Kohulan/DECIMER_Short_Communication
The data is available at: https://doi.org/10.5281/zenodo.5155036

Abbreviations

CDK - Chemistry Development Kit

DECIMER - Deep lEarning for Chemical ImagE Recognition

GCP - Google Cloud Platform

GPU - Graphical Processing Unit

InChI - International Chemical Identifier

PCA - Principal Component Analysis

SDF - Structure Data File

SDG - Structure Diagram Generator

SELFIES - Self-referencing embedded strings

SMILES - Simplified molecular-input line-entry system

TPU - Tensor Processing Units

Declarations

Competing interests

AZ is co-founder of GNWI - Gesellschaft für naturwissenschaftliche Informatik mbH, Dortmund, Germany.

Funding

The authors acknowledge funding by the Carl-Zeiss-Foundation. Open Access funding enabled and organized by Projekt DEAL.

Authors' contributions

KR developed the software and performed the data analysis. CS and AZ conceived the project and supervised the work. All authors contributed to and approved the manuscript.

Acknowledgements

We are grateful for the company Google making free computing time on their TensorFlow Research Cloud infrastructure available to us.

References

1 D. Weininger, *J. Chem. Inf. Comput. Sci.*, 1988, **28**, 31–36.
2 N. O'Boyle and A. Dalke, , DOI:10.26434/chemrxiv.7097960.v1.
3 M. Krenn, F. Häse, A. Nigam, P. Friederich and A. Aspuru-Guzik, *Mach. Learn.: Sci. Technol.*, 2020, **1**, 045024.
4 K. Rajan, A. Zielesny and C. Steinbeck, *J. Cheminform.*, 2020, **12**, 65.
5 A. Gaulton, A. Hersey, M. Nowotka, A. P. Bento, J. Chambers, D. Mendez, P. Mutowo, F. Atkinson, L. J. Bellis, E. Cibrián-Uhalte, M. Davies, N. Dedman, A. Karlsson, M. P. Magariños, J. P. Overington, G. Papadatos, I. Smit and A. R. Leach, *Nucleic Acids Res.*, 2017, **45**, D945–D954.
6 S. Kim, J. Chen, T. Cheng, A. Gindulyte, J. He, S. He, Q. Li, B. A. Shoemaker, P. A. Thiessen, B. Yu, L. Zaslavsky, J. Zhang and E. E. Bolton, *Nucleic Acids Res.*, 2019, **47**, D1102–D1109.
7 C. Steinbeck, Y. Han, S. Kuhn, O. Horlacher, E. Luttmann and E. Willighagen, *J. Chem. Inf. Comput. Sci.*, 2003, **43**, 493–500.
8 G. Landrum and Others, *URL http://www. rdkit. org/, https://github. com/rdkit/rdkit*.
9 H. L. Morgan, *J. Chem. Doc.*, 1965, **5**, 107–113.
10 S. Wold, K. Esbensen and P. Geladi, *Chemometrics Intellig. Lab. Syst.*, 1987, **2**, 37–52.
11 S. R. Heller, A. McNaught, I. Pletnev, S. Stein and D. Tchekhovskoi, *J. Cheminform.*, 2015, **7**, 23.
12 Q. Xie, M.-T. Luong, E. Hovy and Q. V. Le, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10687–10698.
13 M. Tan and Q. Le, in *International Conference on Machine Learning*, PMLR, 2019, pp. 6105–6114.
14 S. van der Walt, S. C. Colbert and G. Varoquaux, *Computing in Science Engineering*, 2011, **13**, 22–30.
15 K. Rajan, A. Zielesny and C. Steinbeck, .
16 TFRecord and tf.train.Example, https://www.tensorflow.org/tutorials/load_data/tfrecord, (accessed 7 April 2021).
17 T. Norrie, N. Patil, D. H. Yoon, G. Kurian, S. Li, J. Laudon, C. Young, N. Jouppi and D. Patterson, *IEEE Micro*, 2021, **41**, 56–63.

18 A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, *arXiv [cs.CL]*, 2017.
19 M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu and X. Zheng, *arXiv [cs.DC]*, 2016.
20 T. T. Tanimoto, *An Elementary Mathematical Theory of Classification and Prediction*, International Business Machines Corporation, 1958.