ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

Transformers for Molecular Graph Generation

Tim Cofala[1] and Oliver Kramer[1] *

1- University of Oldenburg - Department of Computing Science
26129 Oldenburg - Germany

Abstract.
This work introduces an autoregressive generative model for graphs which is based on the transformer architecture and applied to the domain of molecular graph generation. Utilizing the multi-head self-attention mechanism to directly model distributions over atoms and bonds, it can sample new molecular graphs in an autoregressive manner. The benchmark framework MOSES is used to compare the proposed approach to other state-of-the-art molecule generation models. It is shown that the model is capable of generalizing from the training data to generate novel and realistic molecules.

1 Introduction

Generative neural networks for graphs have been extensively studied in the last years, e.g. [1, 2, 3]. Graphs are a common representation used in various problem domains, and generative models can play an important part in discovering new graphs. Furthermore, they can be used to traverse the search space in direction of desired properties. Particularly in molecular design, graphs are a common data structure in addition to the string-based representation. Although some graph-based molecule generation models have been introduced in the past, most of these models rely on recurrence to capture the underlying distributions over nodes and edges. In this work, we introduce an autoregressive molecule generation model based on transformers utilizing their self-attention mechanism. To investigate, whether the generative model captures the underlying structural properties of its training data, we evaluate the quality of the generated molecules with MOSES [4] and compare it to other state-of-the-art models for molecule generation.

2 Related Work

Li et al. [1] introduced a graph generative model, which they also adapted to the domain of molecules. They employed an architecture combining message passing networks and recurrent neural networks to autoregressively generate graphs. Building up on this, Liao et al. [2] improved the autoregressive sampling strategy. By sampling a block of one or several rows of the adjacency matrix at a time, the number of necessary decision step during graph generation is drastically reduced. Since multiple edges are generated at the same time, a mixture of multiple Bernoulli distributions is used to decide on the occurrence

*We thank the German Research Foundation (DFG) for supporting our work within the Research Training Group SCARE (GRK 1765/2).

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

of edges. These mixture components can capture correlations between edges in one block. GraphRNN [3] is another approach for the generation of graphs, which uses a graph-level RNN and edge-level RNN for the generation of graphs.

Molecular generation models often operate on string-based representations (commonly SMILES), since these are easy to access and process. Different approaches have been introduced for the generation of molecules as strings, e.g. recurrent neural networks [5] and variational autoencoder [6]. Despite its straightforward implementation, a string-based representation bears some disadvantages. Generated SMILES strings are not necessary valid, and additional model capacity has to be attributed to learning the language's formal rules [4]. To overcome these limitations, Podda et al. [7] introduced a fragment-based approach for the generation of SMILES strings. Rather than sampling symbol by symbol, this approach first extracts a vocabulary of fragments from the training data, on which the language model is then trained.

Directly processing and sampling molecular graphs was e.g. demonstrated by Li et al. [1]. Furthermore, Jin et al. [8] presented a generative model for molecules based on a junction tree variational autoencoder. Rather than generating molecules on an atom-per-atom basis, this approach combines molecular substructures which are extracted from the training data, assuring the generation of valid molecules. A few examples of the application of the transformer architecture for molecular graphs can be found in literature. Cai and Lam [9] demonstrated that it is possible to use transformers for graph-to-sequence learning. Additionally, the transformer architecture has been utilized for molecular property prediction [10, 11]. Yoo et al. [12] presented a transformer specialized in processing graphs. Nodes are processed as tokens, while the edges are incorporated in the transformer's self-attention mechanism. The model was mainly investigated for property prediction on smaller molecules, but can also be applied to graph generation.

3 Transformer For Graph Generation

In the following section, a transformer-based generative model for graphs is introduced. Albeit applicable to various types of graphs, in this work it is evaluated as a generative model in the domain of molecular graphs.

3.1 Data Representation

This work utilizes the data sets provided by the benchmark framework MOSES, which are based on the ZINC database and contain 1.6 million training and 176 thousand test molecules [4]. For our approach, the molecules are converted to a sequence of atom tokens and a corresponding adjacency matrix. The adjacency matrix marks missing bonds as zeros, while bonds are represented by a number unique to the respective bond type – in this work single, double and triple bonds are considered. Since molecules are processed by the transformer in a column-wise manner, the lower left half of the adjacency matrix is masked with zeros. Additionally, the columns are padded with zeros to a user defined maximum

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

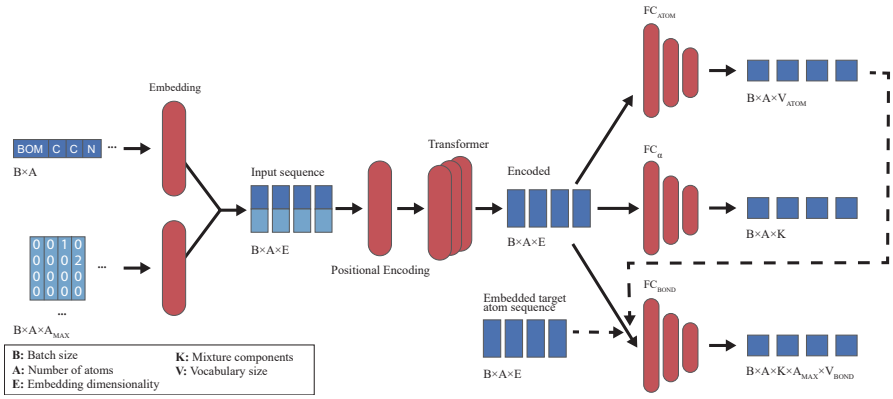

Fig. 1: Overview of the model architecture

number of atoms, to assure a fixed number of elements per column. Molecules are processed in batches. If molecules in a batch are of different sizes, the input sequences are padded to the longest molecule length in the batch.

3.2 Architecture

The architecture[1] is pictured in Figure 1 for an exemplary molecule. The atom sequence is processed by a trainable embedding layer. The adjacency matrix is treated as a sequence of matrix columns and embedded by a linear layer. The resulting two sequences are stacked, and a positional encoding is added, which helps the transformer attending to specific positions in the sequence. The resulting sequence forms the input for the transformer, which consists of multiple stacked encoder blocks and implements the multi-head self-attention mechanism. The encoded sequences are used to predict the target atom and bond logits. However, this process differs between training and inference. In both cases, the probabilities for the next atom type are computed by a fully connected decoder network $\mathrm{FC}_{\mathrm{ATOM}}$. The next column of the adjacency matrix is predicted by combining the encoded sequence with information about the next atom. This way, the model can take a sampled atom's type into consideration when predicting its bonds. During training, this is achieved by passing the ground truth sequence of target atoms to the atom embedding layer and stacking it with the encoded sequence. During sampling, the predicted atom probabilities are used to predict an appropriate next atom type, which is then passed to the atom embedding and stacked with the encoded sequence. In both cases, the atom type enriched sequence is passed to a fully connected decoder network $\mathrm{FC}_{\mathrm{BOND}}$ to predict the next column of the adjacency matrix. Like Lioa et al. [2] proposed, the network not only predicts a single categorical distribution over the different bond types for every value of the column, but rather a mixture of multiple cat-

[1]The source code can be accessed via https://gitlab.uni-oldenburg.de/gies6280/molegent

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

Model type	Valid	Unique@1k	Unique@10k	IntDiv	IntDiv2	Filters	Novelty
CharRNN	0.9700	**1.0000**	0.9994	0.8562	0.8503	0.9943	0.8419
AAE	0.9400	**1.0000**	0.9973	0.8557	0.8499	0.9960	0.7931
VAE	0.9800	**1.0000**	0.9984	0.8558	0.8498	**0.9970**	0.6949
JTN-VAE	**1.0000**	**1.0000**	**0.9996**	0.8551	0.8493	0.9760	0.9143
LatentGAN	0.9000	**1.0000**	0.9968	0.8565	0.8505	0.9735	**0.9498**
Our Model (fixed)	0.9893	**1.0000**	0.9989	**0.8569**	**0.8510**	0.9943	0.6312
Our Model (depth-first)	0.9751	0.9998	0.9994	0.8568	0.8509	0.9896	0.7931

Table 1: Comparison of molecules created by different generative models

egorical distributions. This enables the model to capture correlations occurring within one column of the adjacency matrix. Therefore, outputs produced by the bond decoder have an additional dimension of size K, the user defined number of mixture components. A single layer decoder FC_α produces the K-dimensional vector of probabilities for these components.

3.3 Training and Sampling

During training, the input sequence is processed in parallel. An autoregressive attention mask is used to prevent the model from attending to subsequent entries of the sequence. The model parameters are optimized by gradient descent using the Adam optimizer. The loss is defined by the cross entropy between the predicted atom and bond logits and the respective next atom type in the atom sequence and the next adjacency matrix column.

Molecules are sampled in an autoregressive manner. The process is initiated by passing a *Begin of Molecule* token and an empty adjacency matrix column to the model. After every pass, the predicted probabilities are used to sample the next atom type and adjacency matrix column. The input sequence is then appended with the newly sampled values and passed to the model for the next sampling step. This process is repeated until a maximum number of atoms has been sampled. After sampling, the generated atom sequences are trimmed until the first occurrence of an *End of Molecule* token. The atoms and the adjacency matrix are convert to the desired output format with the help of Rdkit.

4 Experiments

To evaluate the quality of the generative model we utilize the evaluation tools provided by MOSES. This allows a comparison to other generative models for molecules, like the CharRNN, AAE, VAE, JTN-VAE and LatentGan model provided by MOSES [4]. Our model is trained in two experimental conditions, differing in the order of atoms in the training molecules. In the fixed ordering condition, this order is the same as defined by the data set. However, previous work has shown that graph-based generative models can be applied to a variety of node orderings [1, 2]. Some orderings may be more difficult to learn, but utilizing different node orderings per molecule could lead to a more diverse and robust generative model. Therefore, in a second condition, for every molecule

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

	FCD (↓)		SNN (↑)		Frag (↑)		Scaf (↑)	
Model type	Test	TestSF	Test	TestSF	Test	TestSF	Test	TestSF
CharRNN	0.0732	**0.5204**	0.6015	0.5649	**0.9998**	0.9983	0.9242	**0.1101**
AAE	0.5555	1.0572	0.6081	0.5677	0.9910	0.9905	0.9022	0.0789
VAE	0.0990	0.5670	0.6257	0.5783	0.9994	**0.9984**	0.9386	0.0588
JTN-VAE	0.3954	0.9382	0.5477	0.5194	0.9965	0.9947	0.8964	0.1009
LatentGAN	0.2968	0.8281	0.5371	0.5132	0.9986	0.9972	0.8867	0.1072
Our Model (fixed)	**0.0639**	0.5495	**0.6355**	**0.5841**	0.9997	0.9979	**0.9409**	0.0564
Our Model (depth-first)	0.0783	0.5319	0.6148	0.5724	0.9998	0.9981	0.9334	0.0914

Table 2: Similarities between generated molecules and the test/scaffold test set

that is drawn as a training sample a random starting atom is chosen, and the remaining atoms are sorted by traversing the graph in a depth-first manner.

For evaluation, both models are used to sample the recommended amount of 30,000 molecules each, which are passed to MOSES for analysis. This procedure is repeated 10 times and the mean results are pictured in Table 1. Both models generated mostly valid molecules. The model trained on fixed atom orderings is only surpassed by the JTN-VAE, which is only capable of generating valid molecules by design. The fraction of unique molecules in a random subset of 1,000 and 10,000 molecules is comparable to those of the other models. Two internal diversity metrics are given by MOSES estimating the diversity within the generated molecules and therefore are an indicator on how well the model covers the chemical search space. Both of our models slightly surpassed the other models in these metrics. A high fraction of the generated molecules passes chemical filters (e.g. MCF, PAINS). The model trained on a fixed ordering generated a lower number of novel molecules when compared to the other approaches. As expected, the model trained on different depth-first orderings generates a clearly higher fraction of novel molecules while still generating a high amount of valid molecules. However, some other models still show a substantially higher fractions of novel molecules. It is conceivable that the general necessity of a fixed node ordering limits the model's capability of generating more novel molecules. All in all, in the presented experiments the proposed approach was able to compete with other state-of-the-art molecule generation models and is able to generate a high fraction of valid and diverse molecules.

Furthermore, MOSES features four similarity measure to determine how closely the generated molecules resemble the test sets. The statistics for the distance measures are presented in Table 2. The Fréchet ChemNet Distance (FCD) uses ChemNet and compares the different distributions in the activation of its last layer. The Nearest neighbor similarity (SNN) is defined by the mean similarity of all molecules to their nearest neighbor. Fragment similarity (Frag) and Scaffold similarity (Scaf) define cosine similarities between fragments and scaffold frequencies between the sets. Comparing these similarities, both models generalized well and generate molecules similar to those in the test sets. All in all, the similarity scores are close to those of the other models. Furthermore, the fixed ordering model generated molecules with a better FCD, SNN and Scaf similarity to the test set and a better SNN similarity to the scaffold test set.

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

5 Conclusion

This work introduces a transformer-based generative model for graphs that directly utilizes the multi-head self-attention mechanism to predict distributions over nodes and edges. In experiments on the generation of molecular graphs, the model was able to generate a high amount of valid molecules. Different distance metrics suggest that the model generalized well to unseen molecules, and the model is on par with other state-of-the-art molecule generation models. Due to the parallelizability of the transformer architecture and no necessity of message passing, the framework could scale very well with bigger problems.

References

[1] Yujia Li, Oriol Vinyals, Chris Dyer, Razvan Pascanu, and Peter W Battaglia. Learning Deep Generative Models of Graphs. *CoRR*, abs/1803.0, 2018.

[2] Renjie Liao, Yujia Li, Yang Song, Shenlong Wang, Will Hamilton, David K Duvenaud, Raquel Urtasun, and Richard Zemel. Efficient Graph Generation with Graph Recurrent Attention Networks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[3] Jiaxuan You, Rex Ying, Xiang Ren, William L. Hamilton, and Jure Leskovec. GraphRNN: Generating realistic graphs with deep auto-regressive models. In *35th International Conference on Machine Learning, ICML 2018*, volume 13, pages 9072–9081, feb 2018.

[4] Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, Artur Kadurin, Simon Johansson, Hongming Chen, Sergey Nikolenko, Alán Aspuru-Guzik, and Alex Zhavoronkov. Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models. *Frontiers in Pharmacology*, 11, dec 2020.

[5] Marwin H.S. Segler, Thierry Kogej, Christian Tyrchan, and Mark P. Waller. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Central Science*, 4(1):120–131, 2018.

[6] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Central Science*, 4(2), 2018.

[7] Marco Podda, Davide Bacciu, and Alessio Micheli. A deep generative model for fragment-based molecule generation. In *International Conference on Artificial Intelligence and Statistics*, pages 2240–2250. PMLR, 2020.

[8] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *International conference on machine learning*, pages 2323–2332. PMLR, 2018.

[9] Deng Cai and Wai Lam. Graph Transformer for Graph-to-Sequence Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7464–7471, apr 2020.

[10] Benson Chen, Regina Barzilay, and Tommi S Jaakkola. Path-Augmented Graph Transformer Network. *CoRR*, abs/1905.1, 2019.

[11] Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying WEI, Wenbing Huang, and Junzhou Huang. Self-Supervised Graph Transformer on Large-Scale Molecular Data. *Advances in Neural Information Processing Systems*, 33:12559–12571, 2020.

[12] Sanghyun Yoo, Young-Seok Kim, Kang Hyun Lee, Kuhwan Jeong, Junhwi Choi, Hoshik Lee, and Young Sang Choi. Graph-Aware Transformer: Is Attention All Graphs Need? *CoRR*, abs/2006.05213, 2020.