IMAGE-TO-GRAPH TRANSFORMERS FOR CHEMICAL STRUCTURE RECOGNITION

Sanghyun Yoo *Ohyun Kwon* *Hoshik Lee*

Samsung Advanced Institute of Technology
{sam.yoo, o.kwon, hoshik.lee}@samsung.com

ABSTRACT

For several decades, chemical knowledge has been published in written text, and there have been many attempts to make it accessible, for example, by transforming such natural language text to a structured format. Although the discovered chemical itself commonly represented in an image is the most important part, the correct recognition of the molecular structure from the image in literature still remains a hard problem since they are often abbreviated to reduce the complexity and drawn in many different styles. In this paper, we present a deep learning model to extract molecular structures from images. The proposed model is designed to transform the molecular image directly into the corresponding graph, which makes it capable of handling non-atomic symbols for abbreviations. Also, by end-to-end learning approach it can fully utilize many open image-molecule pair data from various sources, and hence it is more robust to image style variation than other tools. The experimental results show that the proposed model outperforms the existing models with 17.1 % and 12.8 % relative improvement for well-known benchmark datasets and large molecular images that we collected from literature, respectively.

Index Terms— Optical chemical structure recognition, molecular graph, deep learning, Transformer

1. INTRODUCTION

Newly discovered chemicals are published in scientific papers such as journals and patents. Their properties or synthesis methods are described in text, which can be extracted by natural language processing and text data mining techniques. However, the chemicals themselves are mainly represented as images of graphs, in which a node and an edge indicates an atom and a bond, respectively. Such chemical graphs are often drawn to have a non-atomic symbol that represents a group of atoms or even a list of groups of atoms to reduce the complexity.

Although a tool for automatic extraction of the molecular graph structures from images would be useful for many applications, only a few ones are available. Most of them are based on the hand-crafted rules and hard to cope with many different drawing styles. Also, they mainly generate a string-based representations such as SMILES (Simplified Molecular-Input Line-Entry System) [1]. A problem of SMILES is that it is not robust to small changes, which can result in the generation of invalid or highly different structures [2]. Another problem is that its syntax does not permit non-atomic symbols for abbreviations.

One of the alternatives to avoid the problems of SMILES is to generate the graph structure directly. Since it is a natural representation of molecules, the transformation of the image to the graph is more intuitive than to SMILES. Also, it is easy to extend the expressiveness so that the output molecular graph can include arbitrary symbols. However, there have not been many studies to generate a graph from an image.

In this paper, we propose an end-to-end deep learning approach to recognize molecular images and generate the corresponding graphs. Our paper's main contributions are as follows: First, We propose an image encoder that consists of the ResNet and the Transformer layers. Unlike the ResNet which mainly focuses on the locality, the Transformer can capture the contextual information from a broad span in the image, and such ability enables the model to correctly recognize the bond between two atoms located far from each other in the image. Second, we design a decoder that can generate the graph structures directly from the encoded image. This decoder based on the Transformer modified to reflect the edge information outputs nodes and edges in an auto-regressive manner. Finally, we propose several training techniques suitable to our own encoder-decoder architecture. We show that our approach significantly improve the model performance.

2. RELATED WORK

Approaches of the chemical image recognition can be classified in two categories: the rule-based and the deep-learning-based [3].

In the past, most tools for molecular graph recognition are rule-based, which are derived from optical character recognition. Although they are widely used since most of them are open to public [4, 5, 6], their heuristic rules often fail to cover diverse styles of images.

Due to the advances in deep learning, there recently have been many attempts to train a model that recognizes chemical images and generates molecular structures. Most of them

consist of an encoder based on convolutional neural networks and a decoder that outputs SMILES [7, 8, 9, 10] strings. Although such architecture is an easy solution, it is limited extracting only chemically valid structure since molecular structures having non-atomic symbols cannot be described in SMILES. ChemGrapher [11] can generate a graph instead of SMILES, and hence it can be extended to handle non-atomic symbols. However, since it outputs the graph information per each image pixel, it needs images labeled pixelwise to train. This hinders the use of diverse images as the training data, although many sources of image-molecule pairs published by other organizations such as USPTO (United States Patent and Trademark Office) [12] are available.

3. PROPOSED MODEL

3.1. Image Encoder

The encoder extracts the information for generating graphs from images. The resolution of the input image of our model is fixed as 800 x 800 pixels. We virtually split the image into 25 pieces horizontally and vertically again so that there are 625 pieces each of which has 32 x 32 pixels. Then we assign the 2D position information for each piece, which is subsequently concatenated to the corresponding raw image features as in Figure 1 (a). The output features of the ResNet-34 are reshaped to 1D sequence as shown in Figure 1 (b). Then, they are fed into the Transformer encoder [13] so that the relation between two atoms far from each other in the image can be captured. The final output of the encoder will be used as the input of the graph decoder explained in Section 3.2.

We hope that all the atom and bond information contained in each piece of the 2D image is encoded into the corresponding step of the sequence so that our decoder can focus on small parts of the sequence when generates each node and the edges. To help the encoder gather the necessary information, we add classifiers on the Transformer encoder to predict the number of atoms, characters in the atom labels, and the IDs of pieces that share edges with for each step of the sequence, i.e., each piece in the image, as shown in Figure 1 (c). These auxiliary losses help train the encoder in the early stages and then gradually decrease after certain steps until they reach zeros.

3.2. Graph Decoder

Our decoder generates the final graph by referring to the embedded sequence created by the image encoder in an autoregressive manner; at step t, it first encodes the sub-graph generated at step t-1 and then generates a new node and its associated edges to the existing ones.

The decoder encodes the sub-graph by considering nodes as tokens of the Transformer decoder and handling edges with the self-attention, which is originally designed for handling the relation between tokens. Since the edge information (e.g., bond types) in the graph should be reflected additionally, we design our graph decoder to modulate the self-attention weights according to the edge information. To make our model learn the importance of each connection solely from the data, we use feature-wise transformation [14]. In this method, the original attention weights are transformed with the scaling factor γ and the biasing factor β, which are generated based on the edge type as the conditioning information. The final attention values, $Att(Q, K, V)$, are as follows:

$$(\gamma_{ij}, \beta_{ij}) = f(e_{ij}) \tag{1}$$

$$Att(Q, K, V) = softmax\left(\frac{\Gamma \odot (QK^T) + B}{\sqrt{d_k}}\right) V \tag{2}$$

where e_{ij} indicates the edge type (in one-hot representation) between nodes i and j, and f is multi-layer perceptrons (MLPs). Γ and B are the matrices whose elements at (i, j) position are γ_{ij} and β_{ij}, respectively. The operator $\odot$ indicates the element-wise multiplication.

After the sub-graph is encoded, the graph decoder takes a special token and makes the embedding of a new node by referring to the embeddings of the sub-graph. Then, the new node and its associated edges are generated from the embedding as in Figure 1 (d). We design a training algorithm doing such two-path decoding in parallel, explained in detail in [15]. In addition to atom and bond labels, we force our graph decoder to generate coordinates in the source image. This modification helps the decoder be aware of atoms decoded so far and the atom it needs to focus on at the current step.

3.3. Training Data

To train our model, we extract about 4.6 M molecules from PubChem databases and generate the corresponding images using RDKit [16]. As in [11], we modify RDKit to get the exact coordinates and characters of atom labels represented in the image, and they are used to train the encoder and the decoder. Also, to recognize a variety of superatoms, we randomly replace carbon atoms with common superatoms, e.g., CF_3, NO_2, t-Bu, and so on.

Although RDKit can generate various styles of images, it is not enough to cover such diverse styles of images appeared in the literature. Therefore, we additionally use 2.5 M image-molecule pairs that USPTO published. Note that the pixel-wise classification model cannot exploit these data, since there is no exact coordinate of atoms on the image. However, our end-to-end learning approach can fully utilize such data by just ignoring the losses related to the coordinates.

4. EXPERIMENTS

4.1. Benchmark Images

To test each model, we used four well-known benchmark images: UoB, USPTO[1], CLEF, and JPO. Detailed explanation

[1]We made sure that images in training and test set were not overlapped.

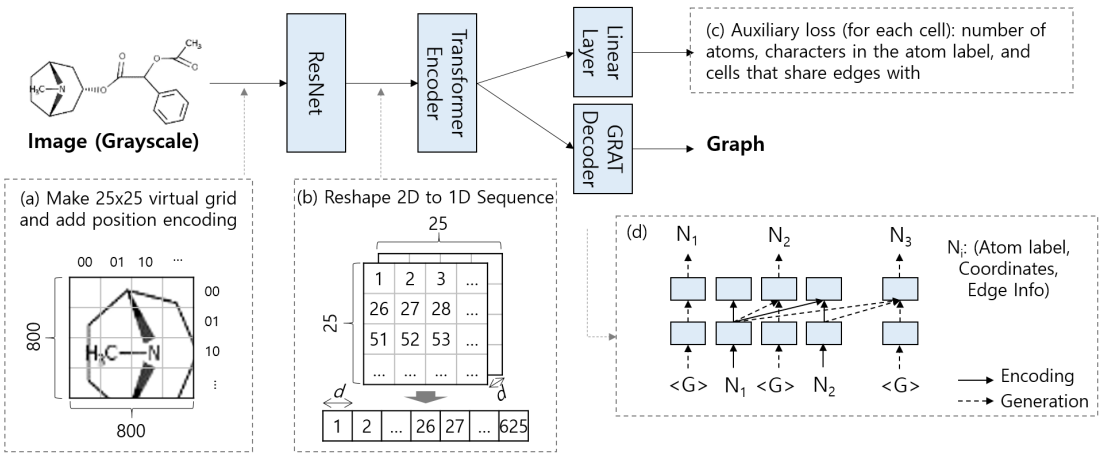


Fig. 1: Encoder-decoder architecture of our model

of each set can be found in [3]. To avoid a potential overfitting of existing tools to these well-known benchmark datasets [9], we converted these benchmark images to PDF files and back to image files again. We insist such conversion is not too much artificial, since the recognition tools are designed to extract information from the literature which is typically published as in the PDF format. The sizes of molecules in the benchmark sets are small; average number of atoms per a molecule is around 15.8. To compare the recognition performance on the image of the large molecules, we collected 12 journal papers about organic light-emitting diodes, manually segmented 434 images of molecular structures, and generated the ground-truth molecular structures. The average number of atoms in a molecule in this dataset, called OLED, is 52.8.

4.2. Effect of Training Techniques

We conducted several experiments to see the training techniques proposed in Section 3. To see the trends quickly, we trained all models with a small portion of training data and tested on UoB benchmark images only. Although our model generates the molecular graph structure, we use the SMILES match measure (SMI) since there is no simple measure to indicate whether two graphs are exactly the same. With this measure, it is correct if the model predicts the exact same canonical SMILES as the answer. TS 1 means the ratio of the prediction whose Tanimoto similarity[2] with the answer is 1.0, and Sim. means the average Tanimoto similarity.

Table 1 shows the results. First we compared two base models which consist of the ResNet only. They cannot recognize images correctly regardless of coordinates prediction. When we simply put Transformer layers on the ResNet layers, the model could not be trained properly either. However, using the auxiliary loss and forcing the decoder to predict coordinates significantly boosts the accuarcy of the model.

[2]The most popular similarity measures for comparing chemical structures. Two structures are usually considered similar if the value is over 0.85.

Table 1: The effect of proposed training techniques. (O and X means 'used' and 'not used', respectively.)

Layers	Aux. Loss	Coord Pred	SMI	TS 1	Sim.
ResNet only	-	X	.000	.000	.120
ResNet only	-	O	.046	.056	.276
ResNet + Trans	X	X	.000	.000	.000
ResNet + Trans	O	X	.009	.010	.176
ResNet + Trans	O	O	**.567**	**.615**	**.759**

4.3. Overall Performance

Table 2 shows the comparison of the performance among OSRA [4], MolVec [6], ChemGrapher [11], and ours. Since we could not find the source code of ChemGrapher, we rewrite the accuracy in their paper [11], which was retrieved from the test on the original (i.e., not degraded) images. We trained two models: one was trained with generated images using RDKit only, and another one was trained with open image-molecule pairs, i.e., USPTO, as well as generated images.

Our models generally outperformed the existing models. The total accuracy over four test cases shows that our model outperformed OSRA and Molvec by 82.3 % and 17.7 %, respectively. Moreover, our model trained with additional USPTO data showed better performance than the one with generated data only. Our model was improved by 20.5 % by the utilization of the large open dataset, which was possible due to our proposed image-to-graph, end-to-end learning architecture.

Then, we tested three models on new benchmark dataset, OLED, to see their performance on the large molecules. Our model outperformed OSRA and Molvec by 20.0% and 12.8 % respectively as shown in Table 2.

Table 2: Performance comparison among four models tested on well-known benchmark datasets.

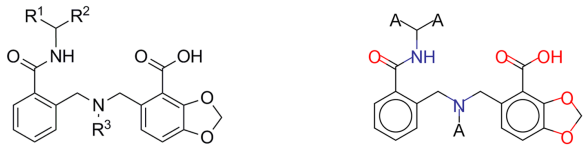
	UOB			USPTO			CLEF			JPO			Total		
Model	SMI	TS 1	Sim.	SMI	TS 1	Sim.	SMI	TS 1	Sim.	SMI	TS 1	Sim.	SMI	TS 1	Sim.
OSRA v2.1.1	.776	.817	.896	.006	.006	.189	.056	.062	.357	.417	.423	.596	.368	.387	.531
MolVec v0.9.8	.783	.827	.905	.398	.422	.625	.400	.408	.737	.488	.506	**.758**	.570	.600	.762
ChemGrapher	.706	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Ours	.720	.747	.851	.449	.555	**.799**	.378	.568	.821	.243	.356	.640	.557	.635	.818
Ours + USPTO	**.829**	**.865**	**.919**	**.551**	**.661**	.792	**.517**	**.749**	**.851**	**.503**	**.572**	.748	**.671**	**.755**	**.852**

Table 3: Performance comparison over images extracted from actual journal papers.

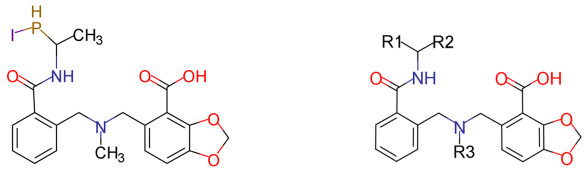
Model	SMI	TS 1	Sim.
OSRA v2.1.1	.661	.666	.803
Molvec v0.9.8	.703	.707	.836
Our model	.747	.760	.852
Our model + USPTO	**.793**	**.795**	**.866**

4.4. Further Analysis

To show the advantage of our graph-based model, we made three models recognize the image having non-atomic symbols. Figure 2 shows one of the source images of the molecule including pseudoatom R's and the recognition result of each model. Since OSRA can only generate SMILES that does not support non-atomic symbols, it recognized R1, R2, and R3 as A (any-atom); it is not possible to distinguish them. Molvec could not recognize them properly at all. However, our model recognized R1, R2, and R3 exactly, and hence it can substitute them with the intended sub-structures by post-processing.



(a) Source (b) OSRA v2.1.1

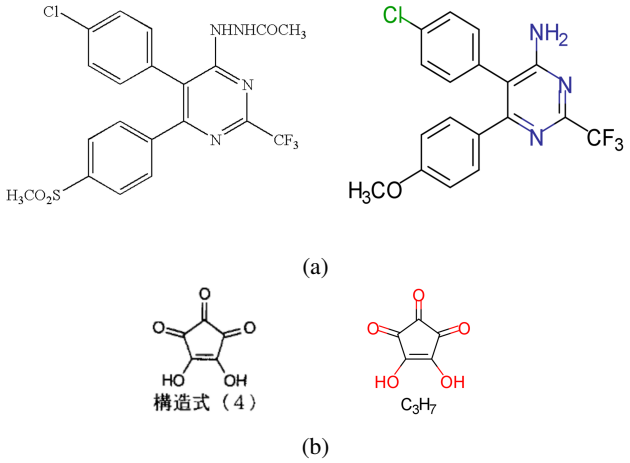


(c) Molvec v0.9.8 (d) Our model

Fig. 2: (a) The source image of the molecule including pseudo-atom R's and recognition results of (b) OSRA, (c) Molvec, and (d) our model.

We analyzed the recognition results of USPTO, CLEF, and JPO, since our model showed poor performance on these datasets. One of the main reasons is that there were many superatoms which our model could not handle with, e.g., $NHNHCOCH_3$ or H_3CO_2S in Figure 3 (a). Although generating more superatoms in training data can be one of solutions, the ultimate solution is to recognize symbols by character-level. We plan to modify the atom classifier to character-level decoder as future work.

The other reason is that, as in Figure 3 (b), our model tried to recognize captions as atoms. This issue was worse in JPO cases than others. Such behaviors might be reduced by augmenting training images to contain arbitrary captions so that our model could ignore them, or by designing an image segmentation model.



(a)

(b)

Fig. 3: Examples of images that our model fails to recognize: source images (left) and the recognized results (right).

5. CONCLUSION

In this paper, a deep learning approach to recognize the molecular images and generate the graph structures directly even when there are non-atomic symbols for abbreviations was suggested. We also provided novel training techniques that significantly boost the accuracy of the model. Through extensive experiments over several benchmark datasets, we showed that the proposed model outperformed existing open source libraries. As future work, we plan to extend the model to generate atom labels at character-level, so that it can recognize any arbitrary symbols.

6. REFERENCES

[1] Daylight Chemical Information Systems Inc., "Smiles - a simplified chemical language," https://www.daylight.com/dayhtml/doc/theory/theory.smiles.html.

[2] Nicola De Cao and Thomas Kipf, "Molgan: An implicit generative model for small molecular graphs," *CoRR*, vol. abs/1805.11973, 2018.

[3] Kohulan Rajan, Henning Otto Brinkhaus, Achim Zielesny, and Christoph Steinbeck, "A review of optical chemical structure recognition tools," *Journal of Cheminformatics*, vol. 12:60, 2020.

[4] Igor V. Filippov and Marc C. Nicklaus, "Optical structure recognition software to recover chemical information: Osra, an open source solution," *Journal of Chemical Information and Modeling*, vol. 49, pp. 740–743, 2009.

[5] Viktor Smolov, Fedor Zentsev, and Mikhail Rybalkin, "Imago: open-source toolkit for 2d chemical structure image recognition," in *The Twientieth Text REtrieval Conference (TREC 2011) Proceedings*, 2011.

[6] "Molvec," https://github.com/ncats/molvec, 2019.

[7] Joshua Staker, Kyle Marshall, Robert Abel, and Carolyn M. McQuaw, "Molecular structure extraction from documents using deep learning," *Journal of Chemical Information and Modeling*, vol. 59, pp. 1017–1029, 2019.

[8] Kohulan Rajan, Achim Zielesny, and Christoph Steinbeck, "Decimer: towards deep learning for chemical image recognition," *Journal of Cheminformatics*, vol. 12:65, 2020.

[9] Djork-Arné Clevert, Tuan Le, Robin Winter, and Floriane Montanari, "Img2mol - accurate smiles recognition from molecular graphical depictions," *ChemRxiv*, 2021.

[10] Hayley Weir, Keiran Thompson, Amelia Woodward, Benjamin Choi, Augustin Braun, and Todd J. Martínez, "Chempix: automated recognition of hand-drawn hydrocarbon structures using deep learning," *Chemical Science*, vol. 12, pp. 10622–10633, 2021.

[11] Martijn Oldenhof, Adam Arany, Yves Moreau, and Jaak Simm, "Chemgrapher: Optical graph recognition of chemical compounds by deep learning," *Journal of Chemical Information and Modeling*, vol. 60, pp. 4506–4517, 2020.

[12] "United states patent and trademark office," http://uspto.gov.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 31*, 2017.

[14] Vincent Dumoulin, Ethan Perez, Nathan Schucher, Florian Strub, Harm de Vries, Aaron Courville, and Yoshua Bengio, "Feature-wise tranformations," *Distill*, 2018, https://distill.pub/2018/feature-wise-transformations.

[15] 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*, vol. abs/2006.05213, 2020.

[16] Greg Landrum, "Rdkit: Open-source cheminformatics," http://www.rdkit.org.