# Decomposing Retrosynthesis into Reactive Center Prediction and Molecule Generation

Xianggen Liu,[†,‡,¶] Pengyong Li,[†,‡,¶] and Sen Song[*,†,‡]

†*Laboratory for Brain and Intelligence and Department of Biomedical Engineering, Tsinghua University, Beijing, China*

‡*Beijing Innovation Center for Future Chip, Tsinghua University, Beijing, China*

¶*These authors contributed equally to this work.*

E-mail: songsen@mail.tsinghua.edu.cn

**Abstract**

Chemical retrosynthesis has been a crucial and challenging task in organic chemistry for several decades. In early years, retrosynthesis is accomplished by the disconnection approach which is labor-intensive and requires expert knowledge. Afterward, rule-based methods have dominated in retrosynthesis for years. In this study, we revisit the disconnection approach by leveraging deep learning (DL) to boost its performance and increase the explainability of DL. Concretely, we propose a novel graph-based deep-learning framework, named DeRetro, to predict the set of reactants for a target product by executing the process of disconnection and reactant generation orderly. Experimental results report that DeRetro achieves new state-of-the-art performance in predicting the reactants. In-depth analyses also demonstrate that even without the reaction type as input, DeRetro retains its retrosynthesis performance while other methods show a significant decrease, resulting in a large margin of 19% between DeRetro and previous state-of-the-art rule-based method. These results have established DeRetro as a power-

ful and useful computational tool in solving the challenging problem of retrosynthetic analysis.

# Introduction

Retrosynthesis aims to derive the suitable set of reactants, by which the given target molecule can be produced. It plays a key role in many applications such as drug discovery, material synthesis, environmental science. Computational retrosynthesis tools has been widely accepted as assistants in designing synthetic routes for novel molecules. Over the last few decades, a mount of researches for retrosynthesis have been proposed on the basis of the emerging computational and analytic techniques.

Retrosynthesis analysis (also known as disconnection approach) was firstly formalized by Corey and Wipke[1], sketching a processing workflow that the target molecule is recursively transformed into simpler precursor molecules until commercially or naturally available molecules are obtained.[2] It consists of two sub-tasks: 1) disconnection, how the given product is breaking down into destructural units, which is also called synthons; 2) planning, an optimal decision sequence of disconnection to recursively transform the target molecule into a set of synthons, each of which corresponds a readily available molecule. Based on the above classical disconnection approach, Corey designed the first computer-assisted organic synthesis (CAOS) system, Logic and Heuristics Applied to Synthetic Analysis (LHASA).[3]

Afterward, rule-based CAOS systems leverages manually or automatically extracted reaction rules as templates for chemical transformations that are applied to an input target molecule to derive the corresponding precursors. Rule-based methods has dominated for several decades[4–12] but are limited by the generalization of the extracted reaction rules. Due to the high dependence on rules, rule-based systems often struggle to predict retrosynthetic reactions for a novel target products that are beyond the scope of the knowledge base or the expert databases.

Recently, deep learning and reinforcement learning have been applied in retrosynthesis to increase the generalization as well as the prediction performance of rule-based methods.[9,12–14] Liu et al.[13] formulated retrosynthesis prediction as a translation task using a sequence-to-sequence (seq2seq) architecture, where molecules are encoded as SMILES[15] sequences. The advantage of the seq2seq model is end-to-end and is able to access global information instead of only the reaction center. But it stipulates a generating order of reactants for each reactions, which is counter-intuitive and sometimes misleading for the learning of a model. Segler et al.[12] developed a reinforcement framework where Monte Carlo tree search is combined with an policy network that guides the search, and a ranking network to pre-select the most promising rules. However, the value function, derived from the performance of final predicted reactants, is relative sparse and thus is difficult to guide the agent when the sampling is ineffective. Baylon et al.[16] built a deep highway network performing multiscale reaction classification to enhance the rule-based method. This method leveraged deep learning technique to select suitable rule candidates in a multi-scale fashion. However, it also has the risk of failing when training data is insufficient or the given target product is novel due to the limitation of rule (symbolic) matching.

To address some of these issues, inspired from the aforementioned disconnection approach, we decompose the retrosynthesis into two sub-tasks including reaction center prediction and molecule generation, and propose a novel framework, named DeRetro, which contains two novel graph-based neural networks to accomplish the above two sub-tasks respectively. The workflow of DeRetro is as follows: 1) identifying the reaction center by a graph-to-graph neural network; 2) automatically splitting the target product into several synthons; 3) generating the corresponding reactant SMILES step by step for each synthon. DeRetro adopts graph-based neural network to model the interactions of atoms in a molecule and thus is able to extract more meaningful and global features for the downstream tasks than the rule templates and other sequence representation.[15,17]

The proposed framework differs from the method named multiscale reaction classifica-

tion[16] that involves categorizing the target product into pre-clustered rule sets. DeRetro identifies the chemical bonds of the product to perform the reaction center prediction, which is more robust and is more easy to generalize. Besides the reaction center prediction, instead of the symbolic planning scheme,[1,12] DeRetro directly generates reactants in an end-to-end and differentiable fashion.

We evaluate the effectiveness of our model on a standard benchmark dataset[13,18] that contains about 50,000 reactions with labeled reaction types.[19] The experimental results showed that DeRetro is able to accurately predict reaction centers with only 1.2% error rate and thus is superior to rule-based expert system in a large scale. Experiments on retrosynthetic reaction prediction demonstrated that DeRetro can significantly outperform the current state-of-the-art methods including rule-based method and seq2seq model.[13] Moreover, in a more realistic setting where the reaction type is unavailable to obtain in advance, DeRetro retains its predictive power while other methods show a significant decrease, resulting in a large margin by up to 19% in terms of prediction accuracy between DeRetro and previous state-of-the-art rule-based method. These results have demonstrated that DeRetro can serve as a powerful and useful computational tool in solving the challenging problem of retrosynthetic analysis.

# Result

## The DeRetro framework

In this study, we propose a novel graph-based deep learning framework, named DeRetro, to predict the reactants by decomposing retrosynthesis into two sub-tasks, including reaction center prediction and molecule generation. More specifically, to boost the performance as well as increase the explainability of the deep learning model, the main workflow for the DeRetro framework can be divided into two deep neural networks, namely the graph-to-graph (Graph2Graph) network and the calibrated graph-to-sequence (CGraph2seq) network

(Figure 1):

- The graph-to-graph (Graph2Graph) network takes the product and the reaction types as input, extracts meaningful graphical features, and predicts the reaction center. Specifically, the reaction center is composed of the bonds that are changed during the reaction. Based on the predicted reaction center, the broken units named synthons can be automatically split from the product.

- The calibrated graph-to-sequence (CGraph2seq) network aims to generate the SMILES sequence of the corresponding reactant for each synthon. To make up the information loss caused by decomposing, the input of CGraph2seq also includes the corresponding original product and the reaction type, in addition to the synthon.

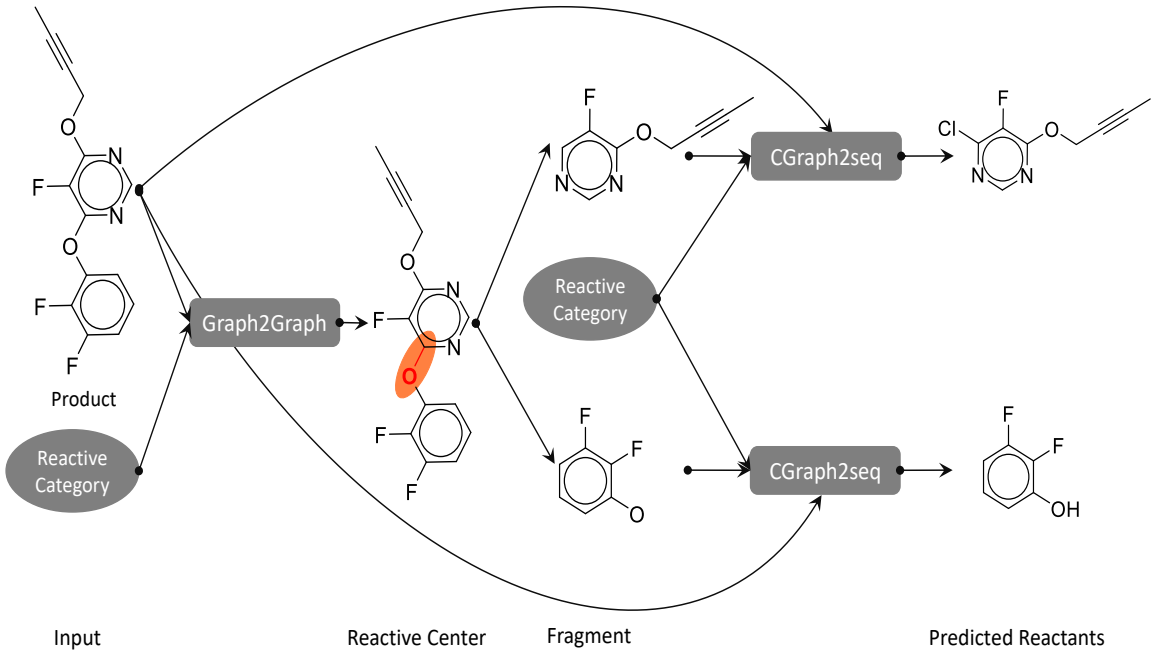


Figure 1: The overview of the proposed approach.

The two components of DeRetro are trained independently since the supervision information of each step is available from atom-mapped reactions. Therefore, the step-by-step

supervision makes the learning of our two modules in DeRetro more effective and straightforward than other end-to-end methods such as reinforcement learning framework [12,20] and sequence-to-sequence model. On the one hand, our neural-based framework make it easy to extend to the reactions outside of the training data, which is difficult for rule-based approach. On the other hand, the decomposing protocol eliminates the side effect caused by the order of generating reactants in Seq2seq model (there is only one correct generating order in Seq2seq model but actually the generating order does not matter). More details of the DeRetro and the corresponding training process can be found in the Methods Section.

Note that in contrast to the other deep-learning approaches (e.g., Seq2seq method) working in a blackbox (difficult to rationalize the generation process), DeRetro explicitly presents the reaction centers and the rationales of the generating process. Besides the overall prediction accuracy, the interpretation of data-driven approach is also important in providing rationales of the output and new insights for the chemists. Therefore, the processing logic of DeRetro would make more sense.

## DeRetro accurately predicts reaction centers

To evaluate the performance of DeRetro, we trained and tested our two modules using a standard retrosynthesis dataset,[18] which contains ∼50,000 atom-mapped reactions. The dataset was divided into three parts for training, validation, and testing (8:1:1). For a particular reaction sample, we decomposed it into two parts: the first part stores the mapping from the product to the reaction center, the second part contains the mapping from the synthon (split by the reaction center from the product, as shown in Figure 1) to the corresponding reactant. After collecting each part from all the data, we built two datasets for our two modules respectively.

We first quantitatively tested the performance of the Graph2graph (the first computational module in DeRetro) in predicting the reaction centers. We compared our method with the typical rule-based method [10] that automatically extracts reaction rules from the training

data set and applies them to a target molecule to obtain the reaction centers and reactants. We implemented the rule-based method according to Law et al.'s algorithms[21] and Coley et al.'s implementation.[10] Jin et al.[22] also developed a template-free approach to pinpoint the reaction center, but it is applied to predict organic reaction outcomes, which is not directly comparable in the retrosynthesis task ( where the input is the reaction product instead of reactants). Besides the metric of accurate rate (top-1 accuracy), we also included the top-$N$ accuracy by matching the top-$N$ candidate solutions with the ground truth, to further learn about the predictive power of rule-based method since it accurate rate is relative low.

Figure 2(a) shows the accurate rate of DeRetro and the top-$N$ accuracy of the rule-base method in predicting the reaction centers. With a sophisticated design of architecture, our Graph2graph module achieved a 98.8% in accurate rate, outperforming the rule-based method by a large margin of 29%. In addition, we also showed that the performance of the Graph2graph is further superior to the top-10 accuracy.

The high accuracy of Graph2graph probably benefits from the input of reaction type, an important indicator that sketches the reaction process. However, the reaction type of a given new product might be agnostic in the realistic retrosynthesis. Thus, we removed the reaction type from the input in both the baseline and our Graph2graph module and retrained them to further study their predictive power (Figure 2(b)). Expectedly, without reaction type as input, all the methods showed a lower prediction performance. Intriguingly, rule-based method decreases $\sim$28% in terms of top-1 accuracy, while Graph2graph only have a slight reduce ($\sim$2%), resulting a larger margin ($\sim$55%) between them. The overall experiments illustrated that Graph2graph is able to accurately predicts the reaction centers in the both scenarios (with and without reaction type).

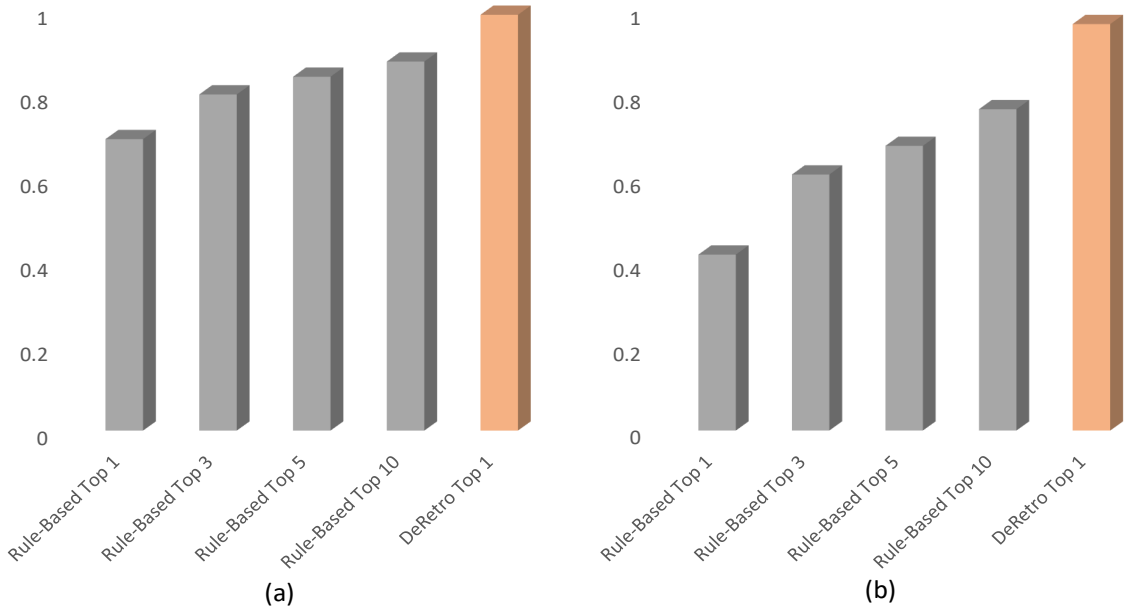


Figure 2: Comparison of performance in predicting reaction centers.

## DeRetro achieves the new state-of-the-art performance in retrosynthesis

Following the traditional evaluation protocol of the retrosynthesis, we examine the retrosynthesis performance of our model when it take both the product and reaction type as input. Besides the rule-based method, we also include deep learning-based model that works in a sequence-to-sequence (Seq2seq) fashion [13] as another baseline. It employs an encoder-decoder architecture that consists of two recurrent neural networks to perform the retrosynthesis. We trained it and evaluated its performance via using its published source codes.

Following the evaluation measurement used in Liu et al., [13] a prediction is correct if and only if all the reactants of a reaction are correctly predicted. In DeRetro and Seq2seq, beam search is used to produce $N$ candidate reactants for the measurement of top-$N$ accuracy. At each time step during the searching process, we keep the top $N$ (beam width) candidate output sequences ranked by overall log probability from the predicted distribution which is computed based on the last top $N^{(b)}$ candidate sequences.

Table 1 shows the comparison of top-$N$ accuracies between DeRetro and the baselines.

We found that Seq2seq obtained comparable performance with rule-based method, which is consistent with the previous report.[13] In addition, DeRetro provided the best prediction performance measured in terms of most of the metrics. We hypothesize that this is because the disentangled tasks make it easier for reactant generation.

Table 1: Top-$N$ accuracy (%) of the models on test set.

| **Model** | Top 1 | Top 3 | Top 5 | Top 10 | Top 20 | Top 50 |
|---|---|---|---|---|---|---|
| Rule-Based | 35.6 | 51.6 | 57.8 | **63.4** | **67.8** | 69.2 |
| Seq2seq | 37.7 | 50.4 | 54.3 | 58.6 | 62.1 | 66.0 |
| DeRetro | **41.1** | **55.5** | **59.1** | **63.4** | 66.4 | **69.6** |

## DeRetro retains the prediction performance when the reaction type is unknown

The aforementioned experiments shows DeRetro achieve the new state-of-the-art performance when both the product and reaction type act as input. However, the reaction type is usually unknown in the realistic drug retrosynthesis. In this section, we attempted to verify whether DeRetro could also obtain the superior performance when the reaction type is unavailable. We noticed in Table 2 that both the rule-based method and seq2seq showed a decrease in prediction performance, mainly due to the critical role of the removed reaction type information. Nevertheless, we found that without the reaction type as input, DeRetro still retained the predictive power of retrosynthesis, yielding superior prediction performance than the two baselines with a large margin. This is because 1) the reaction centers play the same role that describes the reaction process with the reaction type; and 2) the first step (Graph2Graph) of DeRetro obtained nearly 100% accuracy in predicting reaction centers (98.8% with type and 96.5% without type), no matter whether the reaction type is included or not.

Table 2: Top-$N$ accuracy (%) of the models on test set when the reaction type is removed from input.

| **Model** | Top 1 | Top 3 | Top 5 | Top 10 | Top 20 | Top 50 |
|---|---|---|---|---|---|---|
| Rule-Based | 19.7 | 33.7 | 39.7 | 50.2 | 61.2 | 67.8 |
| Seq2seq | 29.4 | 40.5 | 44.1 | 48.3 | 52.7 | 56.8 |
| DeRetro | **38.8** | **54.1** | **58.8** | **63.7** | **67.0** | **70.4** |

## DeRetro provides step-wise rationales in molecule generation

As the previous experiment showed Graph2Graph model in step one obtained nearly 100% accuracy of predicting reaction centers, providing a high interpretation in the reaction retrosynthesis, then we are curious about the transparency of the CGraph2seq in step two. Therefore, we visualized the attention weights at each step of the molecule generation in Figure 3. To be specific, we randomly selected several generation processes, recorded the generated characters and visualized both attentions to the corresponding synthon and the product. The SMILES are generated letter by letter, for example, chlorine atom (Cl) will be generated by 'C' and 'l' by steps, we regard these two steps' attention as chlorine atom's attention. To provide a clear and intuitive illustration, we only presented the step that generated a new atom and omitted the step of generating other characters (e.g., left or right bracket). we only plot the atom whose attention weight is larger than 0.15, with the color indicating the value, and the attention weight that is less than 0.15 is also ignored. Figure 3 illustrates one of the generation processes (See Supporting Information for more), which shows the CGraph2seq possesses the following properties: 1) when CGraph2seq generates an atom that already exists in the corresponding synthon, the step-wise attention to the synthon aligned well with the generating position 2) when CGraph2seq generates an atom that is outside the corresponding synthon, the step-wise attention to the synthon focuses on the reaction center. 3) the step-wise attention to the corresponding product just considers some active functional groups or other relevant positions since it plays a less important role than the synthon in molecule generation.

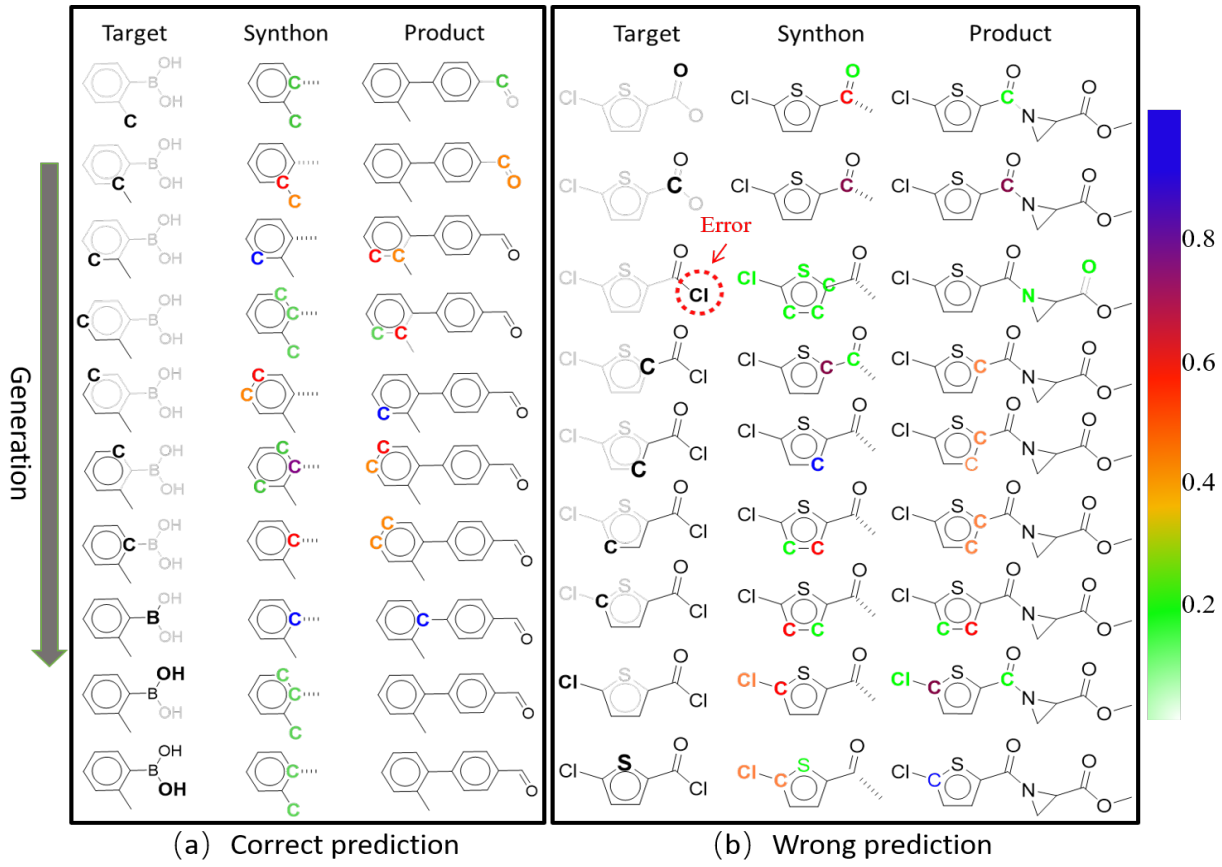


Figure 3: Visualization of atom attention among molecule generation process. The black bold atoms of target molecule represent the generated atoms at current step while the molecule structure with light color denotes the desired target (one of the reactants). The color of atoms in synthon and product represent the attention value. The dashed circle of (b) denotes the generated error atom.

# Discussion

**Molecular Representations.** Molecular fingerprint is a widely used way of encoding the structure of a molecule. The most common type of fingerprint is a series of binary digits (bits) that represent the presence or absence of particular substructures in the molecule. Although molecular fingerprint features in its flexibility and ease of computation for reaction prediction,[9] it also gives several issues including bit collisions and vector sparsity. Besides, molecules can be encoded as simplified molecular input line Entry System (SMILES)[15] in the format of single line text. Thus reaction prediction could be modeled as a sequence-to-sequence mapping task.[13] Nonetheless, a key weakness with representing molecules using text sequences is the fragility of the representation, since small changes in the text sequence can result in a large change in the molecular structure.

Compared to SMILES and other text notation, a hydrogen-depleted molecular graph is more suitable and natural to represent the structure information of molecules, whose nodes and edges corresponds to the atoms and chemical bonds respectively.[23] Recently, graph neural networks (GNNs)[24–26] are adopted to deal with graph data and has shown extraordinary performance on the applications of quantitative structure activity relationships (QSAR) prediction,[23,27] quantum chemistry,[28] and molecules generation.[29,30] These works reveal that GNNs are good at extracting the feature from molecule graph.

**Comparison between DeRetro and the previous methods.** The main characteristics of our approach are the decomposing strategy and the two novel and sophisticated graph-based neural networks, compared to the rule-based methods, deep learning enhanced rule-based expert system, and other end-to-end deep learning models.

The rule-based methods focus only on the local molecular environment because the automatically extracted reaction rules only consider neighboring atoms around the reaction centers. This results in an important issue with rule-based expert systems when performing the retrosynthetic reaction prediction task. The partial input may lead poor performance in reaction prediction. On the contrary, DeRetro takes the molecule graph structure as input

and learns task-oriented patterns end-to-end. The deep learning enhanced rule-based methods that combine a rule-based expert system with a neural network for candidate ranking eliminate the above issue but they are also limited by the generalization of the extracted rule set. The DeRetro models the reactant prediction as a generation task, which is more robust when a novel product molecule is given.

As discussed in Introduction section, Liu et al. [13] formulated retrosynthesis prediction as a translation task using a seq2seq architecture, as the first end-to-end deep learning model for retrosynthesis. However, it stipulates a generating order of reactants for each reactions, which is counter-intuitive and sometimes misleading for the learning of a model. The DeRetro does not have the above problem and the decomposing strategy further makes the reactant generation shorter and thus easier to learn.

# Conclusion

Retrosynthetic reaction prediction has long been regarded as an important computational chemistry problem. Inspired by Corey's disconnection approach, we propose a deep learning based framework, named DeRetro, which decomposes the retrosynthesis into reaction center prediction and molecules generation. Experimental results shows that DeRetro outperform the rule-based expert system and Seq2Seq model in terms of prediction accuracy, and retains the performance even without the reaction type as input. Besides the high prediction accuracy, DeRetro explicitly presents the reaction centres and the rationales of the generating process, providing the interpretation of the prediction and valuable information for the chemists. These results have established DeRetro as a powerful and useful computational tool in both reaction center prediction and Retrosynthetic reaction prediction.

# Methods

In this section, we firstly introduce the dataset we used in our experiments, then we elaborate two components of our method in details .

## Retrosynthesis Dataset

The retrosynthesis dataset was filtered from the USPTO database, containing ∼50,000 atom-mapped reactions,[18] and each reaction was annotated with one of the ten reaction types (Supplementary Table 1). Following the protocol of previous research,[13] trivial products such as inorganic ions and reagents were removed, and the multiple product reactions were split into multiple single product reactions, resulting into only one product and several corresponding organic reactants in each reaction sample. Additionally, we further eliminated the reactions that lacks atom-mapping. The data was divided into three parts for training, validation, and testing for the following experiments (8:1:1).

For a particular reaction sample of each data set, we split it into two parts: the first part storing the mapping from the product to the reaction center, the second part storing the mapping from the synthons to the corresponding reactants. The synthons are obtained by breaking down the product based on the reaction center, as shown in Figure 1). In this study, we omit the charge of synthon to simplify the representation. The two mappings make two data sets for our two models, respectively, with each dataset containing a training, validation, and test data subset according to the above data division. Table 2 shows the statistics of our two datasets. For the representation of molecules, we adopted the graphical structures (node features and their adjacent matrices) to represent products and synthons (which can be encoded efficiently by graph networks), and used SMILES sequences to represent reactants, which is a popular way for molecule generation.

Table 3: Statistics of the two processed datasets for the two sub-tasks respectively.

| **Sub-task** | Training | Validation | Testing |
|---|---|---|---|
| Reaction center prediction | 39650 | 4956 | 4956 |
| Molecule generation | 66097 | 8259 | 8249 |

## Predicting reaction center by the Graph2Graph network

A graph-to-graph (Graph2Graph) neural networks takes a reaction product and corresponding reaction type as input, and predicts the probability of being the reaction center for each chemical bond in the product. Concretely, the chemical bonds of the product are directly encoded as the corresponding binary adjacent matrix $\boldsymbol{A}^{(p)} \in \{0,1\}^{N\times N}$ (regardless of the types of bonds), where $N$ denotes the number of atoms in the product. We defines another adjacent matrix $\boldsymbol{A}^{(b)} \in \{0,1\}^{N\times N}$ as the broken adjacent matrix corresponds to the original binding information of the product atoms before the reaction. $\boldsymbol{A}^{(b)}$ describes the decomposing process whose entries will changes from 1 to 0 if the corresponding bonds break from the perspective of retrosynthesis. Then we could directly derived the reaction center by $\boldsymbol{A}^{(p)} - \boldsymbol{A}^{(b)}$. Therefore, the goal of Graph2Graph is to accurately predict the broken adjacent matrix $\boldsymbol{A}^{(b)}$ given the product features as input.

To provide an appropriate representation of the product, for each atom, we leverage a one-hot encoding scheme to represent the features of information including atom type, number of hydrogen atoms, number of directly-bond neighbors, whether belongs to a benzene ring. For example, after indexing all the atom types, the $m$-th atom type is encoded as a binary vector with length of 53 (i.e., the number of atom types), in which the $m$-th element is set to one while the others are set to zeros. The other three types of information are encoded similarly, which results in three binary vectors of lengths of 7 (maximum number of hydrogen atoms), 5 (maximum number of directly-bond neighbors) and 1, respectively. We denote the concatenation of the above four vectors by $\boldsymbol{x} \in \mathbb{R}^{D=53+7+5+1}$ as the features of corresponding atom, and thus the set of the product atoms is represented by $\boldsymbol{X}^{(p)} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, where

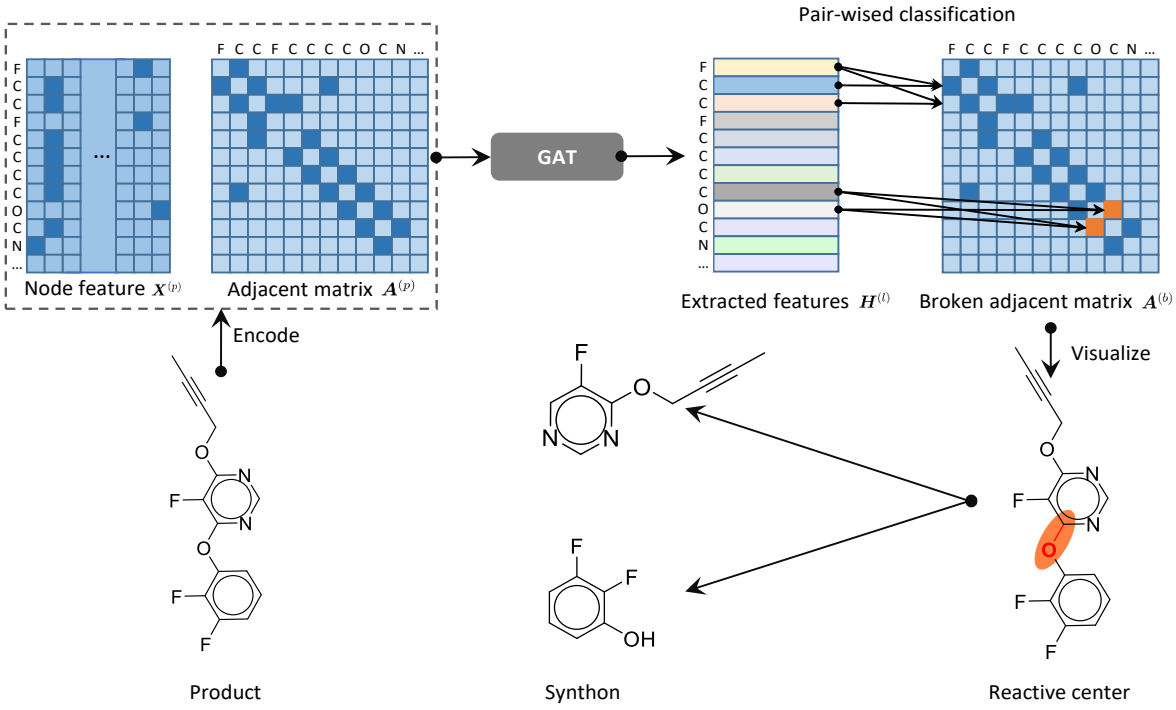


Figure 4: The architecture of graph-to-graph (CGraph2Seq) neural network. See main text for more details.

$N$ is the atom number of the product (Figure 4).

Given the atoms (also called nodes generally) features $X^{(p)}$ and the corresponding adjacent matrix $\boldsymbol{A}^{(p)}$, we leverage a graph attention network[31] (denoted as GAT) to capture the interaction information between atoms. Formally, GAT firstly performs shared self-attention mechanism on the nodes to indicate the importance of node $j$'s features to node $i$, computed by

$$s_{i,j} = \text{LeakyReLU}(\boldsymbol{u}^T[\boldsymbol{W}\boldsymbol{x}_i||\boldsymbol{W}\boldsymbol{x}_j]) \tag{1}$$

where $||$ represents concatenation, $\boldsymbol{W} \in \mathbb{R}^{F\times D}$ and $\boldsymbol{u} \in \mathbb{R}^F$ are learnable weight matrices, and LeakyReLU stands for LeakyReLU nonlinear function. To make coefficients easily comparable across different nodes, we then normalize them across all choices of $j$ using the softmax function:

$$a_{i,j} = \text{softmax}(s_{i,j}) = \frac{e^{s_{i,j}}}{\sum_{j\in\mathcal{N}_i} e^{s_{i,j}}}, \tag{2}$$

where $\mathcal{N}_i$ stands for the neighbors of node $i$.

Once obtained, the normalized attention coefficients together with the corresponding atom features are used to apply weighted summation operation, to derive the final processed features $\boldsymbol{h}_i$ for every node:

$$\boldsymbol{h}_i = \delta(\sum_{j \in \mathcal{N}_i} a_{i,j} \boldsymbol{W} \boldsymbol{x}_j), \tag{3}$$

where $\delta(\cdot)$ represents nonlinear function, e.g., ReLU function. GAT also employs multi-head attention to stabilize the learning process of self-attention, that is, $K$ independent attention mechanisms execute the transformation of Equation 3, and then their features are concatenated, resulting in the following output feature representation:

$$\boldsymbol{h}_i = \mathrm{GAT}(\boldsymbol{X}, \boldsymbol{x}_i) = ||_k^K \delta(\sum_{j \in \mathcal{N}_i} a_{i,j}^k \boldsymbol{W}^k \boldsymbol{x}_j), \tag{4}$$

where $||$ represents concatenation, $a_{i,j}^k$ are normalized attention coefficients computed by the $k$-th attention mechanism, and $W^k$ is the corresponding input linear transformation's weight matrix. Note that, in this setting, the final returned output $h_i$ will consist of $KF$ features (rather than $F$) for each node.

To extract deep representation of the product and increase the expression power of Graph2Graph model, we stacked $L$ layers of GAT:

$$\boldsymbol{h}_i^{(l+1)} = \mathrm{GAT}(\boldsymbol{H}^{(l)}, \boldsymbol{h}_i^{(l)}), \tag{5}$$

where $\boldsymbol{H}^{(l)}$ stand for the total processed features by $l$-th layer of GAT and $h_i^{(l)}$ indicates one of them.

Finally, we use the extracted representation of atom $i$ and atom $j$ to predict the probability of the corresponding bond being the reaction center, which is given by

$$p_{ij} = \mathrm{sigmoid}(\boldsymbol{W}^{(p)}[\boldsymbol{h}_i^{(L)} || \boldsymbol{h}_j^{(L)}]). \tag{6}$$

After obtained all the predicted probabilities (denoted as $\boldsymbol{P}$) of the products, Graph2Graph model can be directly optimized via maximizing the log-likelihood of the true broken adjacent matrix $\boldsymbol{A}_b$, i.e.,

$$\max_{\theta_p} \sum (1 - \boldsymbol{A}^b) \odot \log(1 - \boldsymbol{P}) + \boldsymbol{A}^b \odot \log \boldsymbol{P}, \tag{7}$$

where $\theta_p$ stands for the parameters of Graph2Graph and function $\odot$ indicates element-wise product.

## Generating reactants by the CGraph2seq network

Based on the predicted reaction centers, we regard the broken product as a undirected graph and split the broken product (represented by $\boldsymbol{X}$ and $A^{(b)}$) into several connected components.[32] Each of the connected component stands for a synthon (There might be only a single synthon if it is a decomposition reaction). For each synthon, we propose a novel calibrated graph-to-sequence (CGraph2Seq) neural network to generate the SMILES strings of the corresponding reactant molecule.

We again encode each synthon into two parts, namely the adjacent matrix $A^{(f)}$ and the atom features $\boldsymbol{X}^{(f)}$. CGraph2Seq leverage a another $L$-layer GAT to extract the graphical features of the synthon:

$$\boldsymbol{h}_i^{(f,l+1)} = \mathrm{GAT}(\boldsymbol{H}^{(f,l)}, \boldsymbol{h}_i^{(f,l)}), \tag{8}$$

where $\boldsymbol{H}^{(f,l)}$ stands for its total features processed by $l$-th GAT layer; $\boldsymbol{h}_i^{(f,l)}$ means the features of $i$-th atom; $\boldsymbol{H}^{(f,0)} = X^{(f)}, \boldsymbol{h}_i^{(f,0)} = X_i^{(f)}$.

On the basis of properly represented graphical features of the synthon $\boldsymbol{H}^{(f,L)}$ and the product $\boldsymbol{H}^{(L)}$, our model CGraph2Seq employ a novel attention[33] based Long Short Term Memory (LSTM)[34] architecture to generate the SMILES strings step by step. At each step $t$, to collect useful information of which character to generate, CGraph2Seq performs two

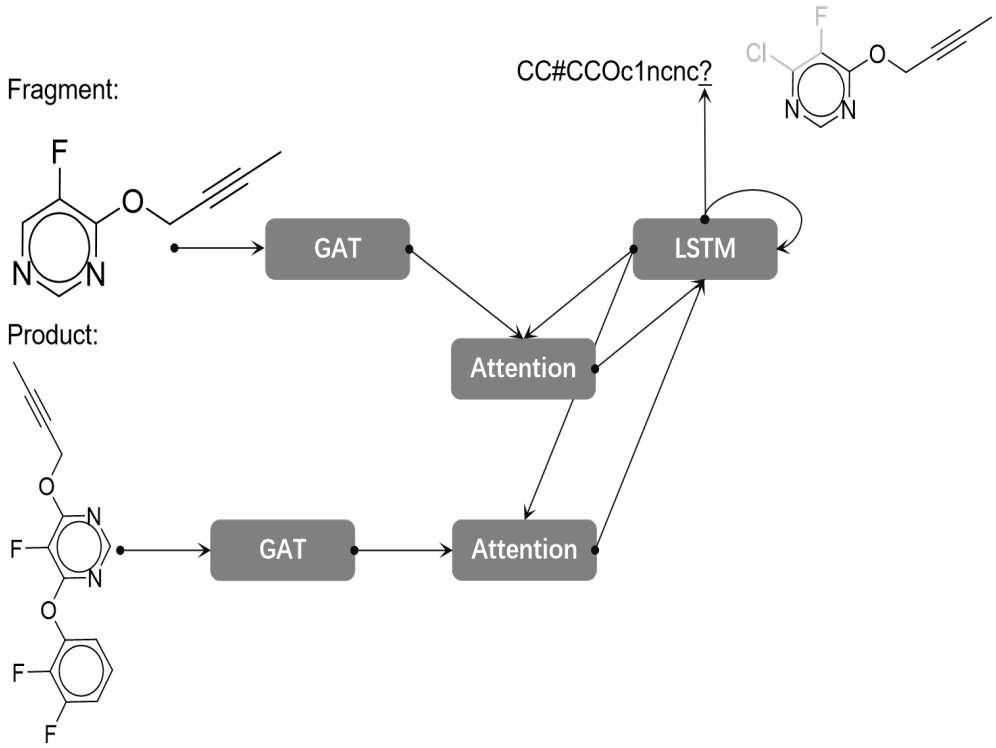


Figure 5: The architecture of calibrated graph-to-sequence (CGraph2Seq) neural network. See main text for more details.

attention operation on the atoms of the product and the synthon respectively.

$$e_i^{(f,t)} = (\tilde{\boldsymbol{W}}^{(f)} \boldsymbol{H}_i^{(f,L^f)})^T \boldsymbol{o}_t, \quad a_i^{(f,t)} = \text{softmax}(e_i^{(f,t)}) \tag{9}$$

$$e_i^{(p,t)} = (\tilde{\boldsymbol{W}}^{(p)} \boldsymbol{H}_i^{(L)})^T \boldsymbol{o}_t, \quad a_i^{(p,t)} = \text{softmax}(e_i^{(p,t)}) \tag{10}$$

where $\boldsymbol{o}_t \in \mathbb{R}^{D_h}$ indicates the hidden state at step $t$; $\tilde{\boldsymbol{W}}^{(f)}, \tilde{\boldsymbol{W}}^{(p)} \in \mathbb{R}^{D_h \times KF}$ are the learnable matrices; $a_i^{p,t}, a_i^{f,t}$ stands for the attention coefficients of atoms at current step $t$ in the product and the synthon respectively, based on which we can derived two context-dependent attention vectors, namely, $\boldsymbol{a}tt_p, \boldsymbol{a}tt_f$, given by

$$\boldsymbol{a}tt^{(p)} = \sum_i a_i^{(p,t)} \boldsymbol{H}_i^{(L)} \tag{11}$$

$$\boldsymbol{a}tt^{(f)} = \sum_i a_i^{(f,t)} \boldsymbol{H}_i^{(f,L^f)} \tag{12}$$

The LSTM in CGraph2Seq considers the above two attention vectors, the previous hidden states $\boldsymbol{o}^{(t-1)}$ and cells $\boldsymbol{c}^{(t-1)}$ , and the last output character together $y^{(t-1)}$, maintaining dependency among the entire generating history. Then a multiple-layer perceptron (MLP) neural network is built upon the hidden state $\boldsymbol{o}^{(t-1)}$ to predict a SMILES character at the current step $t$:

$$\boldsymbol{c}^{(t)}, \boldsymbol{o}^{(t)} = \text{LSTM}((\boldsymbol{att}^{(p)}, \boldsymbol{att}^{(f)}, y^{(t-1)}), (\boldsymbol{c}^{(t-1)}, \boldsymbol{o}^{(t-1)})), \tag{13}$$

$$\hat{p}^{(t)} = \text{softmax}(\text{MLP}(\boldsymbol{o}^{(t)})), \tag{14}$$

$$y^{(t)} = \underset{i}{argmax}\, \hat{p}_i^{(t)}, \tag{15}$$

where $\hat{p}^{(t)}$ means the predicted distribution of step $t$. Similar to GAT, we also stack $L^{(l)}$ layers of LSTM to enhance the its capability. Readers may refer to Prakash et al.[35] for details of stacked LSTM architecture.

The objective of CGraph2seq is also to maximize the log likelihood $J$ of the ground truth, given by

$$J = \sum_{m}^{M} \sum_{t}^{T^{(m)}} \log \hat{\boldsymbol{p}}_{\hat{y}_t}^{(t)} \tag{16}$$

where $M$ indicates the total number of data points; $\boldsymbol{p}_{\hat{y}_t}^{y,t}$ stands for the predicted probability of the golden character $\hat{y}_t$ (ground truth) at step $t$; $T^{(m)}$ means the number of generated steps at the $m$th data point.

During the testing, a beam search procedure is used for model inference. At each time step during decoding, we retain the top $N^{(b)}$ (beam width) candidate output sequences ranked by overall sequence log probability and continue to generate the predicted distribution of next characters on the basis of each of them. The decoding is stopped once the lengths of the candidate sequences reach the maximum decode length of 140 characters. Finally, we derive $N^{(b)}$ candidate output sequences (the reactants) for each the synthon.

# References

(1) Corey, E.; Wipke, W. T. *Science* **1969**, *166*, 178–192.

(2) Corey, E. J. *Pure and Applied chemistry* **1967**, *14*, 19–38.

(3) PENSAK, D. A.; COREY, E. J. *Computer-Assisted Organic Synthesis*; Chapter 1, pp 1–32.

(4) SzymkuÄĞ, S.; Gajewska, E. P.; Klucznik, T.; Molga, K.; Dittwald, P.; Startek, M.; Bajczyk, M.; Grzybowski, B. A. *Computer-Assisted Synthetic Planning: The End of the Beginning*; Angewandte Chemie International Edition, 2016; Vol. 55; pp 5904–5937.

(5) Law, J.; Zsoldos, Z.; Simon, A.; Reid, D.; Liu, Y.; Khew, S. Y.; Johnson, A. P.; Major, S.; Wade, R. A.; Ando, H. Y. *Journal of Chemical Information and Modeling* **2009**, *49*, 593–602, PMID: 19434897.

(6) Satoh, H.; Funatsu, K. *Journal of Chemical Information and Computer Sciences* **1995**, *35*, 34–44.

(7) Christ, C. D.; Zentgraf, M.; Kriegl, J. M. *Journal of Chemical Information and Modeling* **2012**, *52*, 1745–1756, PMID: 22657734.

(8) Bogevig, A.; Federsel, H.; Huerta, F.; Hutchings, M. G.; Kraut, H.; Langer, T.; Low, P.; Oppawsky, C.; Rein, T.; Saller, H. *Organic Process Research & Development* **2015**, *19*, 357–368.

(9) Segler, M. H. S.; Waller, M. P. *Chemistry: A European Journal* **2017**, *23*, 6118–6128.

(10) Coley, C. W.; Barzilay, R.; Jaakkola, T. S.; Green, W. H.; Jensen, K. F. *Prediction of organic reaction outcomes using machine learning*; ACS central science, 2017; Vol. 3; pp 434–443.

(11) Coley, C. W.; Rogers, L.; Green, W. H.; Jensen, K. F. *ACS central science* **2017**, *3*, 1237–1245.

(12) Segler, M. H. S.; Preuss, M.; Waller, M. P. *Nature* **2018**, *555*, 604–610.

(13) Liu, B.; Ramsundar, B.; Kawthekar, P.; Shi, J.; Gomes, J.; Luu Nguyen, Q.; Ho, S.; Sloane, J.; Wender, P.; Pande, V. *Retrosynthetic Reaction Prediction Using Neural Sequence-to-Sequence Models*; ACS Central Science, 2017; Vol. 3; pp 1103–1113.

(14) Coley, C. W.; Green, W. H.; Jensen, K. F. *Accounts of Chemical Research* **2018**, *51*, 1281–1289.

(15) Weininger, D. *Journal of Chemical Information and Computer Sciences* **1988**, *28*, 31–36.

(16) Baylon, J. L.; Cilfone, N. A.; Gulcher, J. R.; Chittenden, T. W. *Journal of chemical information and modeling* **2019**, *59*, 673–688.

(17) Gómez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; Aspuru-Guzik, A. *ACS central science* **2018**, *4*, 268–276.

(18) Lowe, D. M. Extraction of chemical structures and reactions from the literature. Ph.D. thesis, University of Cambridge, 2012.

(19) Schneider, N.; Stiefl, N.; Landrum, G. A. *Journal of chemical information and modeling* **2016**, *56*, 2336–2346.

(20) Segler, M. H.; Waller, M. P. *Chemistry–A European Journal* **2017**, *23*, 5966–5971.

(21) Law, J.; Zsoldos, Z.; Simon, A.; Reid, D.; Liu, Y.; Khew, S. Y.; Johnson, A. P.; Major, S.; Wade, R. A.; Ando, H. Y. *Route Designer: A Retrosynthetic Analysis Tool Utilizing Automated Retrosynthetic Rule Generation*; Journal of Chemical Information and Modeling, 2009; Vol. 49; pp 593–602, PMID: 19434897.

(22) Jin, W.; Coley, C.; Barzilay, R.; Jaakkola, T. Predicting organic reaction outcomes with weisfeiler-lehman network. Advances in Neural Information Processing Systems. 2017; pp 2607–2616.

(23) Ryu, S.; Lim, J.; Hong, S. H.; Kim, W. Y. *arXiv: Learning* **2018**,

(24) Zhou, J.; Cui, G.; Zhang, Z.; Yang, C.; Liu, Z.; Sun, M. *arXiv: Learning* **2018**,

(25) Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Yu, P. S. *arXiv: Learning* **2019**,

(26) others,, et al. *arXiv: Learning* **2018**,

(27) Shang, C.; Liu, Q.; Chen, K.; Sun, J.; Lu, J.; Yi, J.; Bi, J. *arXiv: Machine Learning* **2018**,

(28) Gilmer, J.; Schoenholz, S. S.; Riley, P.; Vinyals, O.; Dahl, G. E. *international conference on machine learning* **2017**, 1263–1272.

(29) You, J.; Liu, B.; Ying, Z.; Pande, V. S.; Leskovec, J. *neural information processing systems* **2018**, 6410–6421.

(30) Simonovsky, M.; Komodakis, N. *international conference on artificial neural networks* **2018**, 412–422.

(31) Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. *Graph attention networks*; ICLR, 2018.

(32) others,, et al. *Introduction to graph theory*; Prentice hall Upper Saddle River, NJ, 1996; Vol. 2.

(33) Bahdanau, D.; Cho, K.; Bengio, Y. *Neural machine translation by jointly learning to align and translate*; ICLR, 2015.

(34) Hochreiter, S.; Schmidhuber, J. *Long short-term memory*; Neural computation, 1997; Vol. 9; pp 1735–1780.

(35) Prakash, A.; Hasan, S. A.; Lee, K.; Datla, V.; Qadir, A.; Liu, J.; Farri, O. *Neural paraphrase generation with stacked residual LSTM networks*; COLING, 2016.