A Machine Learning Approach to Molecular Structure Recognition in Chemical Literature

by

Sophia Tabchouri

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2019

Author . . . .
Department of Electrical Engineering and Computer Science
May 19, 2019

Certified by . . . .
Regina Barzilay
Delta Electronics Professor
Thesis Supervisor

Accepted by . . . .
Katrina LaCurts
Chairman, Department Committee on Graduate Theses

A Machine Learning Approach to Molecular Structure Recognition in Chemical Literature

by

Sophia Tabchouri

Submitted to the Department of Electrical Engineering and Computer Science
on May 19, 2019, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Computer Science and Engineering

Abstract

Reaction diagrams in chemistry papers contain essential reaction information that is not available in the text. In order to extract comprehensive reaction information from chemistry literature, it is vital to convert these diagrams into a format compatible with searchable cheminformatic databases. Existing methods rely on rule-based procedures that have difficulty generalizing to noisy or different styled images. In this thesis, I implement a deep learning pipeline for identifying molecules in chemical diagrams and 'translating' the images into their corresponding SMILES strings. Diagram segmentation is performed using Mask R-CNN trained on an automatically generated set of diagrams. Translation to SMILES strings is performed using a neural machine translation model augmented with domain adaptation. Experimental results suggest that this model outperforms both rule-based and machine learning based models on diagrams extracted from real chemical literature.

Thesis Supervisor: Regina Barzilay
Title: Delta Electronics Professor

Acknowledgments

First, to my parents. Everything I have accomplished is possible because of the lessons you taught me. Thanks Mom and Dad.

I am grateful to my two brilliant sisters, Madeleine and Camille - who are the best (and coolest) female role models imaginable. Thanks as well to the other friends and family that have supported me.

With a special mention to Jiang Guo, without whom this thesis would not exist. Thank you for your careful guidance, essential feedback, and commitment as a research mentor.

Thanks also to my advisor Professor Regina Barzilay and all other MLPDS Consortium staff and sponsors for making research like this possible.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The design and discovery of novel pharmaceuticals requires access to large, accurate databases of chemical reactions. Recent machine learning methods of drug discovery especially rely on reaction information has been translated to a computer-readable and process-able format. Databases of this reaction information are mined from real chemistry literature. Manual extraction methods remain tedious and inefficient. They also expose the process to human error. With the amount of chemical literature growing exponentially, shown in Figures 1-1 and 1-2, automatic information extraction methods may prove to be the key to keeping chemical reaction databases up to date with current publications. Additionally, automatic extraction methods allow pharmaceutical companies to mine reaction information from pre-existing sources that have not been logged in digital forms.

The problem of digitizing chemical literature is not new. Previous methods, dating back decades, have addressed mining the text of chemistry papers. These methods range from grammar-based parsers to NLP machine translation models. But reaction data is unique in that much of the essential content like reaction conditions and reactants is *only* contained in the figures. Comprehensive reaction information extraction necessitates mining the image-based information.

Thus, I propose a deep-learning pipeline to identify molecules in chemical diagrams and convert them to a computer readable format. The pipeline consists of six steps, shown in Figure 1-3:

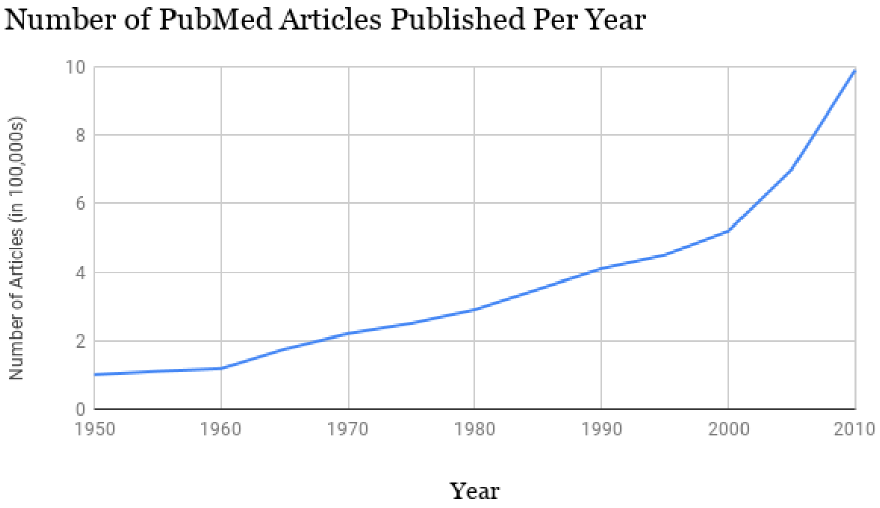


Figure 1-1: *Exponential Growth of PubMed Articles*

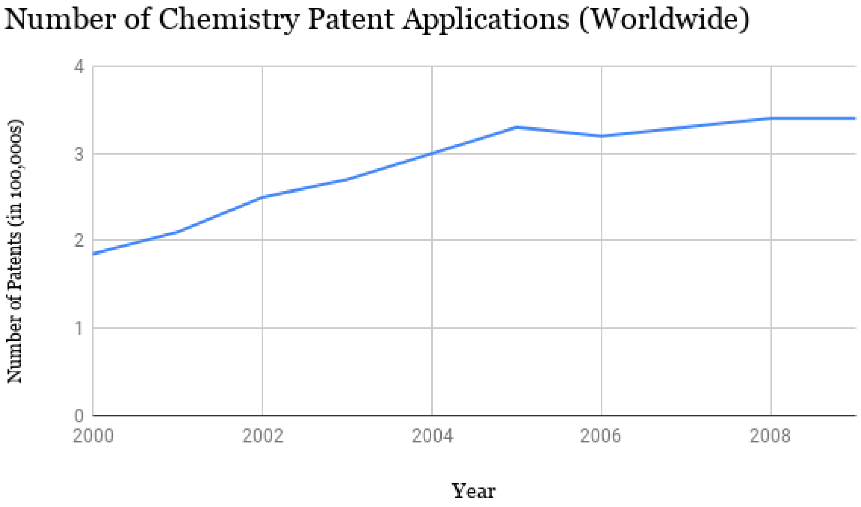


Figure 1-2: *Increase of Chemistry Patent Applications*

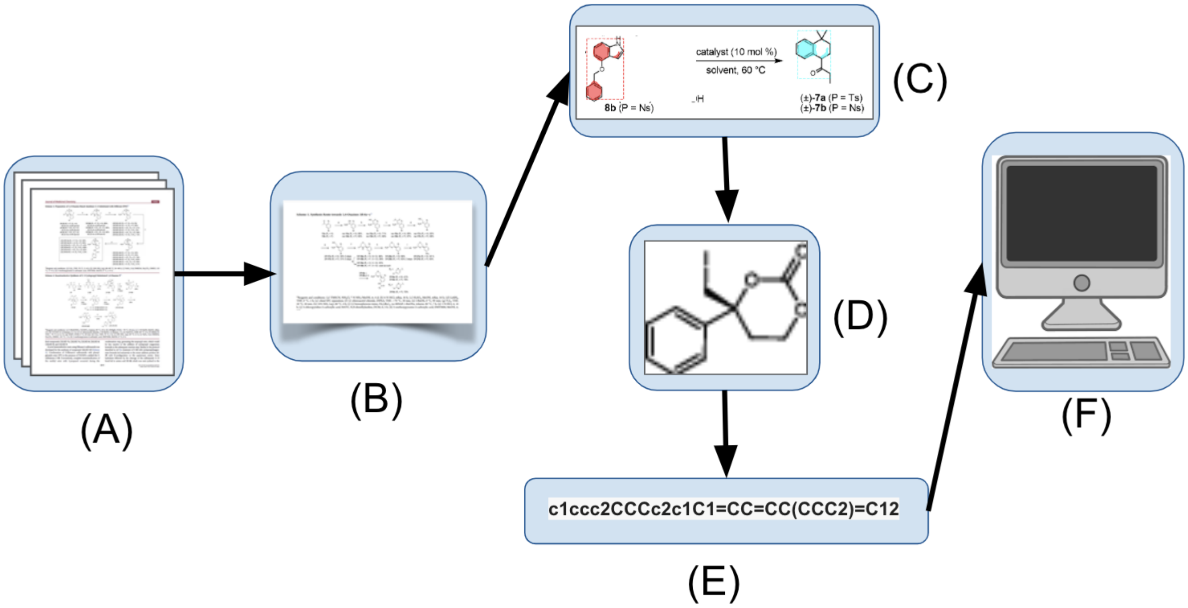


Figure 1-3: *Full System Pipeline*

A) The chemistry literature begins as PDF documents.

B) Diagrams are extracted form the documents using existing python libraries like pdfrw.

C) The locations of individual molecules within diagrams are identified.

D) The individual molecules are extracted from diagrams and input to translation models.

E) Translation models convert the molecular images to a SMILES string.

F) Databases of SMILES strings can be built from large amounts of chemistry literature, and used as input to drug discovery models.

1.1 Project Overview:

The chapters of this work cover, in order:

1. **Related Work:** In this chapter, I explore the positive and negative aspects of previous work in chemical information extraction.

2. **Diagram Segmentation:** This chapter focuses on the segmentation step of the pipeline. It outlines the use and modification of Mask R-CNN for bounding box identification.

3. **Image to SMILES Translation:** In this chapter I describe the creation of a domain adversarial model for converting an extracted molecule into its corresponding SMILES string. The foundation of this model is a neural machine translation model with convolutional and recurrent encoders, and an attention-based recurrent decoder.

4. **End-to-End Pipeline:** The final goal of this project is to create a user-friendly end-to-end pipeline where a user can submit a diagram and the recognized molecules will be returned, along with their predicted SMILES strings.

Chapter 2

Related Work

2.1 Chemical Information Extraction

Information extraction methods have shown promising results extracting important chemical entities from document text [21]. However, these methods ignore reaction diagrams, thus losing essential reaction information in some cases.

To prevent this loss of information, rule-based methods were created to convert diagrams to computer-readable formats.

These programs generally follow the same five step pipeline with some variation:

1. applying algorithms that identify connected components that may be atoms or bonds

2. vectorizing the image to get a graph of vectors in place of multi-pixel wide lines

3. identification of dashed and standard bond lines

4. optical character recognition to identify atoms

5. reconstruction of the connection graph of the entire molecule using the vectorization.

Kekule, the first program of this kind, developed by McDaniel et al.[15], uses neural nets to assign confidence rankings on ocr for individual tokens within the diagram. It does not perform ocr at the entire diagram level. OSRA [4] is an open-source library released in 2009 that improved the rule-based pipeline by performing extraction at three different resolutions and choosing the interpretation with the highest confidence value. ChemOCR [2] looked to enhance the final step of connection graph reconstruction by selecting the best-matching chemical structure fragment against sub-graphs of chemical structures stored in a database using a graph-matching algorithm.

Other works like CliDE [23] and ChemREADER [16] also advance the 'rule-based' pipeline, but no amount of augmentation to rule-based methods will solve an underlying problem - namely that these methods require precise knowledge of the image data domain to define thresholds and procedures. Molecular diagrams can contain complex, rare, or ambiguous representations that are difficult to account for in a rule-based system. As such, they are not robust to features that are not explicitly addressed in their rules, and may not perform as well on noisy images.

These problems are identified by Staker et al. [20], who contend with the problem by applying deep learning techniques to molecule recognition. Their structure prediction model consists of a convolutional encoder and an attention-based decoder constructed with three GridLSTM cells.

They were able to produce accuracies from 41% to 83 %. However, their model required on the order of millions of training examples. As pointed out in the Staker et al. [20] work, annotating new training data from real literature is extremely time-consuming, requiring someone to manually redraw the molecular diagrams into rendering programs that can output their SMILES. It is impractical to generate datasets of that size for diagrams taken from real literature. Additionally, because their exact datasets are not publicly released, it is difficult to accurately gauge their performance on molecules of varying styles or compare to other methods.

We seek to improve model transferability to noisy diagrams and diagrams from new data domains without requiring millions of new labelled training examples. To improve comparison across implementations for molecule ocr, we release our system and data as a benchmark.

2.2 Unsupervised Domain Adaptation

Our model resembles domain adaptation methods based on work by Ben-David et al. [3] that align the feature space between source and target domains.

Works from Ganin et al. [5] and Ajakan et al. [1] use feature extraction layers with two separate classifiers on their output. In addition to the task-specific output classifier, they add a classifier responsible for discriminating between examples from the source and target domain. These domain adversarial neural networks (DANNs) minimize the task-specific loss while trying to maximize the domain classification loss.

Tzeng et al. [22] and Long et al. [13] replace the domain classification output with the Maximum Mean Discrepancy (MMD) metric [8] to measure "distance" between samples in the feature space. These models minimize the MMD metric between featurized batches from the source and target domain. The Deep Domain Confusion Network by Tzeng et al. [22] minimizes the MMD loss of a single output layer from the neural net, while Long et al. [13] compute the MMD at multiple layers.

In this work, we apply the maximum mean discrepancy method to an optical character recognition model by minimizing the MMD between the featurized output of the recurrent encoding layer.

2.3 Image Segmentation

In 2012, the publication of AlexNet [11] ignited a storm of innovation in the field of computer vision using deep learning. The first of its kind, AlexNet became a cornerstone of image processing - the convolutional neural net (CNN). This innovation was extended from image classification to other foundational tasks of computer vision - image segmentation, bounding box identification, object classification, and instance segmentation. Iterations of the original CNN now permit a single neural net to perform all of the tasks. To fully understand current deep learning computer vision models, we can follow the timeline of computer vision, from AlexNet in 2012 to the current state-of-the-art Mask R-CNN model in 2017.

In 2014, VGG [19] was released and marked a substantial improvement over AlexNet with a mere 7.3% error rate on ImageNet. The principle of VGG was straightforward: in computer vision, the networks must be deep and simple. VGG used a staggering 19 layers, but managed to keep the number of parameters down by reducing the kernel size. Instead of using kernels with edge lengths of 11 or 7, they relied on multiple layers of kernels with edges sizes of 3 or 2. This results in the same receptive field as using larger kernel sizes with fewer layers.

In 2013, around the same time that the creators of VGG were having great success with extremely deep convolutional models, researchers at UC Berkeley began working on region-based object detection models. Instead of solely classifying which objects were contained in an image, they devised architectures that could determine the exact locations of recognized objects.

In the R-CNN [6], this process is split into region proposal and classification. The first iteration of region based methods used a non machine learning based method of region proposal: selective search. The 2,000 regions proposed by the region proposal network are fed into a CNN that acts a feature extractor. The featurized vector is then fed into separate neural nets for each class, and classified as the output with the highest probability. It is also input to a bounding box regressor to fine-tune the bounding box.

However, R-CNN was slow and inefficient, requiring approximately 53 seconds per image. To improve runtime, the order of region proposal and feature extraction were switched. Instead of running a CNN on each proposed region, Fast R-CNN [7] proposed regions within the featurized representations. Later iterations of R-CNNs sought to replace the non deep-learning based modules with neural nets. Faster R-CNN [17] used a region proposal neural network, instead of selective search to identify regions of interest in the featurization.

The most recent iteration, Mask R-CNN [9], replaced the object classification and bounding box refinement models with neural nets, but it also introduced a new task to the region-based networks. As the name implies, Mask R-CNN also computed a pixel-level mask of the exact location of the object. This mask is computed using a fully convolutional network, with deconvolution to reconstruct the original image size. In this paper, I experiment with Mask R-CNN to identify molecule bounding boxes and segmentation masks with varying levels of success.

Chapter 3

Diagram Segmentation

Diagram Segmentation is the process of identifying specific objects in an image - in this case molecules. In this section, I evaluate the results of segmenting chemical diagrams by identifying molecule bounding boxes and pixel-level segmentation masks. While highly effective models exist for segmentation, they rely on large training datasets. No labelled training datasets for molecule diagrams segmentation currently exist, so the main challenge in the stage of the pipeline is generating large training datasets.

3.1 Data Generation

There are no existing open source datasets for molecular bounding box identification. A major challenge was generating enough clean data to train a segmentation model. Roughly, the steps for generating this data were:

1. Use existing rule-based libraries to identify the locations of molecules in chemistry papers.

2. Replace those identified bounding boxes with automatically generated molecules with known segmentation masks.

Because existing extraction libraries are imperfect, this process involves filtering and quality control. The following sections detail the data generation procedure.

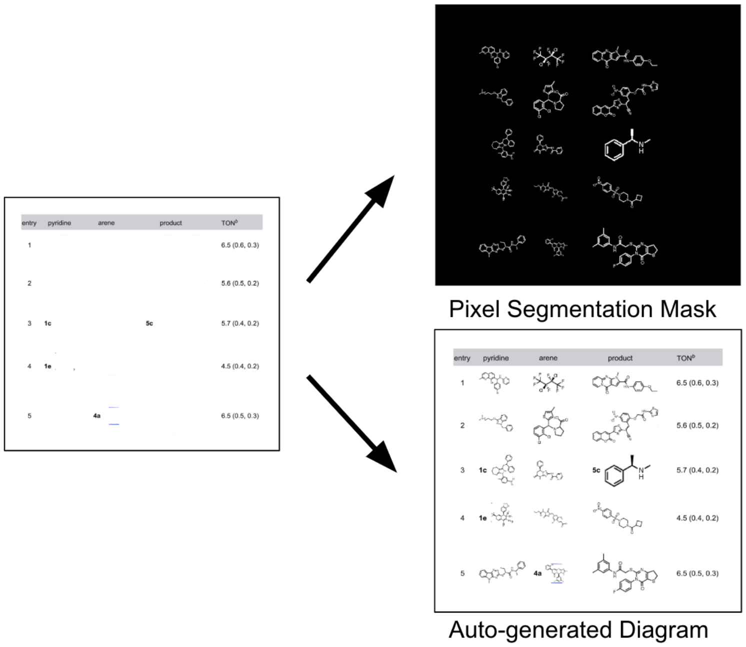


Figure 3-1: *Example of Repopulated Diagram with Corresponding Segmentation Mask*

3.1.1 Automatically Generated Diagrams

Training Mask R-CNN requires a specific format of labelled data. Each training sample must consist of a single sample image, with separate mask files for each object in the image. A mask file is a binary image of the same size as the original image. The binary image contains white in pixels that correspond to object pixels in the original image, and black elsewhere. As example is shown in Figure 3-1.

There are no published datasets for this task, and manual pixel level annotation is impractical. For these reasons, diagrams with known pixel masks had to be automatically generated. To make sure that the diagrams resembled real chemistry diagrams, I created the automatic diagrams using real diagrams as a foundation. I also had access to 100,000 molecule-only images. Pixel masks for these images could be easily generated because all black pixels belonged to the molecule, and white pixels were background. These images merely had to be inverted to get their truth mask.

The data generation pipeline consisted of the following steps:

1. Use the rule-based molecule extraction program, OSRA, to identify the locations

Table 3.1: Data Generation Statistics

Description	Number of Samples
ACS Papers	4000
Diagrams Extracted from Papers	45253
After Manually Filtering Poor OSRA Results	20280
Final	
Train	16280
Test	2000
Valid	2000

of molecules in diagrams form chemistry papers.

2. Manually filter out every diagram that OSRA did not work well on, approximately 60% of the diagrams.

3. On the remaining images, ’white-out’ all bounding box areas that contain molecules.

4. Repopulate the ’white-out’ areas with the automatically generated molecules with known truth masks, adding this to a truth mask of the entire image.

Data statistics for each step of the pipeline are shown in Table 3.1.

3.2 Model

3.2.1 Mask-RCNN for Instance Segmentation

The model that was used to identify molecular structures within diagrams is Mask R-CNN. Mask R-CNN is an open source deep convolutional network that was created in 2017 [9]. Mask R-CNN can perform three different computer vision tasks: bounding box identification, pixel-level instance segmentation, and object classification. We only have a single object class to look for (molecules), so we are only concerned with the accuracies of the bounding box and segmentation outputs.

As an image moves through the Mask R-CNN model, the following four outputs are detected using convolutional layers, each with a separate loss that are summed to find the total loss.

1. Up to 1000 regions of interests are identified. Regions of interest (ROI) are rectangular areas in the image that meet a probability threshold of containing an object. The ROIs are extracted from the image and fed through to following three fully convolutional networks in parallel.
2. A bounding box detection model refines the ROI bounding box detections.
3. A classifier identifies which class an identified object belongs to.
4. Segmentation layers detect a binary mask that identifies exactly which pixels belong to the object.

Specific details of final model configuration are relegated to the appendix.

3.3 Results

3.3.1 Evaluation Metric

The metric that is commonly used to evaluate object detection and segmentation tasks is maximum average precision (mAP). The calculation of mAP scores is calculated using precision, recall, and Intersection over Union (IoU). Traditional definitions of precision and recall are used:

$$\textbf{TP} = \text{True Positive} \qquad \textbf{FP} = \text{False Positive} \qquad \textbf{FN} = \text{False Negative}$$

$$\textbf{Precision} = \frac{TP}{TP + FP}$$

$$\textbf{Recall} = \frac{TP}{TP + FN}$$

An IoU metric is used to measure the overlap between two bounded areas, as shown in 3-2.

$$\textbf{IoU} = \frac{\textbf{area of intersection}}{\textbf{area of union}}$$

In order to find average precision values, we have to treat bounding box identification as a classification problem. An IoU threshold is selected and bounding boxes

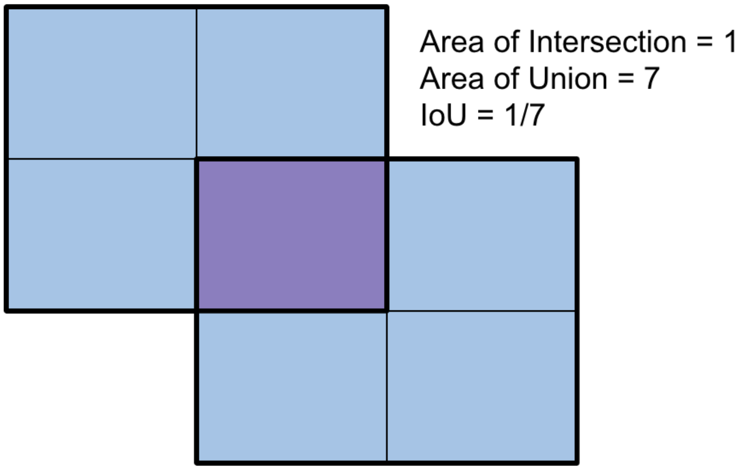


Figure 3-2: *Visual Representation of IoU Calculation*

with IoU less than that threshold are considered incorrectly classified, and IoU values greater than the threshold are considered correctly classified.

We can rank our bounding box predictions by the predicted confidence level, and then measure the precision and recall values of our model as we add each predicted bounding box to our calculations, in the order of most confident to least confident. We graph these values with precision on the y-axis and recall on the x-axis, and find average precision by calculating the area under this curve. The only change in calculating **maximum** average precision is that each precision value is replaced with the maximum precision value so far. This can be represented mathematically by replacing the precision value p at recall r with the maximum precision value $\tilde{p}$ for any recall greater than r:

$$p(r) = \max_{\tilde{r} \geq r} p(\tilde{r})$$

Since our data was formatted using the COCO style, this research reports the standard metrics for COCO datasets, defined in Table 3.2.

3.3.2 Model Performance

Examples of the bounding box and segmentation results are shown in Figure 3-3. More examples of results are in Appendix B. The bounding boxes are indicated by

Table 3.2: Standard COCO Dataset Metrics

Average Precision (AP)	
AP	% AP at IoU=0.50:0.05:0.95
$\text{AP}_{IoU=0.50}$	% AP at IoU=0.50
$\text{AP}_{IoU=0.75}$	% AP at IoU=0.75
AP across scales	
AP_{small}	% AP for small objects: area $\leq 32^2$
AP_{medium}	% AP for medium objects: $32^2 \leq area \leq 96^2$
AP_{large}	% AP for medium objects: area $\geq 96^2$

the dashed lines, and the segmentation are the shaded regions.

The segmentation results are not accurate enough to pass on to the SMILES translation model. They consistently fail to identify the branches of molecules, or any fine-grained regions. This effect is predictable, as Mask R-CNN is more commonly used to identify large continuous segmentation regions. Mask R-CNN only uses a single up-sampling layer to enlarge the pixel-level mask to the size of the original image. This differs from segmentation models that are specifically designed to identify fine-grained features. Such models, like U-Net [18], include multiple upsampling layers that have residual links to original down-sampling layers, so that the model has some notion of the original image as it upsamples.

However, the abysmal performance of Mask R-CNN in identifying pixel level masks is made up for in its bounding box performance. For this reason, I chose to focus on generating highly accurate bounding boxes, and disregard the pixel level segmentation masks.

3.3.3 Realistic Data Results:

Because the realistic data are not annotated, it is not possible

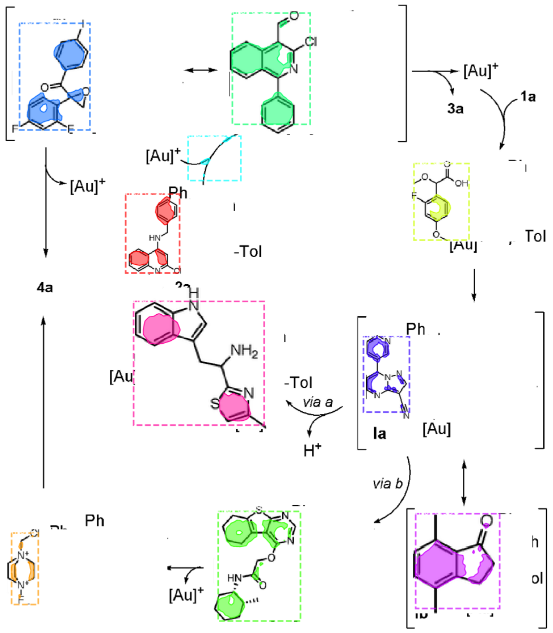


Figure 3-3: *Visual Representation of IoU Calculation*

3.4 Post-Processing Methods

When analyzing the visualized results on the realistic dataset, I found that the bounding boxes often failed to include the entire molecule. For example, they would include nearly all of a molecule, except a single bonded atom. This phenomenon is demonstrated in 3-4. This doesn't influence the IoU value significantly, so the model may have trouble learning these extra, unconnected atoms. Instead of introducing more training data, I defined a simple post-processing method and tweaked the heuristics using a small subset of 200 realistic training examples.

A naive approach to pre-processing would be to simply extend all bounding boxes by a pre-defined threshold amount, such as 10-20 pixels. However, the size of molecules can vary widely, from 200 pixels - 2000 pixels. This approach would fail

Table 3.3: Mask R-CNN Precision Results

Average Precision (AP)	
AP	0.529
$\text{AP}_{IoU=0.50}$	0.911
$\text{AP}_{IoU=0.75}$	0.590
AP across scales	
AP_{small}	0.138
AP_{medium}	0.569
AP_{large}	0.541

Table 3.4: Mask R-CNN Recall Results

Average Precision (AP)	
AP	0.130
$\text{AP}_{IoU=0.50}$	0.528
$\text{AP}_{IoU=0.75}$	0.596
AP across scales	
AP_{small}	0.134
AP_{medium}	0.572
AP_{large}	0.625

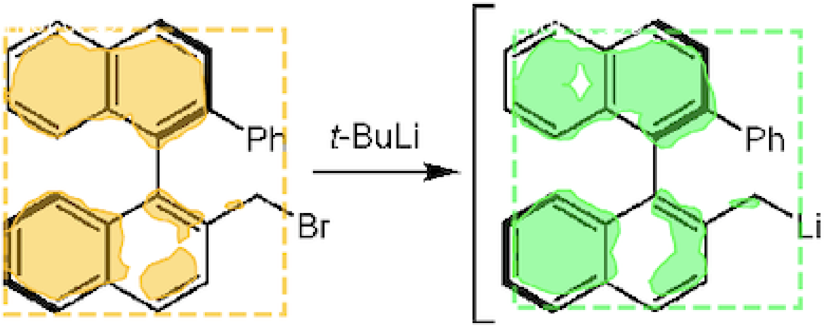


Figure 3-4: *Example of Inadequate Bounding Box*

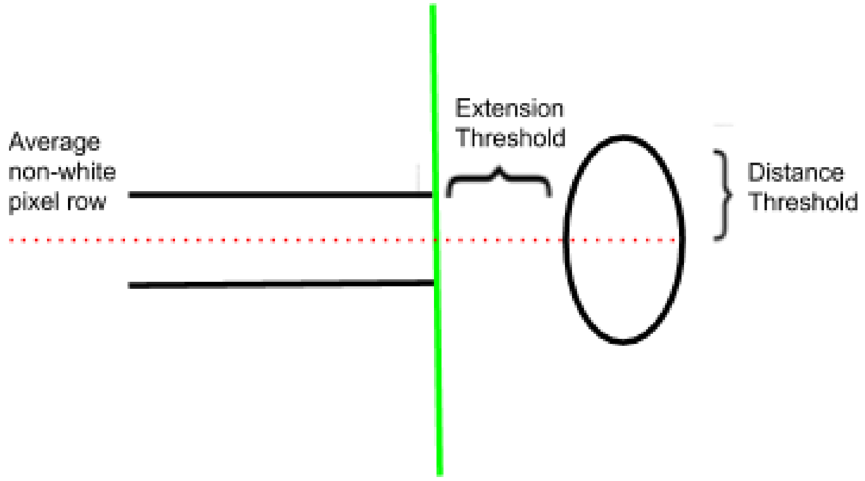


Figure 3-5: *Post-Processing Thresholds Visualized. Double bond connected to Oxygen atom.*

in most cases.

The post-processing method relies on the fact that most of the features that weren't included were atoms occurring at the end of a bond line. By that logic, the boundary should only be extended if there are non-white pixels outside the boundary that are in the same row or column as non-white pixels inside the boundary. I define two thresholds:

1. Extension threshold: how many columns outward to look for non-white pixels
2. Distance threshold: how far apart laterally the non-white pixels inside and outside the boundary can be.

A visualization of the two threshold is shown in 3-5. In the case of the figure, the green line indicates the original boundary, and the red line indicates the average non-white pixel row location on the outermost right column within the bounding box. Columns are checked for non-white pixels, outward to the extension threshold. At each column, the average non-white pixel location of that column is calculated. If that location falls within the distance threshold from the average non-white pixel location within the boundary, the boundary is extended to that column. Additionally, once the boundary has been extended to a new column, it extends until it reaches the next all-white column, to assure that the entire atom symbol is added.

The example shown in Figure 3-5 shows a double bond to an Oxygen atom. The green line indicates the boundary identified by Mask R-CNN. The red dashed line shows the average row location of non-white pixels in the right-most column of the bounded region, in this case, between the two bond lines. Columns to the right of the boundary are checked to see if their average non-white pixel row location falls near the red line.

As shown in Figure 3-5, the extensions threshold would not go far enough to reach the oxygen atom; the bounding box would not be altered. However, if the extension threshold were larger, it would reach the first column of the oxygen atom. That column clearly has a similar average non-white pixel location to the red dotted line, so the boundary would be extended to encompass the entire O. As shown in the original Figure 3-4, having the distance threshold for the mean pixel location prevents the yellow bounding box from being extended to included the reaction arrow. Similar procedures are followed for each other boundary of the bounding box.

The post-processing is applied to the realistic dataset. Because the realistic dataset is not annotated, it is not possible to get new bounding box IoU metrics on the post-processing technique. However, qualitative analysis of the results showed that, while it only helped approximately 75% of the images, the post-processing technique never negatively affected the bounding box.

Chapter 4

Image to SMILES Translation:

In this section I seek to demonstrate that a domain adaptation based learning model improves the robustness of optical molecule recognition to different drawing styles and noisy images and create a benchmark for converting realistic molecular images to their corresponding SMILES representations.

Problem Definition

We only have access to labelled training examples for automatically generated molecules. However, we would like our ocr model to perform well on molecules with styles that are not included in our automatically generated dataset. Thus we frame this as a domain adaptation problem, where the automatically generated molecules are our source domain S, and the realistic molecules are our target domain T.

Optical Molecule Recognition

We use the neural machine translation model implemented by Klein et al. [10] as the foundation for our domain adaptation model. Our model consists of an *encoder* and *decoder*.

The decoder is made up of a convolutional and a recurrent encoder. The convolutional encoder is able to encode spatial information in the image, while the recurrent encoder can encode important sequential information about the molecular structure.

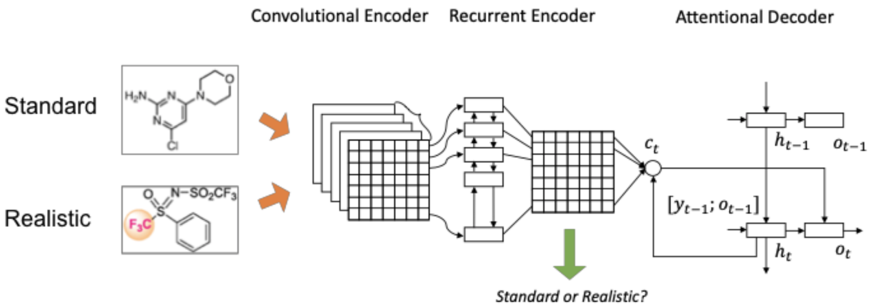


Figure 4-1: *Architecture of Optical Molecule Recognition Model*

The encoded feature representation is passed to the decoder. The decoder generates the SMILES string token by token. The generated token at time t depends upon the previous token $t-1$, as well as a weighted combination of all features in the encoded feature vector. Because our decoder is attention-based, the weights are learned during training, representing how much "attention" the model should pay to different features in the encoding when generating each token. Specific details of model architecture are relegated to the implementation section.

Improving Robustness with Domain Adaptation

The OpenNMT model relies only on labelled training data, however we only have labelled training data for our source domain. We need to ensure that the model encodes data from the source domain and the target domain into the same feature space. We add an output after the encoder layers, shown in Figure 4-1. This output layer takes the encoded feature vectors form the source and target domain and calculates the 'distance' between them, quantified by the maximum mean discrepancy. Minimizing this distance suggests that the source and target domain have been mapped to the same feature space, and only contain features that the decoder can accurately decode.

Training

During training, the labelled source data and unlabelled target data are separately split into smaller training batches. At each iteration, a batch from the source and target domain are passed through the encoding layers. The maximum mean discrepancy is calculated between the two encoded batches.

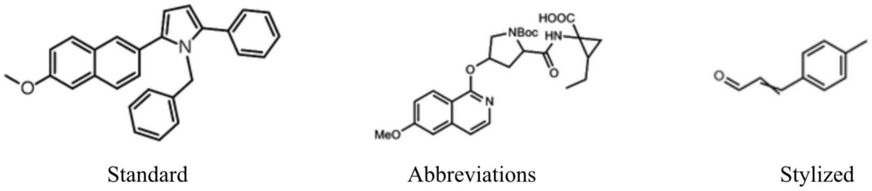
The encoded source batch is fed through the decoder and the translation loss is also calculated. Both the translation loss and MMD loss are backpropogated to train the model weights.

Dataset Construction

Labelled Data

The domain adaptation model requires labelled data from our source domain and unlabelled data from our target domain.

The source domain consisted of molecular structures with corresponding SMILES labels. It was automatically generated using ChemDraw. There were three different domains of images contructed using ChemDraw, shown in Figure 4-2. The standard domain contained only single atom labels and traditional bond lines. The abbreviations domain contained superatom symbols. The stylized domain contained additional bond and molecule drawing styles.



Standard

Abbreviations

Stylized

Figure 4-2: *Automatically Generated Molecules*

Unlabelled Data

Our target domain molecules were extracted from real chemical literature using a Mask R-CNN model [9] that is trained specifically on chemical diagrams to identify molecules. An example of the bounding boxes recognized by Mask R-CNN are shown in Figure 4-3.

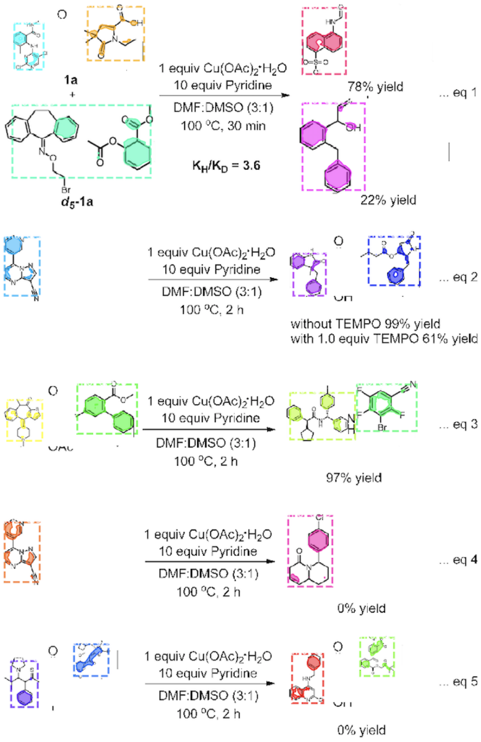


Figure 4-3: *Mask R-CNN Output*

This unlabelled dataset contains molecules that were not drawn using ChemDraw. Thus, it differs from the automatically generated molecules in rendering style, scale, resolution, font, and line style and width. Additionally, the Mask R-CNN model does not have 100% accuracy, so extracted molecules may contain artifacts such as captions, arrows, and other extraneous text. The model also extracts some diagrams that are not molecules. In this paper, no quality control filtering was performed on the images extracted by Mask R-CNN, so the unlabelled data contains these imperfections. We suspect that the molecule recognition model could be improved by implementing some quality control on which extracted images we allow in the unlabelled target domain dataset.

Data Preprocessing

Both the labelled and unlabelled data were split into training,test, and validation sets then run through the same preprocessing pipeline. There are two stages to

Table 4.1: Data Statistics

		Initial	**After Preprocessing**
Realistic	Train	90000	83450
	Validation	-	-
	Test	121	121
Auto-Generated	Train	150000	14856
	Validation	20000	19311
	Test	20000	19421

preprocessing.

1. **Preprocess the images.** To reduce the data size, all images were downsampled (by a factor of 2 for the labelled data, and 6 for the unlabelled data due to unlabelled images being larger in size). Pad images with white space so that they match the size of predetermined buckets. This step is essential to create batches of images with the same size during training.

2. **Filter large data.** Remove any data samples with SMILES strings longer than 150 tokens. No unlabelled data is filtered out in this preprocessing step because they do not have associated SMILES strings. Filter out images larger than the maximum allowed width **x** height (200 **x** 500 for labelled data, and 400**x**400 for unlabelled data, due to unlabelled images being larger in size).

Experimental Setup

Evaluation

Results are evaluated using three different metrics: exact SMILES matching accuracy, valid SMILES strings, and BLEU score.

Exact SMILES matching accuracy is defined as the ratio of test examples whose predicted string is an exact match to the true string.

The model may predict SMILES strings that are not a valid representation of a molecule - for example, a SMILES string that does not have correct alignment of opening and closing parenthesis. The valid SMILES metric is the ratio of predicted strings that are valid.

The BLEU score, commonly used to evaluate machine translation tasks, is a measure of the amount of n-grams that are present in both the predicted and actual SMILES string, where an n-gram is defined as a series of n adjacent tokens in the SMILES string.

Baselines

We verify the efficacy of our domain adaptation approach (referred to going forward as OpenNMT*A) against rule-based and non-adversarial approaches, comparing to the following two baselines:

1. **OpenNMT:** the OpenNMT model without the domain adaptation augmentations

2. **OSRA:** a rule-based approach to molecule ocr

The models are evaluated on their performance on the set of 200 test images extracted from real chemistry papers. On each, we report our three evaluation metrics.

Implementation Details

Our model was built using the OpenNMT library [10] with the following architecture.

The convolutional encoder has six 2D-convolutional layers, each with stride 1, kernel size of 3, and 1 unit of padding. The number of filters for each layer, in order, is [64, 128, 256, 256, 512, 512]. After each convolutional layer is a 2D-maxpool layer with kernel size of 2 and stride of 2. The recurrent encoder is a bidirectional LSTM cell with 64 features in the hidden state. The attention-based rnn decoder has 128 hidden states, uses Luong attention [14], a scaled-dot product [24] alignment score function, and softmax global attention function.

Results

Table 4.2 compares the results of all three extraction methods. Both versions of the OpenNMT model outperform the rule-based OSRA. The domain adversarial model performs markedly better on the realistic dataset than both OSRA and non-adversarial OpenNMT. The domain adversarial model saw the greatest improvement in how many valid SMILES are produced on the realistic dataset, suggest that the domain adversarial training allows the model to learn some type of rules for what SMILES strings are valid. Even when tested on realistic data containing new and ambiguous symbols, OpenNMT*A is still able to return mostly valid SMILES strings. The problem of forcing models to return valid SMILES strings is ongoing in deep learning for chemistry, even in tasks not related to information extraction. These results suggest the experimentation with domain adaptation could improve such models.

Table 4.2: Evaluation Results on Realistic and Automatically Generated Data

	Auto-Generated			**Realistic**		
Model	BLEU	valid	exact	BLEU	valid	exact
OpenNMT*A	-	-	-	40.63	88.5	17.9
OpenNMT	78.3	95.0	54.0	34.5	63.5	11.6
OSRA	64.0	72.0	56.4	24.1	31.0	12.0

Chapter 5

Information Extraction Pipeline:

The previous sections detail the design and training of optical molecule recognition models. This section focuses on creating a pipeline that can run the information extraction procedure from end-to-end. The pipeline is designed to take as input a single chemistry PDF and return a folder including the extracted images, their SMILES strings, and important entities from the text. The pipeline is also modular, allowing for easy changes in which parts of the system to run.

5.1 Designing an End-to-End Pipeline:

The pipeline is structured as a single class, with individual methods representing different steps of the pipeline. Some methods require other preprocessing steps, while others can be run stand-alone.

By structuring the pipeline in this way, the user can easily design customized extraction pipelines. For instance, if a user only wants to insert diagrams, instead of entire PDFs, for image-based information extraction, an extraction sequence can be designed to skip the text-based method. If a user does not want the extracted images to be displayed, the HTML visualization step can be skipped in the extraction method.

The different types of extraction must be explicitly defined in code, but the modular design makes it easy to create these new extraction methods.

5.1.1 Modules:

1. **EXTRACTOR INITIALIZATION**

   This is the initial step when a user inputs a PDF file. The PDF extractor class is initialized with the PDF filename, and a project folder is created on the server. Each project folder is given a six-digit random ID, to prevent concurrency errors when multiple users are running the demo.

2. **GET CHEMISTRY DOCUMENT**

   This module performs text-based information extraction. For the current pipeline, this step relies on ChemDataExtractor, which is a grammar-based text parser.

3. **GET DIAGRAMS**

   The PDF document is searched for images. A Python library called pdfrw is used to identify and save all XObjects as png files.

4. **CREATE JSON**

   The image input to Mask R-CNN must be in the same JSON format as the MS-COCO dataset [12]. This method acts as a pre-processing module for the diagram input to the segmentation model.

5. **GET SEGMENTATIONS**

   This module uses a JSON file as input to the Mask R-CNN model. It runs segmentation inference on the diagrams and saves the visualized bounding boxes, as well as extracted molecules, to the project folder.

6. **OMR DATA PREP**

   The SMILES translation model requires a text file with a list of molecule diagram filenames. This method simply creates that list using the diagram directory.

7. **TRANSLATE TO SMILES:**

   This module calls the inference process for SMILES string translation. It creates a text file with all of the SMILES strings listed.

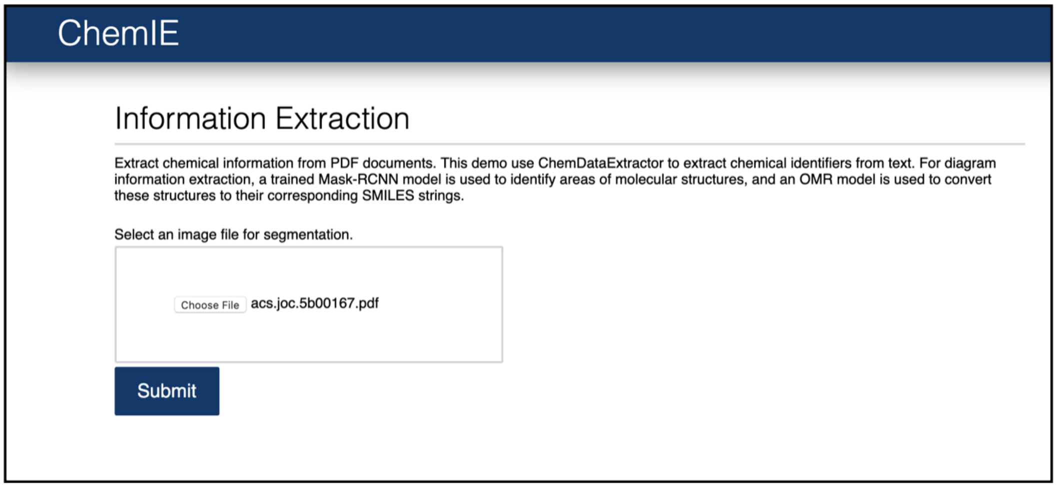
8. **CREATE HTML DATA:**

   The final module is only necessary if you would like to display the visualized and extracted molecules with their SMILES strings. It creates a file list that is compatible with how Flask hosts and sends file names to HTML templates.

5.2 Web Demo:

The pipeline can run stand-alone as a python program. For users with no programming background, I designed a web-based application that can run the pipeline.

The pipeline is wrapped in a web application built using the Flask API. A graphical user interface promotes easy use. Screenshots of this interface are show in 5-1 and 5-2.



ChemIE

Information Extraction

Extract chemical information from PDF documents. This demo use ChemDataExtractor to extract chemical identifiers from text. For diagram information extraction, a trained Mask-RCNN model is used to identify areas of molecular structures, and an OMR model is used to convert these structures to their corresponding SMILES strings.

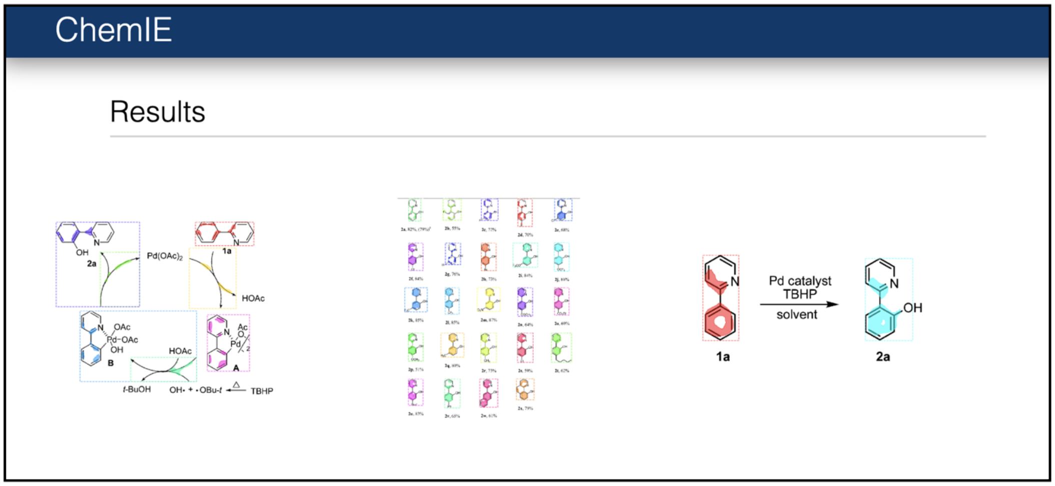
Select an image file for segmentation.

acs.joc.5b00167.pdf

Figure 5-1: *Web Demo Home Page*

The GUI is designed to minimize ambiguity in how to run the software. The main page simply asks a user to upload a PDF, then runs the entire pipeline and automatically sends the output zip file to the user's device for download. The contents of the zip file can be modified using the modular structure described in the previous section.

Because the Mask R-CNN model and segmentation model are trained on different version of Pytorch, they are not compatible and cannot be run in the same demo.


ChemIE

Results

Figure 5-2: *Web Demo Results Page*

Recently, a version of Mask R-CNN has been released for PyTorch 1.0. Demo pipelines that includ both of these modules are left for future work.

5.3 Conclusion

In this thesis, I built a pipeline for recognizing molecules in chemistry literature. While the molecule extraction model did not perform well in pixel level segmentation as hypothesized, the bounding boxes were precise enough to be an alternative. With some post-processing the extracted molecuels could be fed to the SMILES translation model. Domain adaptation of the SMILES translation model led to clear improvement on different style drawings over models without domain adaptation. Though the accuracy still needs to be increased to be reliable enough for information extraction, this model marks a substantial improvement over preious methods. Finally, I created an end-to-end pipeline that allows a user to submit a PDF for information extraction. This pipeline runs quickly, taking less than a second to perform text and image based extraction per PDF page.

5.3.1 Future Work

While this pipeline serves as a good starting benchmark for deep learning based optical molecule recognition, there is much room for improvement in each step. Errors made in converting the PDF to diagrams propogate to the segmentation model which then propogate to the translation model etc. Small improvements in early parts of the pipeline can have a snowball effect, resulting in much better final results. In this section I propose potential avenues of research that would improve the model.

The segmentation model can be improved in many ways. First, unlike the translation model, the segmentation model was not trained on datasets of different styles. Simply augmenting the dataset by populating some generated diagrams with molecules extracted from real data, as opposed to the automatically generated molecules, could have a noticeable effect on the results on real literature. In addition to data augmentation, the layers in Mask R-CNN that identify the pixel level mask could be improved to resemble segmentation models specifically designed for fine-grained images. By creating a model that could identify exact masks, we could solve the problem of captions or unwanted identifiers being present in the bounding boxes.

Improving the translation model would rely heavily on data augmentation. The model can only identify tokens that exist in the training set vocabulary. Its failures often occured on data examples with uncommon atoms in. However, the model has been proven to have high accuracy on molecules that contain features that exist in the training set. Augmenting the data would require more human analysis of symbols in chemistry literature that could be added to the training set.

The full pipeline could be explored in many ways. The most ambitious improvement would be to create a single end-to-end neural network that locates molecules and subsequently translates them to their SMILES strings. One benefit of a joint model is that errors in the SMILES strings can provide feedback to both modules - segmentation and translation. Both Mask R-CNN and OpenNMT are large complex models, so building and training a joint version would certainly be challenging, though not impossible.

Appendix A

Model Configurations

A.1 Segmentation

The open source Mask R-CNN library allows customization of the model architecture. Our optimal model hyper-parameters are described here.

Architecture:

Convolution Body: Resnet 101 Feature Pyramid

Number of Regions of Interest: 2000

Bounding Box Layers:

Pool Resolution: 7

Pool Scales: 0.25, 0.125, 0.0625, 0.03125

Pool Sampling Ratio: 2

Mask Layers:

Pool Scales: 0.25, 0.125, 0.0625, 0.03125

Pool Resolution: 14

Pool Sampling Ratio: 2

Resolution: 28

A.2 Translation

Our model was built using the OpenNMT library [10] with the following architecture.

The convolutional encoder has six 2D-convolutional layers, each with stride 1, kernel size of 3, and 1 unit of padding. The number of filters for each layer, in order, is [64, 128, 256, 256, 512, 512]. After each convolutional layer is a 2D-maxpool layer with kernel size of 2 and stride of 2. The recurrent encoder is a bidirectional LSTM cell with 64 features in the hidden state. The attention-based rnn decoder has 128 hidden states, uses Luong attention [14], a scaled-dot product [24] alignment score function, and softmax global attention function.

Appendix B

Figures

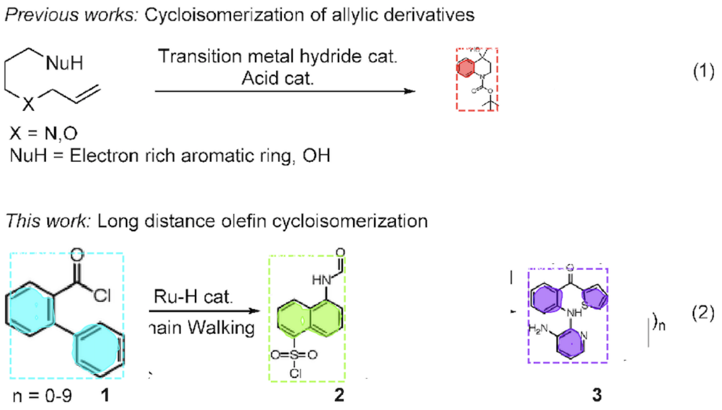

Previous works: Cycloisomerization of allylic derivatives
NuH
X
Transition metal hydride cat.
Acid cat.
(1)
X = N,O
NuH = Electron rich aromatic ring, OH
This work: Long distance olefin cycloisomerization
O
Cl
Ru-H cat.
ıain Walking
HN
O
O=S=O
Cl
NH
H_2N
N
S
)n
(2)
n = 0-9
1
2
3

Figure B-1: *Mask R-CNN Output*

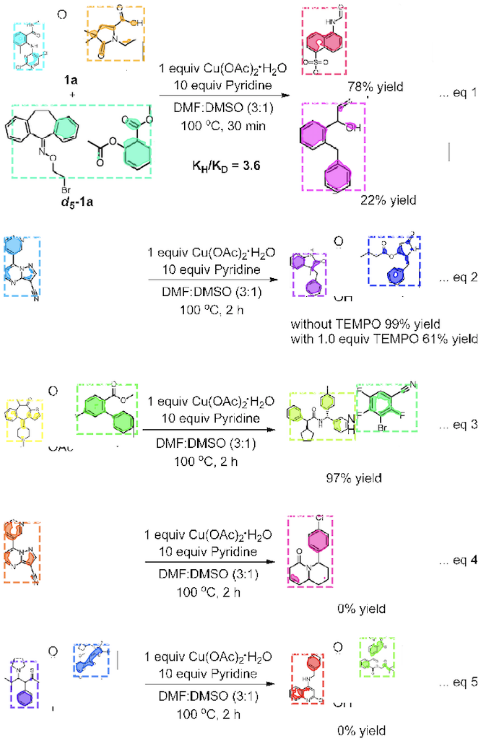


Figure B-2: *Mask R-CNN Output*

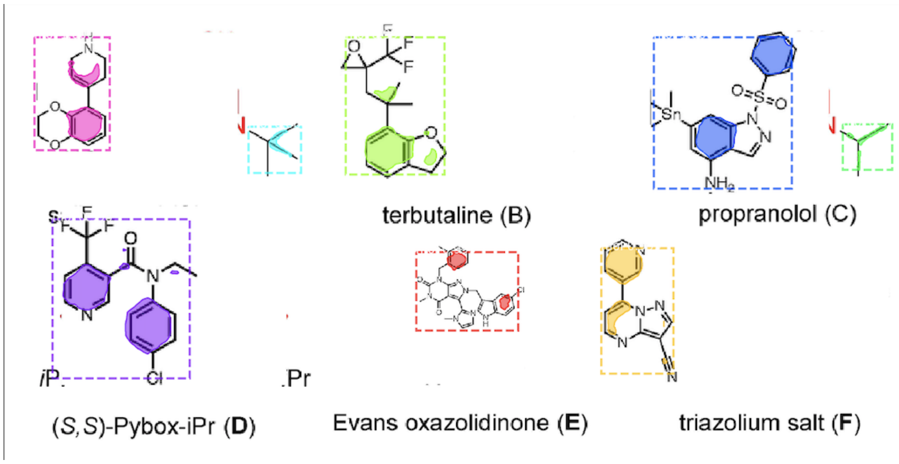


Figure B-3: *Mask R-CNN Output*

Bibliography

[1] Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, and Mario Marchand. Domain-adversarial neural networks. *arXiv preprint arXiv:1412.4446*, 2014.

[2] M. Algorri, M. Zimmermann, C. M. Friedrich, S. Akle, and M. Hofmann-Apitius. Reconstruction of chemical molecules from images. In *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4609–4612, Aug 2007.

[3] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 137–144. MIT Press, 2007.

[4] Igor V Filippov and Marc C Nicklaus. Optical structure recognition software to recover chemical information: Osra, an open source solution, 2009.

[5] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[6] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):142–158, 2015.

[7] Ross B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015.

[8] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017.

[10] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[12] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[13] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015.

[14] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. *CoRR*, abs/1508.04025, 2015.

[15] Joe R McDaniel and Jason R Balmuth. Kekule: Ocr-optical chemical (structure) recognition. *Journal of chemical information and computer sciences*, 32(4):373–378, 1992.

[16] Jungkap et al. Park. Automated extraction of chemical structure information from digital raster images. *Chemistry Central journal*, 34, 2009.

[17] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[18] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[19] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[20] Joshua Staker, Kyle Marshall, Robert Abel, and Carolyn M. McQuaw. Molecular structure extraction from documents using deep learning. *Journal of Chemical Information and Modeling*, 59(3):1017–1029, 2019. PMID: 30758950.

[21] Matthew C. Swain and Jacqueline M. Cole. Chemdataextractor: A toolkit for automated extraction of chemical information from the scientific literature. *Journal of Chemical Information and Modeling*, 56(10):1894–1904, 2016. PMID: 27669338.

[22] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4068–4076, 2015.

[23] Aniko T. Valko and A. Peter Johnson. Clide pro: The latest generation of clide, a tool for optical chemical structure recognition. *Journal of Chemical Information and Modeling*, 49(4):780–787, 2009.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.