Deep Learning Multimodal Extraction of Reaction Data

by

Alex Wang

S.B. Computer Science and Engineering
Massachusetts Institute of Technology, 2024

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

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2024

Authored by:	Alex Wang Department of Electrical Engineering and Computer Science August 14, 2024
Certified by:	Regina Barzilay Professor of Electrical Engineering and Computer Science, Thesis Supervisor
Accepted by:	Katrina LaCurts Chair Master of Engineering Thesis Committee

Deep Learning Multimodal Extraction of Reaction Data

by

Alex Wang

Submitted to the Department of Electrical Engineering and Computer Science
on August 14, 2024 in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

ABSTRACT

Automated extraction of structured information from chemistry literature is vital for maintaining up-to-date databases for use in data-driven chemistry. However, comprehensive extractions require reasoning across multiple modalities and the flexibility to generalize across different styles of articles. Our work on OpenChemIE presents a multimodal system that reasons across text, tables, and figures to parse reaction data. In particular, our system is able to infer structures in substrate scope diagrams as well as align reactions with their metadata defined elsewhere. In addition, we explore the chemistry information extraction potential of Vision Language Models (VLM), which allow powerful large language models to leverage visual understanding. Our findings indicate that VLMs still require additional work in order to meet the performance of our bespoke models.

Thesis supervisor: Regina Barzilay
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

First, I would like to thank my thesis supervisor Regina Barzilay for the opportunity to work on this project and for all her guidance and advice throughout my M.Eng. I am forever thankful for the experience of working in such an amazing lab.

I am deeply grateful to Yujie Qian for taking the chance to bring me on as a UROP, and whose continued mentorship throughout the past few years has been invaluable. I would also like to thank Vincent Fan for his help and support as we worked together through the UROP and M.Eng.

Thank you to my friends for making this such a wonderful time at MIT. I will treasure the countless memories we've created together.

And finally, thank you to my parents, who have supported me through every step of the way. Your belief in me and endless encouragement have been the foundation of my achievements. I couldn't have done this without you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Extracting structured information from chemistry literature is vital for modern data-driven chemistry. Currently, these datasets are manually curated by experts from journal articles and patents, such as in Reaxys [1, 2], but the rapid rate at which new publications are released poses a daunting challenge for keeping these databases up to date. Furthermore, chemistry machine learning models require increasingly comprehensive data as models grow in nuance and complexity [3–6]. Existing approaches for automated information extraction provide only partial solutions due to only addressing individual subproblems, such as extracting reactions from only text or from segmented diagrams [7–10].

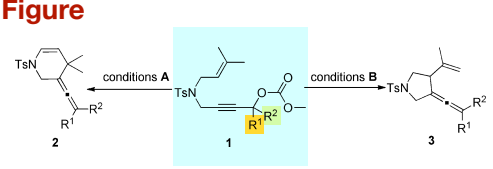
Extracting comprehensive reaction data presents a challenge due to many reactions being specified across multiple modalities, often requiring chemical reasoning to fully determine relevant molecular structures. We show an example of the possible difficulties in Figure 1.1. For one, the figure does not fully specify the molecular structures, as they contain R-groups, which are abbreviations serving as a placeholder for another structure. The defined R-groups can be found by parsing the entries of the accompanying table, or by finding the differences in the molecular graphs of **1** and **1u** in entry 21 for column R^2. For another, there is additional reaction information that must be aligned to the corresponding structures. In Figure 1.1, the footnote text contains important reaction conditions, but highlighted label **1** in the is

referencing a molecule defined in the diagram. In other cases, figures or tables could also contain the condition details, or the reaction itself could be specified within the text.

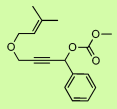
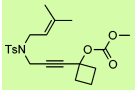
To address these challenges, we develop OpenChemIE, a system for document-level extraction that combines information across individual modalities to recover implicitly defined reactions. We integrate models for text reaction extraction[8], molecule recognition[11], and reaction diagram parsing[7] as well as develop additional modules for multimodal analysis to create a comprehensive reaction extraction system. In particular, we implement modules to resolve R-group structures and to align reactions with reaction metadata presented in tables, annotated in figures, or discussed in texts. In OpenChemIE, our approach to R-group resolution involves identifying and substituting the substructures listed in tables and text or inferred from substrate scope diagrams. Evaluating our system on our curated real world datasets, we find that we achieve an F1 score of 69.5% when extracting from challenging substrate scope diagrams, which require inferring implicit molecular structures. Our system further achieves a precision of 64.3% when comparing our predictions to reactions found in Reaxys.

We additionally investigate the capabilities of vision language models (VLMs) in the context of chemistry information extraction. We finetune a state-of-the-art VLM on reaction diagram parsing and molecule structure recognition. For reaction parsing, our model attains an F1 score of 35.5%, and for molecule recognition, we see that the model struggles to identify atoms and learn meaningful graph structures. Our findings show that significant additional work may be necessary in order for VLMs to match the current performance of our bespoke OpenChemIE models.

Figure



Table

Entry	1	R^1	R^2	Yield [%]	
				2[a]	3[b]
1	1a	H	Ph	75	96
2	1b	H	*p*-ClC_6H_4	66	55
3	1c	H	*m*-ClC_6H_4	64	79
4	1d	H	*o*-ClC_6H_4	63	67
...					
20	1t			-[c]	-[c]
21	1u			-[d]	-[c]

Text

[a]Condition A: The reaction was carried out by using **1** (0.2 mmol), PdI_2 (10 mol %), P(2-furyl)$_3$ (40 mol %), Et_3N (2 equiv), and TBHP (2 equiv) in DMF (2 mL) at 85 °C under an argon atmosphere.

[b]Condition B: The reaction was carried out by using **1** (0.2 mmol), $Pd(dba)_2$ (10 mol %), P(2-furyl)$_3$ (10 mol %), and KOAc (2 equiv) in DMF (2 mL) at 85 °C under an argon atmosphere.

Figure 1.1: Example of a multimodal reaction description from Zhao et al.[12] highlighting connections across modalities.

Chapter 2

Related Work

2.1 Extracting From Figures

Useful tasks for figure extraction include molecule recognition and reaction extraction. Molecule recognition involves converting images of molecules into machine-readable SMILES strings. Early methods for determining the structures of molecules relied on rule-based techniques, using various algorithms and heuristics to identify bonds and atoms, with which the structure can be recovered [13, 14]. Later advancements utilized CNN-based encoder-decoder architectures from deep learning, which improved robustness across diverse styles [11, 15–19]. More recent studies have focused on extracting reaction schemes from figures, identifying the reactants, products, conditions, and yield for each reaction. These approaches either used a combination of heuristics and image filters[20] or employed deep-learning models for object detection and sequence generation[7, 21].

Furthermore, several studies have also presented systems for automatically extracting infromation from figures in real-world documents. These works include ReactionDataExtractor [20] and its updated version ReactionDataExtractor 2.0 [21], which parse reaction diagrams, as well as MolMiner [22] and DECIMER.ai [23], which are designed for molecule recognition. Another system for molecule recognition is ChemSchematicResolver[9], which

adds functionality for resolving R-groups defined in text labels. Despite these advancements, creating a robust figure-based extraction system is still challenging due to the diverse styles and complexities of molecules and reaction schemes.

2.2 Extracting From Text

Relevant tasks for mining from text include chemical entity and role identification, as well as parsing reaction details. For the former, a few past studies have centered on dataset curation[24, 25]. A proposed solution for this task involves creating a parser using regular expressions and classifiers to indentify key terms [26]. Some works have created systems for extracting from PDF files of documents instead of from plain text, including ChemDataExtractor[10] and PDFDataExtractor[27]. For the latter task, solutions include a parsing approach involving grammars and dictionaries [28] as well as a deep-learning model that instead approaches it from a sequence labeling perspective and uses a fine-tuned transformer encoder architecture [8].

2.3 Document Understanding with VLMs

Vision language models have made significant strides in their capabilities since their recent inception. These models augment a base large language model with a vision encoder, often making use of an additional cross-modal fusion module to transform the vision tokens to be in the same space as the language tokens [29–33]. The effect of this is that the language model decoder is able to reason about and between both the image prompt and the text prompt in a unified manner. The flexible manner of inputs, due to being able to ask free-form queries about arbitrary images, presents vast potential for nuanced reasoning and extraction.

One area of interest for VLMs is in visual document understanding. This involves obtaining structured information from documents, which can have a wide range of styles and formatting. A common form of benchmark for VLMs is Visual Question Answering (VQA)

[34], which involves answering user text queries based on the image input. Document understanding can be seen as a more specific form of VQA task, and certain works have modified VLMs to specialize in this. One such work is mPLUG-DocOwl [35], which performs instruction tuning on the visual language model mPLUG-Owl [36] in order to align it with the task of extracting from documents. For chemistry-specific information extraction, the authors of [37] investigate applying VLMs for electrosynthesis reaction diagram parsing, which involves identifying the anode, cathode, electrolytes, solvents, and other reaction parameters. In contrast, we aim to parse general more commonly seen reaction diagrams.

Chapter 3

Overview

In this section, we detail the contributions made to the OpenChemIE system. Namely, we discuss the added modules for multimodal integration, as well as the investigations of the viability of vision language models for certain subtasks of OpenChemIE.

3.1 OpenChemIE

We first provide an overview of the OpenChemIE system. The primary goal of OpenChemIE is to extract a comprehensive list of reaction data from chemistry literature, which could for example be used for downstream training of other models. Figure 3.1 depicts the inputs to the system and the overall flow of the pipeline in order to achieve the final complete reaction list. Prior works introduced the modules for reaction diagram parsing, molecule recognition, and text reaction extraction [7, 8, 11]. Our contribution in OpenChemIE was integrating the set of models into a practical system and developing multimodal modules for further combining the information from these models. Specifically, OpenChemIE aims to integrate information from across figures, texts, and tables, recovering data that a single modality alone would not contain.

At a high level, each modality is initially processed separately using bespoke modules, which are data-driven in the case of figures and texts and rule-based for tables. From figures,

we extract the depicted reactions, the individual molecular structures, and the coreferences between molecules and their identifiers. From text, we parse the described reactions as well as the categories of any chemical entities mentioned. And for tables, we obtain the headers and column data, and we further categorize the columns based on their headers. The resulting data is then passed into our modules for reaction condition alignment and R-group resolution, which we discuss further in the following subsections.

3.1.1 Reaction Condition Alignment

Chemistry literature often pairs figures with associated condition screening tables containing variations in the experimental setup, such as different reaction conditions and R-groups. The corresponding yields for running the reaction under each setup are also recorded in the table. This information must be aligned with the extracted reactions in order to obtain complete substrate data.

For this task, we developed a parser for extracting the table headers and entries. We then categorize each column based on its header, such as being for temperature, solvent, yield, by constructing a dictionary of common forms for these headers. For each row in the table, we can infer a full specification of conditions for the reaction, which we add to the set of extracted reaction conditions.

The accompanying text can also contain descriptions and details of reactions. However, due to the constraints of this modality, reactants and products are often referred to by their unique coreference identifier, with the structural information of the molecules and the association with their identifier defined elsewhere in figures (see Figure). These text-based reactions would be incomplete without machine-readable molecular structures for the extracted reactants and products. Our approach for resolving this issue is to align the molecular structures defined in figures with their identifiers in text. We first associate the identifiers with their structures using our deep-learning model for molecule coreference identification, MolCoref. Then, whenever we parse an identifier while extracting reactions from text, we substitute

the identifier with the associated SMILES representation of their molecular structure. These integrations result in more detailed reaction data than extracting from any one modality alone.

3.1.2 R-Group Resolution

Previous work from ChemSchematicResolver [9] aimed to parse R-groups explicitly expressed as text chemical formulae within figures (e.g., "R=Me") and substitute the definition into the corresponding machine-readable molecule structure. In addition to this case, we seek to comprehend R-groups defined within substrate scope diagrams, which depict a general reaction consisting of template molecules, for which there is an array of different substrates to substitute into the templates, each resulting in different yields. These substrates can be defined by tables separate from the reaction scheme that explicitly express the R-groups, or by figures where products are depicted as different molecular structures. Both of these cases necessitate additional reasoning to fully determine molecular structures.

We address these two forms of substrate scope in OpenChemIE. For the former, where the R-groups are defined as text formulae in labels or a corresponding table, we parse the text for the R-group information and use MolScribe[11] to extract the template molecules in the form of graph structures. We then replace the placeholders in the graphs with the parsed R-group chemical formulae. Finally, these resolved graph structures are converted to SMILES strings by postprocessing methods from MolScribe.

For the latter case, the reaction template is accompanied by a set of possible products depicted in the same diagram. These products are used to imply the structures of the R-groups and the reactants for the various substrates experimented with, and so extracting the complete reactions requires reasoning backward from the product and template. In OpenChemIE, we associate the molecules and their identifiers using MolCoref, and then use the label prefixes (e.g. "6" as the prefix of "6a") to match the products with their template molecule, as the reaction template could potentially have multiple products. With

the graph structures of the template molecule and specific product, we then use a subgraph isomorphism algorithm from RDKit [38] to map the shared atoms between the two molecular structures. We can then determine the R-group substructures by taking the unmapped atoms from the specific product, as these remnants are what replaced the placeholder symbols in the original structure. Having obtained the structures of the R-groups, we make the substitution in the reactant templates as well, thereby determining the full structures for all participating molecules in the reaction.

3.2 Diagram Parsing with VLMs

Another goal of our investigation was to determine whether the visual reasoning skills of vision language models could be leveraged for chemistry information extraction in a more unified manner. For one, this would reduce the need to design bespoke architectures for specific extraction tasks, and another potential benefit would be to reduce the components necessary in our current OpenChemIE system.

Here we make the first steps of exploring VLMs in the context of reaction extraction from figures. We divide this task into two stages as in OpenChemIE. We first parse the high level reaction details from the diagram, namely the reactants, conditions, and products for each depicted reaction. Afterwards, we extract the machine-readable SMILES representation of the molecule structure from each identified reactant and product. In OpenChemIE, these stages would be accomplished by RxnScribe and MolScribe respectively. We provide more detail for these stages in the following subsections.

3.2.1 Reaction Diagram Parsing

Extracting reactions from figures in chemistry literature involves identifying the reactions, locating the individual entities involved in each reaction, and determining whether each entity is a reactant, condition, or product. We formulate this as a sequence-to-sequence

prediction task. The input image and pre-set prompt are both tokenized, which are then passed to the vision language model to generate a machine-readable response string. This output is then parsed for the reaction data. The predictions of our model are in the form of a JSON string containing the bounding boxes for each reaction. Due to the pre-training of the VLM used, the bounding boxes are in the form `[[x1, y1, x2, y2]]`. More specifically, we structure our output as follows

```
[
    {
        "reactants":  [[[x1, y1, x2, y2]], ... ],
        "conditions": [[[x1, y1, x2, y2]], ... ],
        "products":   [[[x1, y1, x2, y2]], ... ]
    },
    ...
]
```

3.2.2 Molecule Recognition

Once the bounding boxes of molecular entities have been identified, we must still translate the images of the molecules into a machine-readable form. A common structure for this are SMILES strings, which encode both the structure and chemical formula of the molecule into a condensed, human-readable string.

Similar to reaction diagram parsing, we frame this as another sequence-to-sequence prediction task. This time, the output is structured in two parts. For the first part, we predict the locations and labels of the atoms. Then for the second, we predict the graph structure of the bonds connecting the atoms. We choose to format the output in this way for a few reasons. Predicting the atoms and then the bonds separately is the approach taken in MolScribe [11], which achieved state of the art performance on this task. This also relaxes

what the model must learn to accomplish, in that it does not need to identify both the correct atom and molecular structure simultaneously when initially extracting the atoms, instead deferring the structure to a later prediction. In MolScribe, bonds are determined by using a feedforward network to predict the bond type for every pair of identified atoms, but in a vision language model, this approach would be prohibitively computationally expensive. In contrast, we use the same model for predicting both atoms and bonds, and we only produce a single output containing and atom and bond information.

Concretely, our model predictions are in the following form

```
{A1 x y} {A2 x y} ...
Single Bonds: (i, j), ...
Double Bonds: (i, j), ...
Triple Bonds: (i, j), ...
Aromatic: (i, j), ...
Wedge: (i, j), ...
```

The first line is a list of each atom and its coordinates. The following lines list the pairs of atoms by their indices with the given bond type. We format the bonds in such a way in order to compress the amount of tokens required to represent the output to be within the maximum context length for most molecules within the dataset.

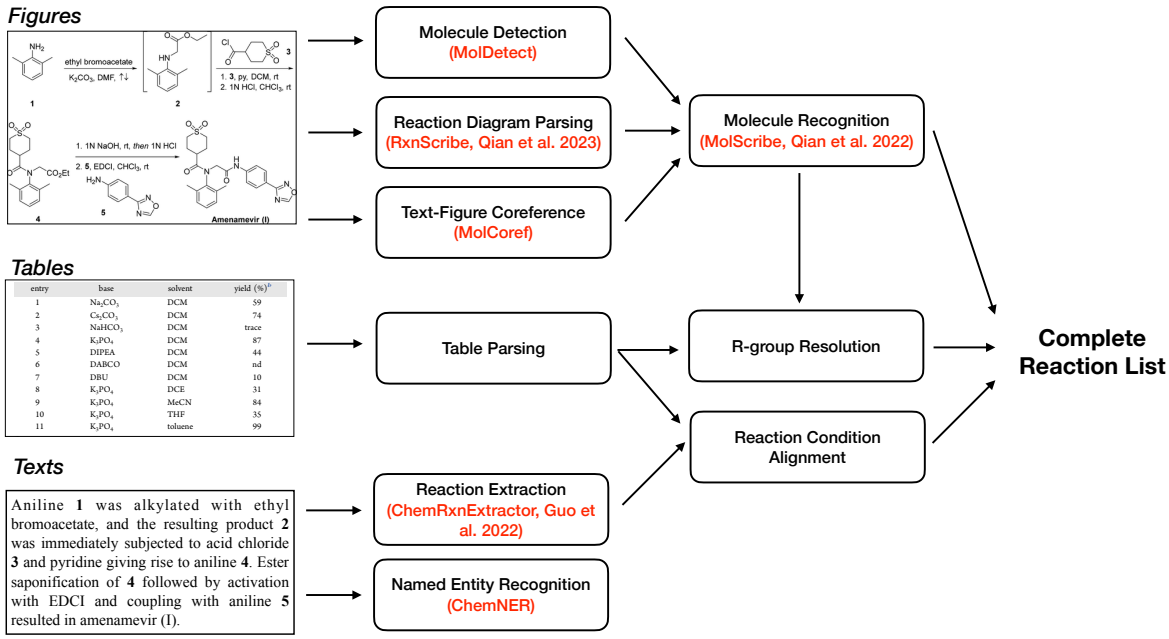


Figure 3.1: Overview of the OpenChemIE system.

Chapter 4

Methods

In this section, we detail the methodology for evaluating OpenChemIE and our training details for the vision language model investigations.

4.1 Substrate Scope Dataset

To assess the effectiveness of OpenChemIE, we curated a dataset of 78 substrate scope diagrams collected from recent articles of various chemistry journals, namely the Journal of Organic Chemistry, Organic Letters, Angewandte Chemie International Edition, European Journal of Organic Chemistry, and Asian Journal of Organic Chemistry. Extraction from substrate scope is a challenging task that requires the use of most of the components in OpenChemIE, making it a suitable task for system evaluation.

We label each diagram with the complete set of fully-specified reactions. For this, we first annotate the structures of the reactants and products in the template reaction using ChemDraw. Often, these molecules will contain R-groups, which must be substituted with a concrete structure in order to be fully specified. As such, given the template and the products in the substrate scope, we then determine the R-group structures and infer the corresponding reactant structures by hand. We then generate the canonical SMILES strings for these inferred reactants and specified products to form the target labels. Canonicalization

results in unique strings for each structure, allowing for comparison with our OpenChemIE predictions.

4.2 Reaxys Dataset

We additionally evaluate OpenChemIE against a curated test dataset of reactions found in Reaxys, a commercial database of reactions manually extracted by chemistry experts and updated regularly. We use this as another measure for the completeness of the extracted reactions from OpenChemIE. Manual extraction as in Reaxys is costly and time consuming, which means keeping a database of reactions up to date manually will require delays to process new papers. Automated systems would allow for some alleviation in workloads, as well as quicker turnaround times.

We collected 19 journal articles, which in total contain 155 figures, from issues of the Journal of Organic Chemistry and Organic Letters. We selected for articles in which a subset of the figures were of substrate scope diagrams so that the entire system would be evaluated as well. To form the dataset, we take the reactions for each article found in Reaxys, and then canonicalize the SMILES strings for each participating molecular structure.

4.3 Reaction Extraction Dataset

For training our reaction extraction vision language model, we use the same dataset that was used in training RxnScribe [7]. This dataset contains 1,378 diagrams collected over 662 articles from the Journal of Organic Chemistry, Journal of the American Chemical Society, Organic Letters, and Organic Process Research & Development. The labels for these diagrams consist of a list of reactions present, each with the bounding boxes of the reactants, conditions, and products present.

We filter the original dataset for only bounding boxes of molecules (i.e. excluding text), except when either the filtered reactants or products are empty, in which case we take the

	Single-line	Multiple-line	Tree	Graph	Overall
Number of Diagrams	730	260	286	102	1378
Number of Reactions	882	948	1313	633	3776

Table 4.1: Breakdown of RxnScribe [7] Dataset

original set. This is for the sake of having the tokenized training data fit in the maximum context length of our model. As noted in [7], the labels for bounding boxes of text can be ambiguous and are disregarded in a similar manner for their soft match evaluation.

Notably, the diagrams depicted in this dataset fall into four different categories of different difficulty levels. The simplest are single-line diagrams, which depict all reactions on a single line. In multiple-line diagrams, the reactions still proceed linearly but can span multiple lines going in reading order (from left to right and then top to down). For tree diagrams, there are branching possibilities for the reactions to proceed. Finally, for graph diagrams, the sequence of reactions can be cyclic. We show the number of reactions for each of these categories in Table 4.1.

We additionally make use of data augmentations in order to alleviate the relatively small size of this dataset. For one, we compositionally augment the images by stacking two random diagrams vertically with a random horizontal offset. Furthermore, we randomly rotate the augmented images in 90 degree increments. We split the diagrams in this dataset into training and test sets in a 9 to 1 ratio.

4.4 Molecule Recognition Dataset

In order to train our model on molecule recognition, we make use of the dataset curated for MolScribe [11]. This data is composed of 1 million synthetic samples and 680k images from patents. The synthetic data is generated by rendering molecules from the PubChem database [39] and rendered with the Indigo toolkit [40]. Ground truth data for atoms, bond types, and labels are available as a result of the rendering. The patent data is sourced from the United States Patent and Trademark Office [41], which includes MOLfile data that can

be used to obtain ground truth labels for atoms and bonds.

4.5 Model

For our base model, we make use of the CogAgent vision language model [32]. There are a few reasons for this decision. For one, the authors of [32] report state-of-the-art performance across a wide range of Visual Question Answering (VQA) tasks. The grounding capabilities available after pre-training allow for decent object detection performance on general images. For another, CogAgent improves on its previous iteration CogVLM [34] to allow for higher resolution image inputs (1120×1120 compared to 224×224). For more complicated diagrams, a higher resolution is necessary in order to determine the details of individual molecules and for optical character recognition of text.

The architecture of the model consists of a vision encoder, a vision language model decoder, an MLP adapter to adjust the feature space of the vision encoder, and a cross attention module to allow for high-resolution inputs. The vision encoder used is EVA2-Clip [42] and the VLM decoder is Vicuna-1.5-7B [43] augmented with a visual expert module.

4.6 Training

We use Low-Rank Adaptation [44], or LoRA, in order to finetune the base model for our chemistry information extraction tasks using a feasible amount of computation resources. LoRA leverages the observation that the changes required for finetuning are often of low rank. By representing these changes through low-rank factorization, LoRA reduces the number of parameters that need to be updated, thereby lowering both the computational cost and memory footprint without compromising model performance. Given that the object detection abilities are present in the pre-trained model, we claim that the new tasks do not deviate enough from the pre-trained corpus to where LoRA finetuning is not sufficient.

For both our finetuning tasks, we train with rank 96 update matrices for 25,000 iterations

on 8 A6000 GPUs. We use a batch size of 32 with a learning rate of 1e-5 and Adam optimizer. Our learning rate follows a cosine decay with a warmup of 1% of the total iterations. For the reaction extraction, we train on an easier subset of the dataset for an initial 25,000 iterations and then on the full dataset for another 25,000 iterations.

Chapter 5

Experiments

In this section, we will cover the experiments and results first for the evaluation of our OpenChemIE system, and then for our VLM investigations.

5.1 OpenChemIE System Evaluation

We evaluate OpenChemIE on our two curated datasets, namely the substrate scope dataset and the Reaxys dataset.

5.1.1 Substrate Scope Evaluation

For our evaluation, we use a hard match in order to determine whether a prediction is a hit or a miss. Specifically, we compare the canonical SMILES for every reactant and product in the prediction to the canonical SMILES of the ground truth reactions. We only consider the two to match if and only if there is an exact match between predicted reactants and products and those of the ground truth.

We quantitatively evaluate OpenChemIE by computing the precision, recall, and F1 score. Here, precision refers to the proportion of predictions that are correct, recall refers to the proportion of ground truth predictions that were correctly predicted, and the F1 score is

Dataset	Correct	Num. Pred.	Num. Labels	Prec.	Rec.	F1
Substrate Scope	624	790	1007	79.1%	62.0%	69.5%
Reaxys	257	400	-	64.3%	-	-

Table 5.1: OpenChemIE performance on each evaluation dataset.[1]

the harmonic mean of precision and recall. As shown in Table 3.1, for this task, our system achieves a precision of 79.1%, recall of 62.0% and F1 score of 69.5%.

We further provide a breakdown of the error contributions of each stage in the OpenChemIE pipeline for this task in Table 5.2. We note that many of the individual components of OpenChemIE perform robustly and do not contribute much to the errors for substrate scope extraction. Of particular note is RxnScribe, which in the original paper [7] was shown to achieve a strong F1 score of 91% on single line diagrams, the most common type of template reaction. In this substrate scope evaluation, we see none of the errors that occurred originated from RxnScribe.

We also see strong performance for modules like MolDetect for molecule detection and our R-group resolution algorithm. In particular, when the input is free of errors, the R-group resolution algorithm is able to correctly identify the R-group structures and accordingly perform the necessary substitutions in almost all circumstances. In the few errors that did occur, there was often a limitation in scope or violation of assumptions for our algorithm. For example, in certain diagrams, the template product is symmetrical other than for the R-groups. Since our algorithm does not take into account color or absolute positions, it is unable to consistently determine the correct assignments of R-group structures.

In contrast, the vast majority of our error contributions arise from problems during molecule recognition and coreference resolution. MolScribe specifically accounts for 64% of all errors. The reasons for this are twofold. One is that MolScribe struggles with molecules of diverse styles, as the authors of [11] report a 71.9% accuracy on realistic molecules taken from the American Chemical Society. Another is that an error when extracting the structure

[1]Ground truth counts omitted for Reaxys due to the database including reactions that are not necessarily present in the given paper.

Task	Num. Errors	% Error Contribution
Molecule Recognition	245	64
Coreference Resolution	58	15
Optical Character Recognition	45	12
R-Group Resolution	24	6
Molecule Detection	9	2
Reaction Diagram Parsing	0	0

Table 5.2: Breakdown of error contributions by sub-task for the substrate scope dataset

of a template molecule will propagate through the rest of the pipeline and result in incorrect predictions for all resolved reactions. However, this exceptionally large penalty for potentially a single mistake does not detract from our choice in how to fairly evaluate the system, as it is important to penalize failures of a critical module when so many results depend on it.

5.1.2 Reaxys Evaluation

We further evaluate OpenChemIE by comparing its extractions to those found in Reaxys. We measure the performance of our system using a soft match. For one, certain compounds structurally rearrange to different isomers in solution, a process known as tautomerization, and so the arrangements stored in Reaxys may differ from the explicitly depicted molecule structures. To address this, we determine tautomers for each molecular structure and canonicalize the corresponding SMILES strings for comparison. For another, we consider a predicted reaction to be correct if Reaxys contains a reaction such that the predicted reactants are a subset of the Reaxys reactants and similarly for the products. This evaluation metric helps to alleviate inconsistencies in annotation conventions, such as whether certain entities are considered reactants or are instead reagents specified in the conditions.

As shown in Table 5.1, our system predicts a total of 400 reactions on this dataset, of which 257 have soft matches, yielding a precision of 64.3%. We also compare these results with ReactionDataExtractor 2.0[21] on the same dataset as a benchmark. Since ReactionDataExtractor 2.0 does not take text or tables as input, we manually segment the diagrams for each journal article to pass in as input. Their model extracts 102 reactions,

Model	Precision	Recall	F1 Score
CogAgent	28.0	39.3	32.7
RxnScribe	83.8	76.5	80.0

Table 5.3: Performance of finetuned CogAgent[32] on reaction extraction compared to RxnScribe [7] soft-match performance. Scores presented in %.

of which 9 have soft matches in Reaxys, achieving 8.8% precision. This gap is largely due to being unable to extract reactions described in texts or tables, as well as being unable to resolve the R-group structures present in substrate scope diagrams, which account for a large portion of the reactions present.

5.2 VLM Investigations

In the following subsections, we detail and discuss the results of our experiments with finetuning CogAgent on our selected chemistry information extraction tasks.

5.2.1 Reaction Extraction

After finetuning our model on the training subset of our reaction extraction data, we evaluate it on the remaining holdout data. In order for a predicted reaction to be counted as correct, we must have the predicted bounding boxes for all reactants, conditions, and products match those of one of the ground truth reactions for the given diagram. Two entities are considered to match if they have an Intersection over Union score (IoU) greater than 0.5. We compute the precision, recall, and F1 score with the same definitions as before.

As shown in Table 5.3, our vision language model approach is only able to achieve an F1 score of 32.7%, significantly lower than that of the original RxnScribe [7], which has an overall F1 score of 69.1% for their hard match evaluation. In Table 5.4, we further break down the scores by type of diagram, from which we can observe that the VLM is able to rather consistently handle simpler single-line diagrams. However, it severely decreases in performance once there are more than a few reactions present per diagram.

Diagram Type	Num. Correct	Num. Predictions	Num. Ground Truth
Single-line	40	177	91
Multiple-line	48	138	93
Tree	43	160	147
Graph	23	78	61

Table 5.4: Breakdown of performance of finetuned CogAgent by diagram type. Entries represent the number of reactions.

We illustrate some example predictions of our model in Figure 5.1. We observe a few common forms of error. For one, the model often drops a single bounding box despite correctly predicting the rest of the entities, resulting in the whole reaction being counted as incorrect. The most common error case is when there are many reactions, in which case our model breaks down in a variety of ways, including repeated predictions of the same reaction, incorrectly identifying reactants versus products, and missing participating entities. We also note that in the case of a malformed prediction, we count the number of predicted reactions as the number of reactions in the ground truth, which may account for the lower accuracy.

5.2.2 Molecule Recognition

Following the finetuning of the model on the molecule recognition dataset, we observe that while the generated outputs are of the same form as the training targets, there is little useful information contained in the predictions, which we show examples of in Table 5.5. When first examining the atom predictions, we note that while the general content resembles that of the ground truth molecule, the counts for individual elements are often incorrect. This suggests that the decoder is unable to properly learn to identify occurrences of each atom in the transformed and tokenized input image, which reflects the known limitations of large language models in quantitative tasks like counting.

Furthermore, a look into the predicted bonds shows that the single bonds all follow a consecutively increasing pattern. If one were to construct a molecule with this predicted graph structure, the entire molecule would be a single chain with kinks from higher order

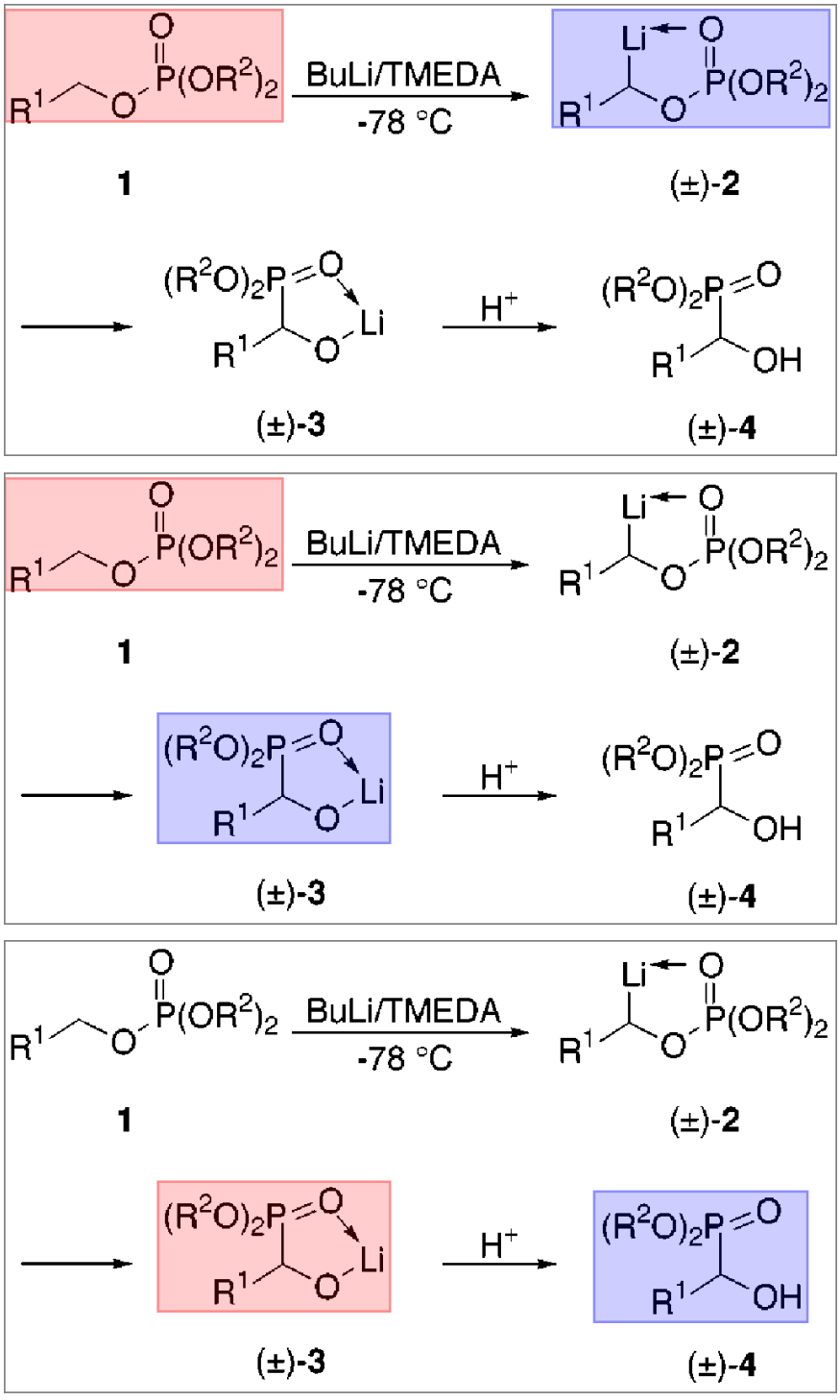


Figure 5.1: Example predictions of finetuned CogAgent where predicted bounding boxes for reactants are in red and products are in blue. We see that for one reaction, the model does not identify the correct reactant, misinterpreting the multi-line diagram.

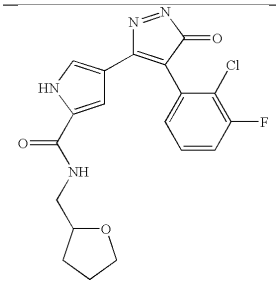
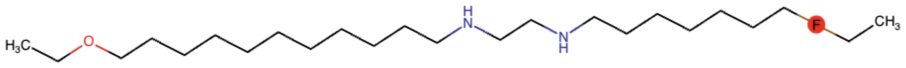
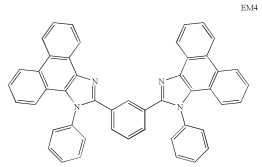
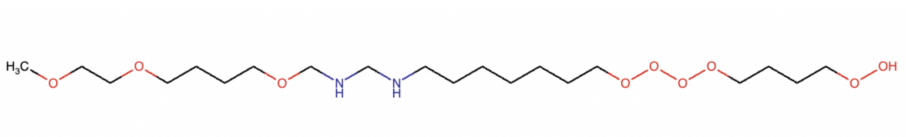
Ground Truth	Predicted Molecule
	
	

Table 5.5: Example predictions of finetuned CogAgent for molecule recognition. We observe that atom identification is often inaccurate, and structure prediction is devoid of meaning.

bonds, which do not make up the majority of the training dataset. This indicates that the language model decoder was unable to learn to associate the pairs of indices with specific locations in the image. We expect this result, given the poor performance of atom detection, since if the initial predicted locations are incorrect, then it is unreasonable to correctly attend to where the bonds are. There is also perhaps a limitation to what the base language model can learn with this input format, as the pattern in the training data makes a prediction of a consecutively increasing sequence for single bonds have seemingly low perplexity.

Chapter 6

Conclusion

Automated chemistry information extraction systems have much potential to offer for cheminformatics and modeling. Our OpenChemIE system has made strides toward multimodal analysis of chemistry literature, allowing for more comprehensive data. While many of our modules achieve robust performance on their respective tasks, some remain to be improved or are limited by their scope. In particular, we found molecule structure recognition to be the main source of errors, as well as being limited to molecules with a well defined SMILES representation. A significant portion of structures depicted in literature fall outside this specification, and addressing these would greatly expand the possible domain of extractions.

Further work is also necessary to address the shortcomings of vision language models. As it stands, our bespoke modules for the corresponding tasks that we investigated significantly outperform that of the finetuned VLMs. However, the language reasoning skills of VLMs should not be discounted, and additional exploration could see whether VLMs offer solutions for a wider breadth of tasks, given their more flexible form of inputs. Overall, there remains ample room for improvement. Addressing the limitations of OpenChemIE and VLMs would be significant steps toward realizing the full potential of the field. Continued advancements in these areas will undoubtedly enhance the comprehensiveness and accuracy of data extraction, ultimately contributing to more effective and insightful cheminformatics research.

References

[1] *Reaxys*. (accessed on 07/01/2023). URL: https://www.reaxys.com.

[2] *SciFinder*. (accessed on 07/01/2023). URL: https://scifinder.cas.org.

[3] Z. Tu, T. Stuyver, and C. W. Coley. "Predictive chemistry: machine learning for reaction deployment, reaction development, and reaction discovery". In: *Chem. Sci.* 14 (2 2023), pp. 226–244. DOI: 10.1039/D2SC05089G. URL: http://dx.doi.org/10.1039/D2SC05089G.

[4] M. R. Maser, A. Y. Cui, S. Ryou, T. J. DeLano, Y. Yue, and S. E. Reisman. "Multilabel Classification Models for the Prediction of Cross-Coupling Reaction Conditions". In: *Journal of Chemical Information and Modeling* 61.1 (2021). PMID: 33417449, pp. 156–166. DOI: 10.1021/acs.jcim.0c01234. eprint: https://doi.org/10.1021/acs.jcim.0c01234. URL: https://doi.org/10.1021/acs.jcim.0c01234.

[5] H. Gao, T. J. Struble, C. W. Coley, Y. Wang, W. H. Green, and K. F. Jensen. "Using Machine Learning To Predict Suitable Conditions for Organic Reactions". In: *ACS Central Science* 4.11 (2018). PMID: 30555898, pp. 1465–1476. DOI: 10.1021/acscentsci.8b00357. eprint: https://doi.org/10.1021/acscentsci.8b00357. URL: https://doi.org/10.1021/acscentsci.8b00357.

[6] Y. Qian, Z. Li, Z. Tu, C. Coley, and R. Barzilay. "Predictive Chemistry Augmented with Text Retrieval". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by H. Bouamor, J. Pino, and K. Bali. Singapore:

Association for Computational Linguistics, Dec. 2023, pp. 12731–12745. URL: https://aclanthology.org/2023.emnlp-main.784.

[7] Y. Qian, J. Guo, Z. Tu, C. W. Coley, and R. Barzilay. "RxnScribe: A Sequence Generation Model for Reaction Diagram Parsing". In: *Journal of Chemical Information and Modeling* 63.13 (2023), pp. 4030–4041. DOI: 10.1021/acs.jcim.3c00439.

[8] J. Guo, A. S. Ibanez-Lopez, H. Gao, V. Quach, C. W. Coley, K. F. Jensen, and R. Barzilay. "Automated Chemical Reaction Extraction from Scientific Literature". In: *J. Chem. Inf. Model.* 62.9 (2022), pp. 2035–2045. DOI: 10.1021/acs.jcim.1c00284. URL: https://doi.org/10.1021/acs.jcim.1c00284.

[9] E. J. Beard and J. M. Cole. "ChemSchematicResolver: A Toolkit to Decode 2D Chemical Diagrams with Labels and R-Groups into Annotated Chemical Named Entities". In: *Journal of Chemical Information and Modeling* 60.4 (2020). PMID: 32212690, pp. 2059–2072. DOI: 10.1021/acs.jcim.0c00042. eprint: https://doi.org/10.1021/acs.jcim.0c00042. URL: https://doi.org/10.1021/acs.jcim.0c00042.

[10] M. C. Swain and J. M. Cole. "ChemDataExtractor: A Toolkit for Automated Extraction of Chemical Information from the Scientific Literature". In: *J. Chem. Inf. Model.* 56.10 (2016), pp. 1894–1904. DOI: 10.1021/acs.jcim.6b00207. URL: https://doi.org/10.1021/acs.jcim.6b00207.

[11] Y. Qian, J. Guo, Z. Tu, Z. Li, C. W. Coley, and R. Barzilay. "MolScribe: Robust Molecular Structure Recognition with Image-to-Graph Generation". In: *Journal of Chemical Information and Modeling* 63.7 (2023), pp. 1925–1934. DOI: 10.1021/acs.jcim.2c01480.

[12] S.-C. Zhao, K.-G. Ji, L. Lu, T. He, A.-X. Zhou, R.-L. Yan, S. Ali, X.-Y. Liu, and Y.-M. Liang. "Palladium-Catalyzed Divergent Reactions of 1,6-Enyne Carbonates: Synthesis of Vinylidenepyridines and Vinylidenepyrrolidines". In: *The Journal of Organic Chemistry* 77.6 (2012). PMID: 22364228, pp. 2763–2772. DOI: 10.1021/jo202590w. eprint: https://doi.org/10.1021/jo202590w. URL: https://doi.org/10.1021/jo202590w.

[13] I. V. Filippov and M. C. Nicklaus. *Optical structure recognition software to recover chemical information: OSRA, an open source solution*. 2009.

[14] T. Peryea, D. Katzel, T. Zhao, N. Southall, and D.-T. Nguyen. "MOLVEC: Open source library for chemical structure recognition". In: *Abstracts of papers of the American Chemical Society*. Vol. 258. 2019.

[15] J. Staker, K. Marshall, R. Abel, and C. M. McQuaw. "Molecular Structure Extraction from Documents Using Deep Learning". In: *J. Chem. Inf. Model.* 59.3 (2019), pp. 1017–1029. DOI: 10.1021/acs.jcim.8b00669. eprint: https://doi.org/10.1021/acs.jcim.8b00669. URL: https://doi.org/10.1021/acs.jcim.8b00669.

[16] K. Rajan, H. O. Brinkhaus, A. Zielesny, and C. Steinbeck. "A review of optical chemical structure recognition tools". In: *J. Cheminf.* 12.1 (2020), pp. 1–13.

[17] M. Oldenhof, A. Arany, Y. Moreau, and J. Simm. "ChemGrapher: optical graph recognition of chemical compounds by deep learning". In: *J. Chem. Inf. Model.* 60.10 (2020), pp. 4506–4517.

[18] D.-A. Clevert, T. Le, R. Winter, and F. Montanari. "Img2Mol – accurate SMILES recognition from molecular graphical depictions". In: *Chem. Sci.* 12 (42 2021), pp. 14174–14181. DOI: 10.1039/D1SC01839F. URL: http://dx.doi.org/10.1039/D1SC01839F.

[19] K. Rajan, A. Zielesny, and C. Steinbeck. "DECIMER 1.0: deep learning for chemical image recognition using transformers". In: *J. Cheminf.* 13.1 (2021), pp. 1–16.

[20] D. M. Wilary and J. M. Cole. "ReactionDataExtractor: A Tool for Automated Extraction of Information from Chemical Reaction Schemes". In: *J. Chem. Inf. Model.* 61.10 (2021), pp. 4962–4974.

[21] D. M. Wilary and J. M. Cole. "ReactionDataExtractor 2.0: A Deep Learning Approach for Data Extraction from Chemical Reaction Schemes". In: *Journal of Chemical Information and Modeling* 63.19 (2023). PMID: 37729111, pp. 6053–6067. DOI:

10.1021/acs.jcim.3c00422. eprint: https://doi.org/10.1021/acs.jcim.3c00422. URL: https://doi.org/10.1021/acs.jcim.3c00422.

[22] Y. Xu et al. "MolMiner: You Only Look Once for Chemical Structure Recognition". In: *J. Chem. Inf. Model.* 62.22 (2022), pp. 5321–5328. DOI: 10.1021/acs.jcim.2c00733. URL: https://doi.org/10.1021/acs.jcim.2c00733.

[23] K. Rajan, H. O. Brinkhaus, M. I. Agea, A. Zielesny, and C. Steinbeck. "DECIMER.ai - An open platform for automated optical chemical structure identification, segmentation and recognition in scientific publications". In: (2023).

[24] M. Krallinger, F. Leitner, O. Rabal, M. Vazquez, J. Oyarzabal, and A. Valencia. "CHEMDNER: The drugs and chemical names extraction challenge". In: *J. Cheminformatics* 7.S-1 (2015), S1. DOI: 10.1186/1758-2946-7-S1-S1. URL: https://doi.org/10.1186/1758-2946-7-S1-S1.

[25] D. Q. Nguyen et al. "ChEMU: Named Entity Recognition and Event Extraction of Chemical Reactions from Patents". In: *Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14-17, 2020, Proceedings, Part II*. Vol. 12036. Lecture Notes in Computer Science. Springer, 2020, pp. 572–579. DOI: 10.1007/978-3-030-45442-5_74. URL: https://doi.org/10.1007/978-3-030-45442-5%5C_74.

[26] L. Hawizy, D. M. Jessop, N. Adams, and P. Murray-Rust. "ChemicalTagger: A tool for semantic text-mining in chemistry". In: *J. Cheminformatics* 3 (2011), p. 17. DOI: 10.1186/1758-2946-3-17. URL: https://doi.org/10.1186/1758-2946-3-17.

[27] M. Zhu and J. M. Cole. "PDFDataExtractor: A Tool for Reading Scientific Text and Interpreting Metadata from the Typeset Literature in the Portable Document Format". In: *J. Chem. Inf. Model.* 62.7 (2022), pp. 1633–1643. DOI: 10.1021/acs.jcim.1c01198. URL: https://doi.org/10.1021/acs.jcim.1c01198.

[28] D. M. Jessop, S. E. Adams, E. L. Willighagen, L. Hawizy, and P. Murray-Rust. "OSCAR4: a flexible architecture for chemical text-mining". In: *J. Cheminformatics* 3 (2011), p. 41. DOI: 10.1186/1758-2946-3-41. URL: https://doi.org/10.1186/1758-2946-3-41.

[29] J.-B. Alayrac et al. *Flamingo: a Visual Language Model for Few-Shot Learning*. 2022. arXiv: 2204.14198 [cs.CV].

[30] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny. *MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models*. 2023. arXiv: 2304.10592 [cs.CV].

[31] H. Liu, C. Li, Y. Li, and Y. J. Lee. *Improved Baselines with Visual Instruction Tuning*. 2024. arXiv: 2310.03744 [cs.CV]. URL: https://arxiv.org/abs/2310.03744.

[32] W. Hong et al. *CogAgent: A Visual Language Model for GUI Agents*. 2023. arXiv: 2312.08914 [cs.CV]. URL: https://arxiv.org/abs/2312.08914.

[33] L. Beyer et al. *PaliGemma: A versatile 3B VLM for transfer*. 2024. arXiv: 2407.07726 [cs.CV]. URL: https://arxiv.org/abs/2407.07726.

[34] W. Wang et al. *CogVLM: Visual Expert for Pretrained Language Models*. 2024. arXiv: 2311.03079 [cs.CV]. URL: https://arxiv.org/abs/2311.03079.

[35] J. Ye et al. *mPLUG-DocOwl: Modularized Multimodal Large Language Model for Document Understanding*. 2023. arXiv: 2307.02499 [cs.CL].

[36] Q. Ye et al. *mPLUG-Owl: Modularization Empowers Large Language Models with Multimodality*. 2024. arXiv: 2304.14178 [cs.CL]. URL: https://arxiv.org/abs/2304.14178.

[37] S. X. Leong, S. Pablo-García, Z. Zhang, and A. Aspuru-Guzik. *Automated Electrosynthesis Reaction Mining with Multimodal Large Language Models (MLLMs)*. 2024. URL: https://chemrxiv.org/engage/chemrxiv/article-details/668f151001103d79c5a59f47.

[38] G. Landrum. *RDKit*. https://www.rdkit.org/. 2010.

[39] S. Kim et al. "PubChem Substance and Compound databases". In: *Nucleic Acids Research* 44.D1 (Sept. 2015), pp. D1202–D1213. ISSN: 0305-1048. DOI: 10.1093/nar/gkv951. eprint: https://academic.oup.com/nar/article-pdf/44/D1/D1202/9484096/gkv951.pdf. URL: https://doi.org/10.1093/nar/gkv951.

[40] D. Pavlov, M. Rybalkin, B. Karulin, M. Kozhevnikov, A. Savelyev, and A. Churinov. "Indigo: universal cheminformatics API". In: *Journal of Cheminformatics* (2011). DOI: 10.1186/1758-2946-3-S1-P4. URL: https://jcheminf.biomedcentral.com/articles/10.1186/1758-2946-3-S1-P4.

[41] *United States Patent and Trademark Office*. (accessed on 12/31/2021). URL: https://bulkdata.uspto.gov/.

[42] Q. Sun, Y. Fang, L. Wu, X. Wang, and Y. Cao. *EVA-CLIP: Improved Training Techniques for CLIP at Scale*. 2023. arXiv: 2303.15389 [cs.CV]. URL: https://arxiv.org/abs/2303.15389.

[43] W.-L. Chiang et al. *Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality*. Mar. 2023. URL: https://lmsys.org/blog/2023-03-30-vicuna/.

[44] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. *LoRA: Low-Rank Adaptation of Large Language Models*. 2021. arXiv: 2106.09685 [cs.CL]. URL: https://arxiv.org/abs/2106.09685.