Chemical Reaction Mining

Carl Edwards
cne2@illinois.edu
University of Illinois Urbana-Champaign
Urbana-Champaign, Illinois, USA

ABSTRACT

Many new and exciting chemicals have been discovered. However, the reactions to produce them often produce low reaction yields or require expensive intermediate reactants. These chemicals cannot be used because they are too difficult to create at the large scales required for industrial use. In order to remedy this problem, finding new reactions to produce desired products with improved yields or scalability is very desirable. To tackle this problem, I propose using transformer neural network architectures to augment molecule representations for predicting molecule reactions. In this project, I frame the problem as an information retrieval task, and I achieve a Hits@1 value of 50%. Further, I try to predict the reaction yield, although this does not prove effective using the transformer architecture.

KEYWORDS

molecules, chemical reaction mining, transformer, embeddings

ACM Reference Format:
Carl Edwards. 2021. Chemical Reaction Mining. In *Proceedings of CS 512 (Data Mining Principles).* , 9 pages.

1 INTRODUCTION

Chemicals are ubiquitous in everyday life. Whether enabling the latest breakthroughs and products in technology, medicine, or materials, they are an essential component of modern day life. In order to create chemicals, reactions consisting of other chemicals are required. However, finding reaction routes which are scalable and efficient can be very challenging. For example, a critical drug might be found for some disease, but there is no way to produce it at a large enough scale to be made commercially. To address this issue, it is important to find new reactions to create existing compounds. To do this, I propose the use of transformers to enable this planning. Chemical reactions take "reactants" as input, and they produce "products" as output. Thus, my model model takes embeddings of reactants as input, and it tries to predict a product as output.

Initially, I planned to frame the problem as a link prediction task by turning reaction data into a knowledge graph, such as in [20]. However, I encountered considerable difficulty in accessing this data. Further discussion about datasets is in Section 2.1. Due to these problems with data, at the midterm report I planned to use three non-reaction graph datasets. However, between the midterm report and final submission, I have been able to gain access to the Chemical Reactions from US Patents Dataset [11]. Unfortunately, this dataset has not been amenable to creating a graph, so I have instead taken a transformer-based approach. My proposed model integrates mol2vec [7] embeddings into a transformer architecture to predict the final reaction product.

Data Mining Principles, Spring 2021, Illinois, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00

2 RELATED WORK

In the past, chemists would consider hundreds of possible reactants and conduct many experiments to find new reactions. However, the recent success of data mining and machine learning techniques opens a new avenue to find these reactions. Work such as [14, 22] focus on using computers to plan reactions. [13] focuses on automating reaction network construction from reaction databases. Additionally, recent approaches such as [1, 16, 21] apply deep learning to plan retrosynthetic reactions or drug-drug interactions. In particular, [20] creates a knowledge graph from Reaxys [1] data. The data consists of 14.4 million molecules and 8.2 million binary reactions between them, which consists of the majority of chemical reactions in the literature up to 2013. They use reactions from 2014 and 2015 as a validation dataset and find that they can independently re-discover 35% of challenging novel reactions discoverd in those years (reactions without 'template' reactions from prior work). This work uses a breadth-first search based model which prunes possible reaction paths based on Tanimoto similarity of reaction fingerprints.

Additionally, machine learning has become more popular in cheminformatics. Representations for molecules such as mol2vec [7], MolBert [6], and ChemBerta [4] create embedding representations by leveraging the recent success from natural language processing algorithms. These are applicable to various downstream tasks, such as molecule property prediction [26].

Recent approaches have tackled similar problems to reaction prediction. [17] and [19] use transformers to learn "atom-mappings" between reactions and products. [18], which was published four weeks ago, takes this a step further to learn grammars for these reactions. Unlike this work, I focus on integrating molecule representations together rather than aligning atoms. My approach is less rule-based allowing for more flexibility, but errors are more difficult to diagnose.

2.1 Related Datasets

Unfortunately, the chemistry community is much more closed and proprietary than data mining. There are not standard datasets and many of the best databases are proprietary. I tried to use Reaxys, which is the most complete and commonly used database [20]. UIUC has a license to use Reaxys [2], however, this does not include access to the Reaxys API which is needed in order to query reactions in bulk for building a dataset.

[1] https://www.reaxys.com/
[2] https://www.library.illinois.edu/chx/reaxys/

The PubChem [8] dataset is one of the largest openly available public datasets. It contains information on millions of chemicals and a knowledge graph with links between some of them. Although these links are not necessarily reactions, predicting relations in this case could also be useful. In particular, I extracted three different graph datasets. These are graphs linking compounds with the same connectivity, isotopologues, and compounds to their components. The connectivity describes molecules which are isomers with identical constitutions. The isotopologues are molecules with the same formula and configuration, but they have different numbers of neutrons. Finally, the compound2component dataset links a molecular substance to one of its neutral ions.

The Chemical Reactions from US Patents Dataset [11] consists of 1.9 million reactions extracted from US patents from 1976 to 2016. This work focuses on predicting the products of these reactions. This reaction also contains computer-calculated yield rates for these reactions, which I use as a regression task.

3 PROBLEM DEFINITION

Given a dataset D containing n reactions, the goal is to predict the products (output) of the reactions from the reactants (input). I frame this task as an information retrieval task, which is similar to link prediction in knowledge graphs such as in [2]. To do so, I restrict the data to only contain reactions with one product. Thus, multiple reactants (input) combine together to make one product (output). This can be thought of as a graph with non-binary relations between the nodes. The goal is to predict these non-binary links. The products in the dataset are ranked by their probability of being the reaction output. Thus, the problem can be evaluated using standard metrics such as mean rank, mean reciprocal rank (MRR), and Hits@1. Given a vector of ranks, R, for n reactions:

$$MeanRank = \frac{1}{n}\sum_{i=1}^{n} R_i$$

$$MeanReciprocalRank = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{R_i}$$

$$Hits@m = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{R_i \leq m}$$

This can also be thought of as a many-class classification problem, where ranking the correct product at 1 indicates it is correctly classified.

4 METHODOLOGY

4.1 Morgan Fingerprinting

Molecule fingerprints have been used for a long time in cheminformatics [3]. Typically, they map a substructure to some string of bits in a manner that allows them to be used for quick substructure and chemistry search of molecules. The most common method is Tanimoto similarity, also known as Jaccard coefficient. Another example is the extended-connectivity fingerprint [15], often known as the Morgan fingerprint. This computes a bit string for substructures in the molecule as follows:

(1) Assign each atom an initial identifier. This identifier is obtained by using the Daylight atomic invariants rule [25], which incorporates information on the atom type, its bond types, and other chemical properties.
(2) Update each atom's identifier to incorporate the identifiers of its neighboring atoms.
(3) Remove structural duplicates from the identifier list.
(4) Repeat (2) and (3) until desired radius r is reached.

An illustration of this iteration is shown in Figure 1. The algorithm results in an identifier (a bit string) for each substructure in the molecule.

4.2 Mol2vec

We base the approaches in this work on mol2vec representations [7]. The mol2vec algorithm, given an input radius r, is as follows:

(1) Process molecule to obtain Morgan fingerprints for each substructure. An identifier is created for each substructure centered on atom a. The first identifier is for radius zero, so the substructure only contains a. The next identifier is radius 1, so all atoms bonded to a are also included. This process is repeated for all atoms a starting at radius zero until a desired r value is reached.
(2) Reorder the identifiers from the molecule into a sequence. This sequence is unique, since it is based on canonical SMILES [24].
(3) Create a corpus from all the molecules available using their sequences.
(4) Use the word2vec algorithm [12] on this corpus to create embeddings for these substructures.
(5) For each molecule, create its embedding by adding together all its constituent substructure embeddings.

4.3 Baseline

For the model baseline, I use mol2vec. mol2vec representations for molecules are created by summing together the molecule graph "substructure" representations. In a chemical reaction, many substructures are preserved between the input molecules and outputs. Thus, summing the reactants (which themselves are a sum of substructures) is a good way to approximate a chemical reaction. For the baseline, I sum the mol2vec embeddings of all the reactants and compare these to all the product embeddings using cosine similarity. All the products in the dataset are ranked based on their similarity to the sum of reactants (the predicted product).

4.4 Transformer-Based Model

To improve upon the baseline model, I adapt the transformer architecture [23]. As input, I use the molecule representations from mol2vec. Inspired by the work of BERT [5], I also introduce a [CLS] token to to represent the reaction as a whole. This token is connected to a pooling layer, which is densely connected to a product layer which outputs a predicted product vector. The predicted product can be compared to the true product. An example of this model is shown in Figure 2. The output of the model is used to rank all the products in the dataset using cosine similarity.

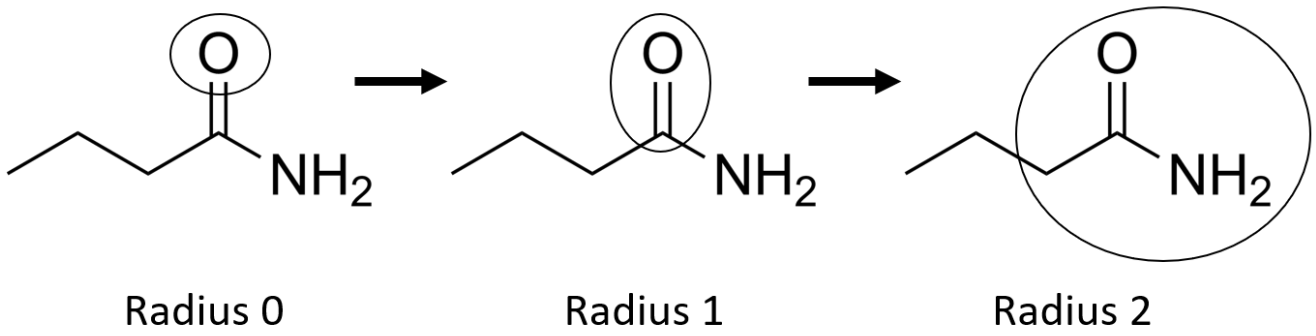

Figure 1: An example of Morgan fingerprinting iteration for an oxygen atom. The initial fingerprint of the oxygen is -1074141656, which is updated to 2099970318 after the first iteration. At each iteration, every identifier is updated based on its neighbors. The distance of nodes incorporated into an atom's identifier grows each iteration, much like how a graph convolutional network works.

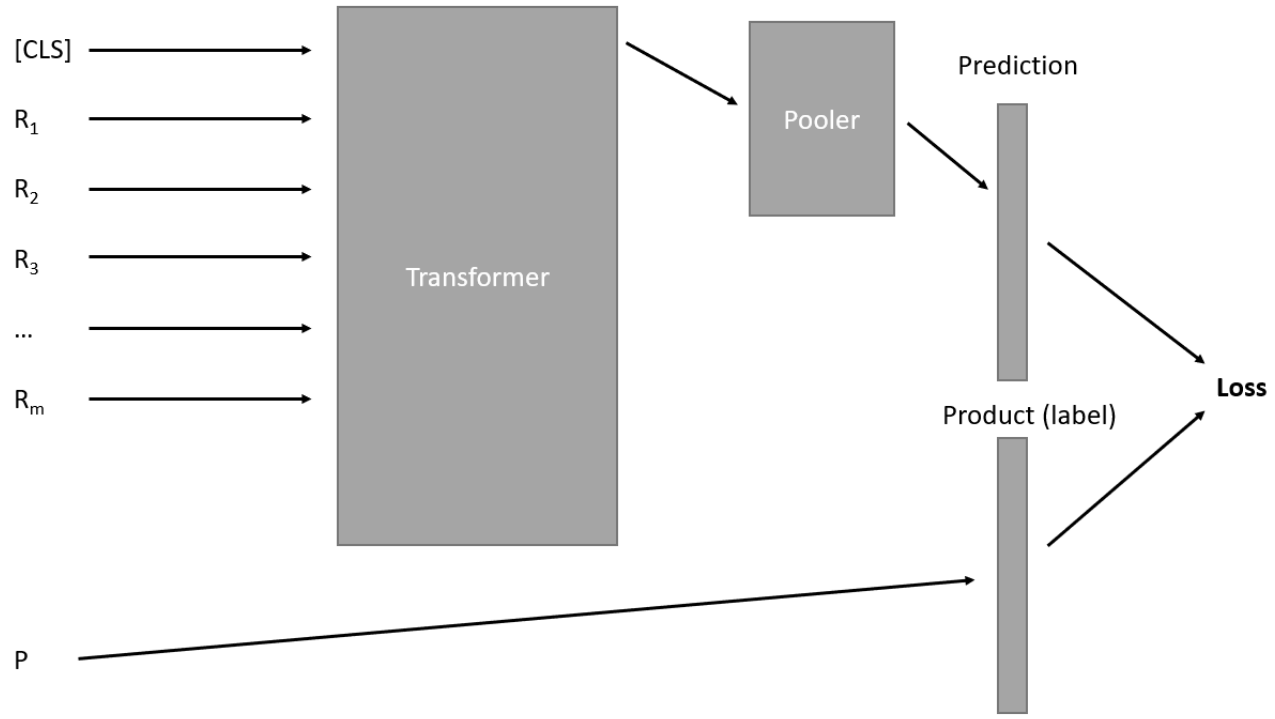

Figure 2: The transformer-based model. R_i are the reactant embedding inputs and P is the product embedding.

4.4.1 Loss. To train this model, two losses are used. The first is mean-squared error loss, which is defined as:

$$L_{MSE} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{d}(Y_i^j - \hat{Y}_i^j)^2$$

where Y_i^j is the jth feature of the ith sample and $\hat{Y}_i^j$ is the true value. d is the number of dimensions used for the molecule embeddings. Second, I use cosine loss, which is defined as:

$$L_{cos} = 1 - \frac{1}{n}\sum_{i=1}^{n} cos\_sim(Y_i, \hat{Y}_i) = 1 - \frac{1}{n}\sum_{i=1}^{n}\frac{Y_i \cdot \hat{Y}_i}{||Y_i|| \times ||\hat{Y}_i||}$$

This is essentially the cosine similarity subtracted from one, averaged over the batch size.

4.5 Transformer-Based Model for Yield Prediction

Beyond predicting reaction results, it is a paramount problem in chemistry to predict how effective the reactions will be: their yield. I try to predict the reaction yields by augmenting the transformer-based model. I add a [SEP] token followed by the product embedding to the input of the transformer-model. This takes the following form:

$$[CLS] \;\; R_1 \;\; R_2 \;\; \ldots \;\; R_m \;\; [SEP] \;\; P$$

where R_i is a reactant mol2vec embedding and P is the product embedding. I also replace the product output layer with a single output neuron and a sigmoid activation function. This allows the model to predict a yield between 0 and 1, which is optimized using mean-squared error loss.

5 EXPERIMENTS

5.1 Dataset

Since the US Patents dataset is very large (1.9 million reactions), I extracted the first 50,000 reactions. These reactions are presented as a type of SMILES string. I processed these reactions using RDKit [10], an open-source cheminformatics package for python. Doing so removed 3,850 reactions which couldn't be processed. Following this, I created an SDF (structure-data file) formatted-file from all the molecules in this reaction. I used the mol2vec algorithm to create representations for these molecules. In the process, I was forced to remove 36 molecules that did not have Morgan fingerprints out of a total of 89,401 molecules. Doing so removed a further 107 reactions, which left 46,043 reactions in the dataset. I further split this dataset using an 80%/20% train/test split, so there are 36,834 training reactions and 9,209 test reactions. Additionally, the reactions are split in chronological order. This is done to prevent the model from cheating on the test set by learning a more complicated, encompassing reaction in the training phase. I did not perform hyperparameter tuning or early-stopping, so I found this split to be sufficient. I trained my models on the training set and evaluated their performance on the test set. I found the most reactants to be 21, so I set the padding length of the transformer model to 24.

The mol2vec algorithm was used to convert the SMILES string representation of a molecule to an embedding. Default values were used: a radius of 1 and a window size of 10. Additionally, I decided not to replace infrequent tokens (substructures) with a standard UNK token (unknown). This would remove the uniqueness of rare tokens, which may substantially decrease the ability to predict reactions. Although the embedding of these rare tokens is likely poor, it is better for them to be unique from other reactants in different reactions. This way, the transformer can still learn about them, rather than learning about an "unknown" molecule substructure with properties that cannot exist in nature. Its properties would not be able to exist in nature because it implies a unique substructure; however, in some molecules the same substructure would not be compatible, so it doesn't exist.

The yields that were reported in the dataset were computed values (rather than experimentally measured). Unfortunately, this meant they had errors: there were many reactions with yields above 100%, which is only possible given impurities in the process. In fact, there was even a reaction with a yield of 28,000%! I removed these reactions, which reduced the dataset substantially by 22,623 reactions. I also normalized the yields to be between 0 and 1.

5.2 Results

All transformer models were trained using Adam optimizer [9] with a learning rate of 1e-4, which was used in the original Transformer paper [23]. A batch size of 256 was used, and the models were trained for 40 epochs.

5.2.1 Baseline. The baseline model performed very well on its own. It achieved a Hits@1 of 31.6%, as shown in Table 1. Additionally, the mean reciprocal rank (MRR) is 0.436. If all reactions had the same rank, this would be equivalent to a mean rank of $\frac{1}{0.436} = 2.29$. However, the mean rank is 301.0. This indicates that there are outliers which the baseline has a lot of trouble predicting.

5.2.2 Transformer-MSE. The transformer-based model shows a substantial improvement. Transformer-MSE (mean-squared error) has a Hits@1 value of 40.2%, which is an absolute improvement of 8.6%. Additionally, the model has a mean rank of 189.1. This indicates that it is not making many of the errors found in the baseline.

5.2.3 Transformer-COS. To further improve the model's performance, we replace the mean-squared error loss with cosine loss. This can be seen as relaxing constraints on the model. The cosine loss forces the target and predicted embeddings to be aligned. This differs from mean-squared error loss, which requires the two embeddings to be located at the same point in the space. This modification allows the model to be more flexible in how it builds an output space of products, which allows it to better handle edge cases.

Transformer-COS proves to perform significantly better than Transformer-MSE. It has a Hits@1 value of 49.6%, which is nearly 50%! This is a 9.4% absolute improvement over Transformer-MSE and an 18% absolute improvement over the baseline.

5.2.4 Transformer-Based Model for Yield Prediction. Since the dataset had computer-calculated yield values for the reaction, I attempted to modify my transformer model to predict these. If effective, this would allow for much faster calculation of reaction yields, which could be used as a tool to screen reactions to test in the lab. Unfortunately, this approach has not been effective. The error during training is shown in Figure 3, which indicates that the model does not generalize well to the test set. Although there is a slight initial improvement, this quickly vanishes. There are likely two reasons for this. The first is that the data is not representative of the true reaction yield. In order to preprocess the yield rate data, removing 22,623 reactions was required. This is because these reactions had yields greater than "100%" (including one over 28,000%), gave a range such as "60-90%", or contained words such as "around 80%". Since there is substantial noise in this data, it is likely that it may not correctly describe the true yield. The second issue is that not

Method	Mean Rank	MRR	Hits@1	Hits@10	Hits@100	Hits@1000
Baseline	301.0	0.436	31.6%	65.9%	81.5%	92.3%
Transformer-MSE	189.1	0.536	40.2%	76.4%	88.9%	95.6%
Transformer-COS	147.0	0.622	49.6%	82.7%	92.0%	96.8%

Table 1: Reaction prediction results for the test set. The transformer-based approach clearly outperforms the baseline. The Hits@1 value for model trained with cosine loss approaches 50%!

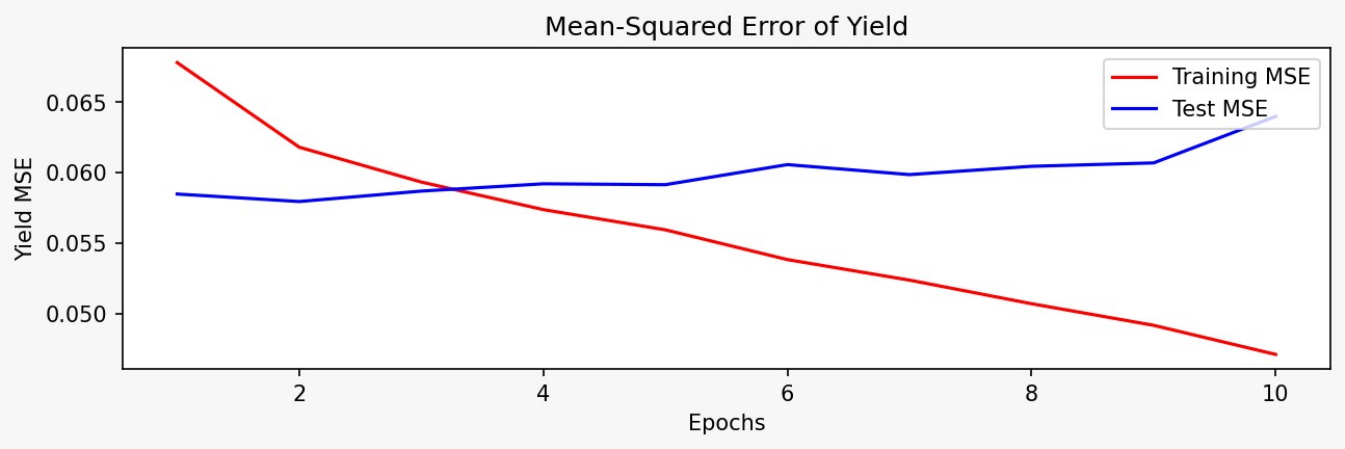

Figure 3: This figure shows the mean-squared error of the yield prediction task as it is trained. The training MSE clearly improves. However, this improvement does not generalize to the test MSE.

enough training data was used. I tried to introduce dropout to regularize the network, but this did not help. More data would help to regularize the network and improve its generalization to the test set. Additionally, since so many reactions were removed in preprocessing, the dataset was much smaller. Finally, using a chronological data split might cause issues for yield prediction, since the two splits may have different underlying data distributions.

5.3 Error Analysis

Due to using deep learning-based models, understanding the errors the model makes is very difficult. To do so, I examine high level trends and specific errors. As suggested above, the majority of reactions have a low rank. This is shown in Figure 6 for Transformer-COS. This histogram of ranks is a log-log curve. I fit a line to the smooth part of the red line (before rank 10^2), which gives the coefficients [-1.54, 7.84]. This indicates that the distribution can be described by the following power law.

$$count = 7.84 \times rank^{-1.54}$$

This indicates that large errors tend to be outliers and that the high mean rank values are skewed by these outliers.

I also investigate whether any properties of the reaction indicate that it will be more difficult to predict. Figure 7 shows that more reactants create uncertainty in the predicted product, which increases the average rank. There are two explanations for this. First, having more reactants leads to a more complex reaction, which is harder to predict. Second, there is significantly less training data for reactions with a large number of reactants, so the model might not be trained well to incorporate so many reactants. Both explanations likely play a part in this phenomenon.

Figure 4 shows an example where the Transformer-COS fails to predict the correct reaction. It is difficult to understand why it makes this error. However, this does reveal an interesting phenomenon in the dataset: many patents leave out additional products from the reaction. In the reaction in Figure 4, the third reactant contains two oxygen molecules (indicated by red Os). However, the correct final product only has one oxygen, which means that there is an additional product from the reaction that wasn't written in the patent. This is likely because chemists would implicitly understand that, for example, the molecule might also produce water (H_2O). This problem with incomplete data is a substantial challenge for computer-based reaction prediction algorithms. The performance would likely be increased with more complete reaction information. However, the model is still able to predict products in many cases without these extra products. Figure 5 shows a reaction where the chlorines are discarded. However, the model still correctly ranks the correct product number one.

6 CONCLUSION

Predicting chemical reactions is an incredibly important problem. There are millions of chemicals, and laboratory experiments are both time-consuming and slow. To facilitate faster chemical understanding and research, I applied data mining techniques to predict these reactions without laboratory effort, which could help scientists to decide which of many pressing experiments to devote their time to performing. In this work, I apply state-of-the-art deep

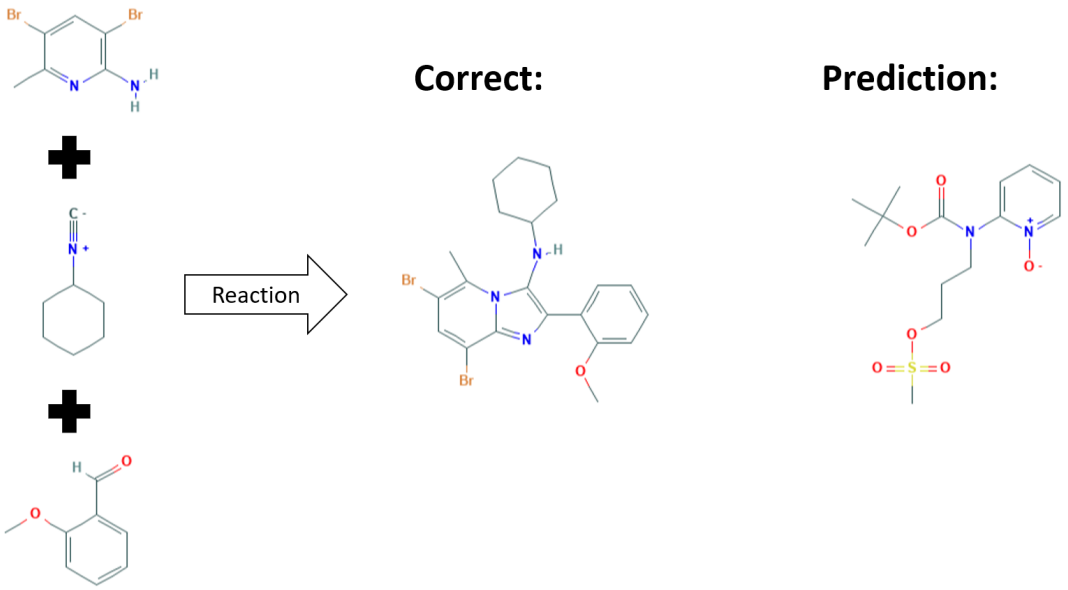

Figure 4: Reaction to form Cyclohexyl-[6,8-dibromo-2-(2-methoxyphenyl)-5-methyl-imidazo[1,2-a]pyridin-3-yl]-amine. Transformer-COS predicts this product at rank 7,488.

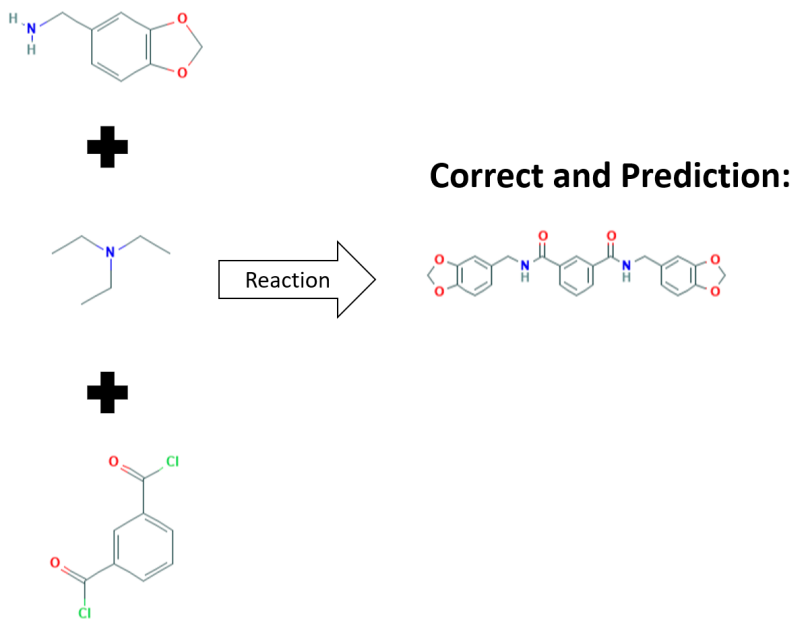

Figure 5: Reaction to form 1-N,3-N-Bis(1,3-benzodioxol-5-ylmethyl)benzene-1,3-dicarboxamide. Transformer-COS predicts this product at rank 1.

learning, data mining, and natural language processing techniques, particularly the transformer, to perform reaction prediction. This approach allowed me to reach 50% Hits@1 for product prediction on my test set. Unfortunately, the model does not work well for yield prediction, but this may be due to issues in the dataset instead.

While the prediction results are quite good, I found this problem to have multiple challenges. The first was finding and obtaining a suitable dataset. I found many possible datasets to be dead ends, and I had to plan contingency plans which took a lot of time. Although I finally found an appropriate dataset, doing so delayed my

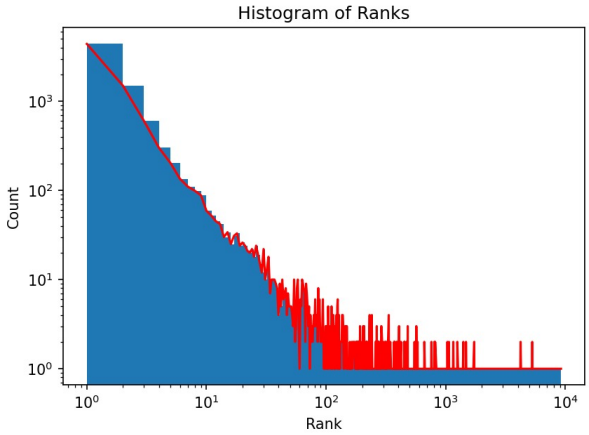

Figure 6: A histogram of the distribution of ranks for Transformer-COS. The red line is a spline visualizing the count of each unique rank.

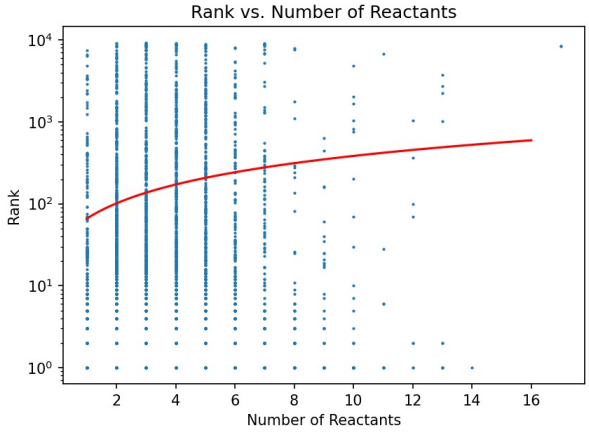

Figure 7: Scatterplot of number of reactants vs. rank for Transformer-COS. The red curve is a linear fit, which shows that more reactants cause a reaction that is more difficult to predict. The y-axis is plotted in log-space for visual clarity.

project's implementation and required me to redirect my approach. This leads to the second challenge: this reaction dataset consisted of non-binary reactions, which made it intractable to construct a graph. Thus, I reframed my problem to use an information retrieval approach, which is still similar to link prediction in knowledge graphs but using non-binary relations. These challenges forced me to be flexible and shift my thinking between different topics in data mining, which I believe has helped to solidify my understanding of the topics learned in class.

7 FUTURE WORK

By applying deep learning-based techniques, I was able improve upon my initial baseline significantly in this project. In the future, I hope to augment this approach using text mining to further improve results. I plan to extract textual information about reactions from the literature, allowing me to better predict how reactions will occur.

Additionally, much work remains in reaction yield prediction. I believe it would be beneficial to incorporate existing knowledge in the form of "reaction rules." By combining a classical approach and a distributional hypothesis-based embedding approach, it may be possible to outperform either method individually. To evaluate this, it will be necessary to find a more complete reaction yield dataset.

REFERENCES

[1] Javier L Baylon, Nicholas A Cilfone, Jeffrey R Gulcher, and Thomas W Chittenden. 2019. Enhancing retrosynthetic reaction prediction with deep learning using multiscale reaction classification. *Journal of chemical information and modeling* 59, 2 (2019), 673–688.

[2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Neural Information Processing Systems (NIPS)*. 1–9.

[3] Adrià Cereto-Massagué, María José Ojeda, Cristina Valls, Miquel Mulero, Santiago Garcia-Vallvé, and Gerard Pujadas. 2015. Molecular fingerprint similarity search in virtual screening. *Methods* 71 (2015), 58–63.

[4] Seyone Chithrananda, Gabe Grand, and Bharath Ramsundar. 2020. ChemBERTa: Large-Scale Self-Supervised Pretraining for Molecular Property Prediction. *arXiv preprint arXiv:2010.09885* (2020).

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[6] Benedek Fabian, Thomas Edlich, Héléna Gaspar, Marwin Segler, Joshua Meyers, Marco Fiscato, and Mohamed Ahmed. 2020. Molecular representation learning with language models and domain-relevant auxiliary tasks. *arXiv preprint arXiv:2011.13230* (2020).

[7] Sabrina Jaeger, Simone Fulle, and Samo Turk. 2018. Mol2vec: unsupervised machine learning approach with chemical intuition. *Journal of chemical information and modeling* 58, 1 (2018), 27–35.

[8] Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, et al. 2019. PubChem 2019 update: improved access to chemical data. *Nucleic acids research* 47, D1 (2019), D1102–D1109.

[9] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[10] G Landrum. [n.d.]. RDKit: Open-source cheminformatics. https://www.rdkit.org/

[11] Daniel Lowe. 2017. Chemical reactions from US patents (1976-Sep2016). https://doi.org/10.6084/m9.figshare.5104873.v1

[12] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546* (2013).

[13] Pieter P Plehiers, Guy B Marin, Christian V Stevens, and Kevin M Van Geem. 2018. Automated reaction database and reaction network analysis: extraction of reaction templates using cheminformatics. *Journal of cheminformatics* 10, 1 (2018), 1–18.

[14] Yannick Pouliot, Annie P Chiang, and Atul J Butte. 2011. Predicting adverse drug reactions using publicly available PubChem BioAssay data. *Clinical Pharmacology & Therapeutics* 90, 1 (2011), 90–99.

[15] David Rogers and Mathew Hahn. 2010. Extended-connectivity fingerprints. *Journal of chemical information and modeling* 50, 5 (2010), 742–754.

[16] Jae Yong Ryu, Hyun Uk Kim, and Sang Yup Lee. 2018. Deep learning improves prediction of drug–drug and drug–food interactions. *Proceedings of the National Academy of Sciences* 115, 18 (2018), E4304–E4311.

[17] Philippe Schwaller, Benjamin Hoover, Jean-Louis Reymond, Hendrik Strobelt, and Teodoro Laino. 2020. Unsupervised attention-guided atom-mapping. (2020).

[18] Philippe Schwaller, Benjamin Hoover, Jean-Louis Reymond, Hendrik Strobelt, and Teodoro Laino. 2021. Extraction of organic chemistry grammar from unsupervised learning of chemical reactions. *Science Advances* 7, 15 (2021), eabe4166.

[19] Philippe Schwaller, Daniel Probst, Alain C Vaucher, Vishnu H Nair, David Kreutter, Teodoro Laino, and Jean-Louis Reymond. 2021. Mapping the space of chemical reactions using attention-based neural networks. *Nature Machine Intelligence* 3, 2 (2021), 144–152.

[20] Marwin HS Segler and Mark P Waller. 2017. Modelling chemical reasoning to predict and invent reactions. *Chemistry–A European Journal* 23, 25 (2017), 6118–6128.

[21] Chence Shi, Minkai Xu, Hongyu Guo, Ming Zhang, and Jian Tang. 2020. A graph to graphs framework for retrosynthesis prediction. In *International Conference*

on Machine Learning. PMLR, 8818–8827.

[22] Ivar Ugi, Johannes Bauer, Klemens Bley, Alf Dengler, Andreas Dietz, Eric Fontain, Bernhard Gruber, Rainer Herges, Michael Knauer, Klaus Reitsam, et al. 1993. Computer-Assisted Solution of Chemical Problems—The Historical Development and the Present State of the Art of a New Discipline of Chemistry. *Angewandte Chemie International Edition in English* 32, 2 (1993), 201–227.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762* (2017).

[24] David Weininger. 1988. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of chemical information and computer sciences* 28, 1 (1988), 31–36.

[25] David Weininger, Arthur Weininger, and Joseph L Weininger. 1989. SMILES. 2. Algorithm for generation of unique SMILES notation. *Journal of chemical information and computer sciences* 29, 2 (1989), 97–101.

[26] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. 2018. MoleculeNet: a benchmark for molecular machine learning. *Chemical science* 9, 2 (2018), 513–530.

8 CONTRIBUTIONS

Carl was resonsible for all research, experiments, and writing. He created the three graph datasets and wrote initial graph convolutional network code to be used. This did not end up being used, since further research pointed in a different direction. He also performed data cleaning and preprocessing, he wrote the code for the transformer models (using PyTorch transformer layers) and result analysis, and he wrote the final report. He used an existing, open source, mol2vec implementation from the original paper, but he modified it to work with gensim and the data.