Predicting Chemical Reaction Outcomes: A Grammar Ontology-based Transformer Framework

Vipul Mann
Department of Chemical Engineering
Columbia University, New York
vm2583@columbia.edu

Venkat Venkatasubramanian
Department of Chemical Engineering
Columbia University, New York
venkat@columbia.edu

Abstract

Discovering and designing novel materials is a challenging problem as it often requires searching a combinatorially large space of potential candidates. Evaluation of all candidates experimentally is typically infeasible as it requires great amounts of effort, time, expertise, and money. The ability to predict reaction outcomes without performing extensive experiments is, therefore, important. Towards that goal, we report an approach that uses context-free grammar (CFG) based representations of molecules in a neural machine translation framework. We formulate the reaction-prediction task as a machine translation problem that involves discovering the transformations from the source sequence (comprising the reactants and agents) to the target sequence (comprising the major product) in the reaction. The grammar ontology-based representation of molecules hierarchically incorporates rich molecular structure information that, in principle, should be valuable for modeling chemical reactions. We achieve an accuracy of 80.1% on a standard reaction dataset using a model characterized by only a fraction of the number of training parameters in other sequence-to-sequence models based works in this area. Moreover, 99% of the predictions made on the same reaction dataset were found to be syntactically valid. We conclude that CFGs-based ontological representations could be an efficient way of incorporating structural information, ensuring chemically valid predictions, and overcoming overfitting in complex machine learning architectures employed in reaction prediction tasks.

1 Introduction

With the recent advances in machine learning algorithms complemented with significant improvements in computational capabilities – availability of better hardware, faster processing, and cheaper memory – the area of computational chemistry is seeing applications that leverage machine learning models. Some of these methods have proven to be extremely successful, thanks to the inherent efficiency of machine learning models in capturing the complex, non-linear dependencies between various factors that govern reactions systems. Machine learning architectures with their proficiency in modeling various probabilistic scenarios subject to certain conditions are therefore well-suited for such applications.

Some of the applications of machine learning methods in the area of computational chemistry include retrosynthetic analysis of chemical reactions [1], [2], [3], [4], molecular structure and property optimization [5], [6], [7], discovering new materials [8], [9], [10] including reaction catalysts [11], [12], energy storage chemicals [13], [14] and drug-like molecules [15], and predicting suitable conditions for and discovery of chemical reactions [16], [17]. Modeling of complex chemical reactions (predominantly organic) with the objective of predicting their outcomes is one such area that has shown significant promise of data-driven machine learning approaches in recent years and is the focus of our work.

The problem of predicting chemical reaction outcomes could either be formulated as a hybrid modeling problem that uses reactions templates (submolecular patterns that encode changes in atom connectivity) coupled with machine learning models or as a primarily data-driven approach using complex, end-to-end machine learning architectures that encode the reactions, discover transformations, and predict the outcomes with little or no explicit incorporation of prior chemistry knowledge. Several studies have demonstrated the reaction templates-based approach. For example, [18] uses knowledge graphs for representing chemical reactions and formulates the reaction prediction task as a search for missing links in the graph. [19] proposed using forward reaction templates to generate a set of plausible products followed by a neural network architecture that performs classification for determining the major product. [20] used a graph convolutional neural network approach for predicting reaction types followed by the application of reaction templates to predict products.

On the other hand, examples of contributions from largely data-driven methods include a two-stage approach [21] that models interactions between molecular orbitals to generate candidates as the first stage followed by the ranking of the candidates to identify the most productive reaction during the second stage; representing reactants pool as an attributed graph and using a graph convolutional neural network (CNN) approach for generating likelihood scores to identify the most likely product [22]; and Weisfeiler-Lehman Networks-based method for scoring candidate molecules by modeling high-order interactions between changes occurring in a molecule [23].

A subclass of methods falling under this category is sequence-to-sequence (seq2seq) models, more commonly seen in the area of natural language processing. The studies using seq2seq models include [24] that formulated the reaction prediction task as a translation problem modeled using gated-recurrent units (GRU) based architecture; an encoder-decoder framework based on the recurrent neural networks (RNN) architecture using long short-term memory (LSTM) [25]; a similar encoder-decoder architecture for retrosynthetic reaction prediction was used in [26]; and more recently [27] demonstrated the use of the transformer architecture [28] for reaction prediction and is claimed to outperform all known algorithms in the reaction prediction literature. The proposed approach in our work is based on the seq2seq class of methods for reaction prediction.

An important aspect of using machine learning methods is the representation of the input and the target features in the model architecture, ensuring that the features are information-rich and have predictive signatures unique to the problem under consideration. Past work in the area of seq2seq models-based reaction prediction have used character-based representations of molecules such as the Simplified Molecular-Input Line-Entry System (or SMILES) representation [29]. A more structured way of representing molecules is by using a formal-grammar underlying the SMILES representation, akin to context-free grammar (CFG) in natural language processing [30]. Recently, [31] has demonstrated the use of a grammar-based representation in the context of a Bayesian framework for single-molecule property optimization while searching for drug-like molecules. Such a representation leverages the formal structure underlying representations such as SMILES and offers several advantages that we highlight in our work. This representation is analogous to the ontology-centric frameworks that are known to be semantically rich and efficiently describe the semantics of the information sources. Ontologies [32] have been around for quite some time across various engineering domains including process engineering [33], pharmaceutical engineering [34], [35], materials science [36], and molecular engineering [37]. The individual SMILES tokens representing the molecules are constituents of the ontology whereas the grammar-rules describe the relationships between these concepts and hence, the underlying SMILES grammar could be exploited as an ontology for molecular representation.

In this paper, we propose the use of such grammar ontology-based molecular representations for predicting the outcomes of chemical reactions in a neural machine translation framework. To the best of our knowledge, such representations have not been used in the context of reaction prediction involving interactions between multiple molecules – reactants, agents (reagents and catalysts), and products. We highlight certain benefits inherent to such representations and propose an approach for leveraging this in the reaction prediction framework. Our approach is based on using the transformer architecture for modeling chemical reactions as natural language translation tasks, albeit using a grammar-based framework. The proposed approach, Grammar Ontology-based Prediction of Reaction Outcomes (GO-PRO) extends to any sequence modeling problem, in general, with the existence of an underlying formal-grammar as a precondition that serves as an ontology for knowledge representation in this framework.

The rest of the paper is organized as follows: In Section 2, we provide a formal description of the problem, objectives, and the machine translation framework that we work with. In Section 3.1, we formally describe a context-free grammar and the SMILES grammar that is used in our work and in Section 3.2, we present the various aspects of the transformer architecture that we use for the sequence modeling task. We summarize the standard datasets used for validation of the proposed approach in Section 4. The main contribution of this work, the GO-PRO framework, is described in detail in Section 5 with descriptions on the necessary preprocessing steps, reaction encoding strategy, and the model architecture and training. In Section 6, we present results on standard datasets along with comparisons with other works highlighting the advantages and limitations of our approach. Finally, a summary of the useful contributions of this work appear in Section 7.

2   Problem formulation and objectives

Given a set of reactants and the agents facilitating the chemical reaction, our objective is to predict the most likely major product of the reaction. We formulate this as a machine translation problem where the input sequences comprising the reactants and agents correspond to the source sentence and the output comprising the major product of the reaction corresponds to the target sentence (from a different language). The sentence analogues in this translation task are the set of SMILES strings whereas the characters in each SMILES string are their word analogues as in a natural language sentence. We exploit this analogy between chemical reactions transformation and natural language translation for predicting reaction outcomes.

The SMILES strings, however, are comprised of arbitrary characters that do not provide chemical or structural information crucial for modeling reaction chemistry systems. We therefore use a SMILES grammar, analogous to context-free grammars in natural language [30], in order to incorporate structural information for each molecule in our reaction prediction framework in a hierarchical manner. The sequence of grammar rules corresponding to each SMILES string therefore becomes their representation in this framework as shown in Figure 1.
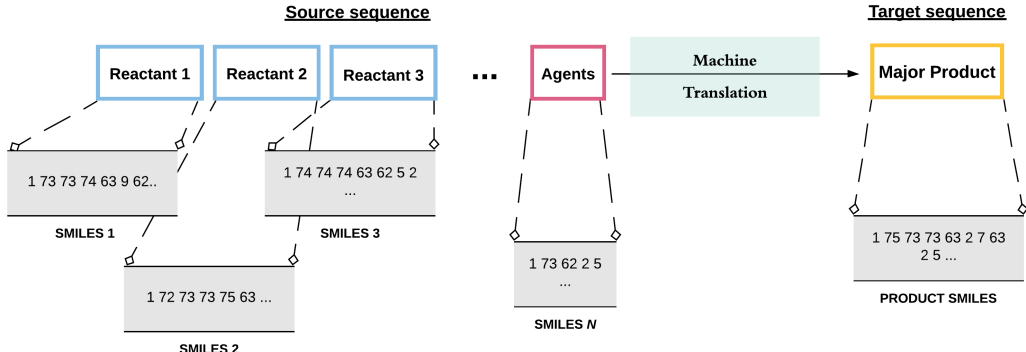

Figure 1: Modeling a chemical reaction prediction task as a machine translation problem

An important contribution of this work is to represent the molecules using a formal SMILES-grammar as proposed in [31] but extended to the reaction prediction framework involving multiple reactants, agents, and product molecules. This ontological representation has several advantages such as explicit incorporation of chemical structure, reduction of strain on the model by letting it discover the transformations in a chemical reaction directly without a need to model the relationships between arbitrary characters, overcoming overfitting in neural machine translation models often characterized by a large number of parameters, and increased likelihood of predicting molecules with valid SMILES representations.

3   Methods

In this section, we describe the concepts of context-free grammar both in the context of natural language processing used to describe the relationships between different parts of a natural language

sentence and in the context of chemistry for SMILES representation of molecules, followed by a brief description of the transformer architecture that we use for performing sequence modeling.

3.1 Formal grammar

Formal grammars have been the backbone of various language modeling tasks such as semantic interpretation of natural language, dialogue understanding, and machine translation. They are largely based on the idea that group of words belong to the same constituent units and that different constituents could be hierarchically grouped together to convey the given meaning [38].

3.1.1 Context-free grammar (CFG)

The most widely used formal grammar is the Context-Free Grammar (CFG) and was formalized in [30]. A context-free grammar consists of a set of productions (or rules) that express the way in which different words or symbols, comprising the lexicon in the language, can be grouped and ordered together. The symbols used in a CFG are grouped into two classes – symbols that correspond to the actual words with meaning in the language, called terminals, and the symbols that represent abstraction over a group of words and are used to represent a class of words or phrases in the language (terminals), called non-terminals.

Formally, a context-free grammar G is represented by four parameters – N, Σ, R, S where

- N: a set of non-terminal symbols
- Σ: a set of terminal symbols
- R : a set of production or rules of the form A $\longrightarrow \beta$, where A is non-terminal and β is a string of symbols from the set $(\Sigma \bigcup N)_*$
- S: a designated start symbol and a member of N

Typical English grammar rules comprise sentence level constructions (S $\longrightarrow$ NP VP, S $\longrightarrow$ VP), the noun phrase (Det $\longrightarrow$ NP), the verb phrase (VP $\longrightarrow$ Verb, VP $\longrightarrow$ Verb NP) and so on, where S, NP, VP, Det, and Verb are the sentence symbol, noun phrase, verb phrase, determiner, and verb, respectively.

A CFG can be thought of as a generator that could be used to generated sentences in a language by sequential application of productions, or as a tool for assigning structure to a given sentence [38]. In our work, we primarily focus on this latter aspect of CFGs and use them to incorporate structural information from a SMILES string.

3.1.2 Grammar for SMILES

Analogous to the context-free grammar for the English language, there exists a formal grammar for the string-based molecular representations used in chemistry such as the most commonly used representation – SMILES [29]. As described in the foregoing section, the set of productions (or rules), non-terminals, terminals, and a designated start symbol are the essential components of a context-free grammar. These components for the SMILES representations are presented in [1], that could be applied sequentially to generate the grammar-based parse trees representing the constituency of various components in a given SMILES string.

For instance, consider the simplified grammar in Table 1. Analogous to the notation for context-free grammar introduced in Section 3.1.1, the following are their equivalents in this grammar.

- N: { SMILES, CHAIN, BRANCHED_ATOM, BOND, ATOM, RINGBOND, BB, RB, BRANCH, AROMATIC_ORGANIC, ALIPHATIC_ORGANIC, DIGIT }
- Σ: { (,), =, c, C, O, 1, 2 }
- R: productions (rules) 1 through 20 in Table 1
- S: SMILES

---

[1]http://opensmiles.org/spec/open-smiles-2-grammar.html

Table 1: Reduced SMILES grammar

S.No	Production rules
1	SMILES $\longrightarrow$ CHAIN
2	CHAIN $\longrightarrow$ CHAIN BRANCHED_ATOM
3	CHAIN $\longrightarrow$ CHAIN BOND BRANCHED_ATOM
4	CHAIN $\longrightarrow$ BRANCHED_ATOM
5	BRANCHED_ATOM $\longrightarrow$ ATOM RINGBOND
6	BRANCHED_ATOM $\longrightarrow$ ATOM
7	BRANCHED_ATOM $\longrightarrow$ ATOM BB
8	BRANCHED_ATOM $\longrightarrow$ ATOM RB
9	BB $\longrightarrow$ BRANCH
10	RB $\longrightarrow$ RINGBOND
11	BRANCH $\longrightarrow$ (CHAIN)
12	RINGBOND $\longrightarrow$ DIGIT
13	BOND $\longrightarrow$ =
14	ATOM $\longrightarrow$ AROMATIC_ORGANIC
15	ATOM $\longrightarrow$ ALIPHATIC_ORGANIC
16	AROMATIC_ORGANIC $\longrightarrow$ c
17	ALIPHATIC_ORGANIC $\longrightarrow$ C
18	ALIPHATIC_ORGANIC $\longrightarrow$ O
19	DIGIT $\longrightarrow$ 1
20	DIGIT $\longrightarrow$ 2

When this grammar is used to parse the SMILES string representation for methyl ethylene (propene), `CC=C`, the following parse-tree is obtained:
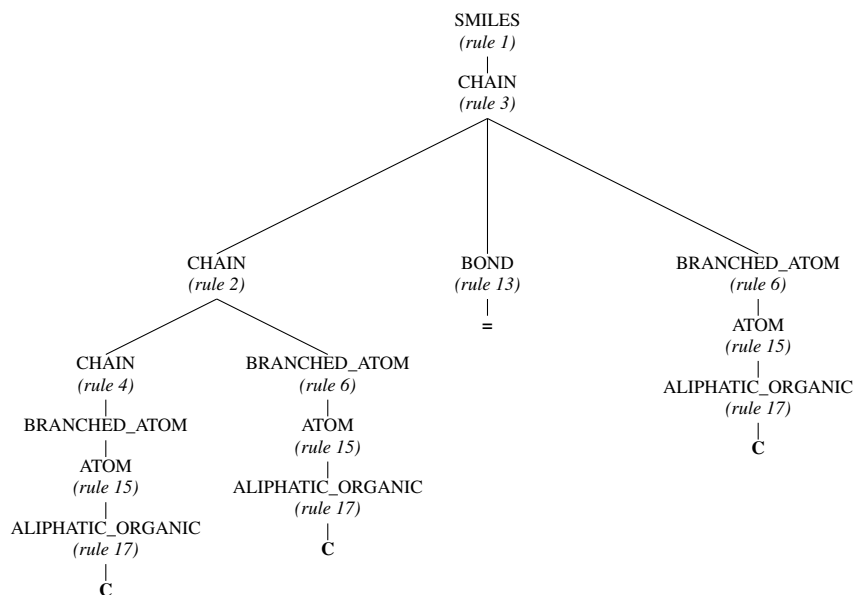
Figure 2: The parse-tree obtained for `CC=C` using the representative grammar above

The above parse tree contains information about the various chemistry aspects of the given molecule. For instance, it contains information such as the number of aromatic carbon atoms, the presence of a ring-structure, the alternating double bonds in the ring, the presence of an aliphatic oxygen atom, and finally the active hydrogen atom attached to the oxygen atom. Moreover, this information is represented in a hierarchical manner, with the broadest class of rules at the top and increasingly more specific ones towards the bottom of the parse-tree. We encode the parse-tree structure in Figure 2

using the sequence of productions used to generate the given structure as the *sentence* analogue in our language model with the individual rule indices as the equivalent *word* analogues.

Contrasting such a grammar-based representation with a purely string-based representation that treats each of the tokens comprising the SMILES string ('C', 'C', '=', 'C) as independent entities, the differences between the two are evident. A model trained on a purely character or string-based representation would require the model to first understand the structural relationships between the different tokens comprising the SMILES string which is not a trivial task for any neural language model architecture, and only then model the transformation between the reaction space to the product space.

Remark 1: Although the grammatical validity of a SMILES string does not necessarily mean that the corresponding compound is chemically feasibly, it is a step closer towards ensuring synthesizable molecules are predicted as the output.

Remark 2: In contrast to the English language, the proposed SMILES-grammar based molecular representation does not suffer from ambiguity with respect to its constituency parsing structure since a given (canonicalized) SMILES string cannot correspond to two completely different molecules under different contexts.

3.2 Transformers

The transformer architecture was proposed recently in [28] for machine translation tasks and comprises an encoder-decoder architecture that is more parallelizable and superior to other seq2seq architectures. Transformers replaced the complex recurrent (or convolutional) neural network layers with simpler attention based mechanisms proposed in [39] combined with positional embedding for encoding sequential information.

In the following sections, we briefly describe the concepts of the encoder-decoder architecture, positional encoding, and the attention-mechanism that comprise the building blocks of a transformer.

3.2.1 Encoder-Decoder architecture

The transformer architecture primarily consists of an encoder-decoder structure, wherein the encoder maps an input sequence $x_1, x_2, \ldots, x_n$ to a continuous latent-space representation $z_1, z_2, \ldots, z_n$. Given $\mathbf{z}$, the decoder generates the output sequence $y_1, y_2, \ldots, y_n$ one element at a time, in an autoregressive manner, consuming the previously generated tokens as additional input while generating the next.

The encoder and decoder consist of stacks of identical layers, each of which are comprised of two sublayers – a multi-head attention mechanism, and a fully connected feed-forward neural network. There are residual connections around each of the sublayers along with a batch normalization. The decoder, in addition, consists of an additional layer which performs a multi-head attention over the output of the encoder.

3.2.2 Positional encoding

Since the transformer architecture does not contain recurrent or convolution layers, the sequential information of the tokens in a sequence is fed to the model through these embeddings. Mathematically, positional encoding is a mapping of the position of a given word (*pos*, an integer) in the sequence to a d-dimensional vector space ($\vec{p}_{pos}$). These mappings are characterized by sines and cosines of different frequencies, given by

$$\vec{p}_{pos,i} = \begin{cases} sin(pos/10000^{2k/d}), & \text{if } i = 2k \\ cos(pos/10000^{2k/d}), & \text{if } i = 2k + 1 \end{cases} \tag{1}$$

The positional encodings are added to the word embeddings representing the individual tokens in a sentence, thus, the dimensions of the two embeddings, d_{word} and d_{pos}, must be the same so that the two can be summed, i.e.

$$\vec{w}(t)' = \vec{w}(t) + \vec{p}_{pos^t} \tag{2}$$

where $\vec{w}(t)'$ represents the word embedding with encoded position information, $\vec{w}(t)$ represents the word embedding, and $\vec{p}_{pos^t}$ represent the positional encoding.

3.2.3 Attention mechanism

The attention mechanism lies at the heart of the transformer architecture and allows the model to focus on different tokens in the sequence at different stages of the network, enabling it to discover multiple relationships between groups of tokens.

The attention-mechanism used in [28] is the 'Scaled-Dot Product Attention', characterized by a set of queries, keys, and values vectors. The query and key vectors are of dimensions d_k and the value vector is of dimension d_v. The attention-score then, is computed as softmax function applied over the dot-products of the queries and key vectors, scaled down by a factor of $\sqrt{d_k}$, given by

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{3}$$

where Q, K, and V are the matrices of query, key, and values vectors, respectively. The attention score computed above determines the importance that should be given to different parts of an input sequence in the current context. In order to allow the model to jointly factor in information from different representation subspaces at different positions, multi-headed attention is computed which involves computing multiple attention scores, in parallel, which are then concatenated and projected using a linear transformation to compute the multi-head attention scores as,

$$MultiHead(Q, K, V) = Concat(head_1, head_2, \dots, head_h)W^O \tag{4}$$

where $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$, and $W_i^Q \in \mathbb{R}^{d_{pos} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{pos} \times d_k}$, and $W_i^V \in \mathbb{R}^{d_{pos} \times d_v}$ are the projection matrices for Q, K, and V, respectively.

4 Data

We work with two standard reaction datasets – Jin's USPTO dataset [23] and its subset of 80 reactions used for benchmarking reaction prediction against human organic chemists in [22]. Lowe's grants database [40], based on the text mining work done on US reactions patents granted between 1976 and 2013, has now become one of the standard datasets for demonstrating quantitative approaches for reaction prediction. Since this dataset contains erroneous and duplicate reactions, there are several datasets derived from this mining work that address such issues – excluding stereochemical information, retaining only single product reactions, and removing certain class of reactions. Jin's USPTO dataset is one such derived datasets that only includes single-product reactions. We therefore primarily work with this dataset to evaluate the performance of the proposed approach and perform a comparative analysis using the latter dataset. Table 2 summarizes the two datasets.

Table 2: A summary of the two datasets used for validating the proposed reaction prediction approach

Dataset	train	valid	test	total
Jin's USPTO				
with (sanitized) single product				479,035
in grammar	385,429	28,269	37,676	451,374
Human dataset				
with (sanitized) single product	-	-	80	80
in grammar	-	-	78	78

5 Grammar Ontology-based Prediction of Reaction Outcomes

In this section, we describe the proposed approach – Grammar Ontology-based Prediction of Reaction Outcomes, or GO-PRO, in detail and the various components involved in this framework. The critical components of GO-PRO are the molecular representations (such as SMILES) for molecules involved in the reaction, an underlying formal grammar describing the syntactic aspects of the chemical

identifier, contextual information about the reaction such as the agents involved, and a neural machine translation framework that *translates* the reaction and agents information to the product.

5.1 Preprocessing

Across all the reactions in the two datasets, we apply certain preprocessing steps. These include sanitizing each molecule involved in the reaction, removing atom-mappings from the reactions, and canonicalizing the SMILES strings since multiple strings could be used to represent a given molecule. The reactions in the given database are filtered based on the number of reactants such that reactions with more than 9 reactants in the reactant pool are discarded.

The reactants involved in the two datasets are then ordered based on their similarity with the major product of the reaction, where the molecular similarity score is computed using molecular fingerprints in RDKit, an open source cheminformatics and machine learning software for reaction chemistry systems. Without loss of generality, we only work with the first three reactants and a single agent molecule – which in our case is the fourth reactant molecule since we do not make any distinction between the reactants and the agents. The rationale behind working with only three reactants is to reduce the the strain on the model training phase since increasing the number of reactants would translate into significantly higher training time even though the number of training parameters remain unchanged. Since we use a grammar-based representation as described in Section 3.1.2, we only retain molecules that are in grammar, i.e. the SMILES strings for the molecules could be parsed using the grammar. The complete grammar used in this work, which is a subset of the official SMILES grammar [41], is presented in the Appendix in Table 8 . Moreover, since the grammar is recursive and could be extremely long for certain molecules, we set a threshold on the maximum number of grammar rules allowed at 300, and molecules longer than this in the datasets are skipped. The encoding strategy is described in detail in the next section.

5.2 Grammar-based encoding of reactions

As described earlier, grammar ontology-based representations have certain inherent advantages over other representations such as explicit incorporation of structural information, ensuring output validity from a chemistry perspective, and incurring less strain on the model. The rule index corresponding to the first rule (SMILES $\longrightarrow$ CHAIN) marks the beginning of the sentence while the last rule corresponding to (NOTHING $\longrightarrow$ NONE) signifies the end of the sentence, similar to the grammar rules proposed for a molecule optimization framework in [31].

Given a molecule, its canonical SMILES representation is parsed using the above grammar which consequently gives rise to a parse-tree representation for the given string. The grammar rules are extracted from the parse-tree, preserving the order in which they were applied. For instance, for a given parse-tree, a bottom-up-left-corner parsing strategy is used wherein all the left-most derivations are first explored until the tree-depth is reached followed by sequentially moving back up until the entire tree is parsed. This ordered sequence of grammar-rules represents our molecular encoding strategy.

Next, after encoding the molecules involved in the reaction, the left-hand and right-hand sides of the entire reaction have to be encoded. To this end, all the reactant representations are concatenated horizontally followed by the concatenation of the agent's representation. Owing to the presence of a unique identifier marking the start and end of a string, the model can still distinguish between the different molecules involved in the reaction. The representation for the right-hand side of the reaction remains the same as the encoding for the product since we only consider single-product reactions.

It should be noted that in order to ensure a fixed-length representation for all the reactions, the left and right hand side representations are zero-padded until the threshold length. We have fixed the the lengths of the vector representations for the left and right hand sides at 600 and 300, respectively. Figure 3 shows an overview of the encoding scheme used in our work (using the reduced SMILES grammar in Table 1).
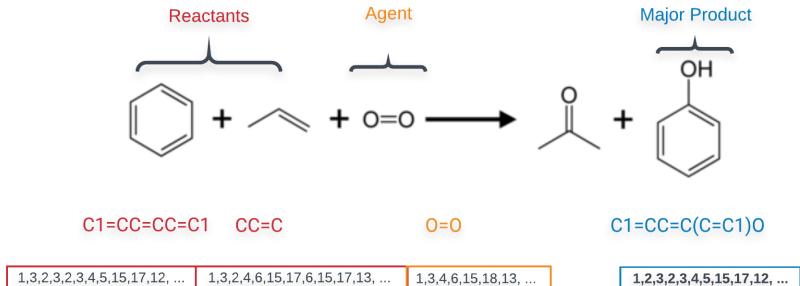
Figure 3: An overview of the proposed grammar-based reaction encoding strategy

5.3 Model architecture and training

We work with a transformer architecture described in Section 3.2 that consists of an encoder-decoder architecture. For decoding from the latent space, there are two possible approaches – first, using a greedy strategy that gives as output only the sequence with the highest likelihood and second, using a beam-search that returns a set of top-k target sequences based on their probabilities and should improve the accuracy of the model due to a wider search at each stage. However, we enforce the strictest constraint on our model and compute the accuracy metrics only against the sequence with the highest probability, and thus use a greedy-strategy for decoding.

The transformer architecture, like any other machine learning architecture, consists of several hyper-parameters that need to be tuned for achieving the desired performance. We therefore search for the best hyperparameter values by evaluating various model architectures on the validation set of Jin's USPTO dataset. Table 3 describes the possible hyperparameters in the model and along with the final values in the model. The final model is characterized by $\sim$ 5M training parameters.

Table 3: Possible and best hyperparameter values for the model architecture

Hyperparameter	Possible values	Final model
Embedding dimensions	128, 256, 512	256
Attention heads	4, 8, 16	8
Feedforward network units	512, 1024	512
Number of layers	4, 6	4
Dropout	0.1, 0.2	0.1
Warmup steps	1k, 4k, 8k, 12k	12k

The model was trained using the Adam optimizer [42] with beta $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 10^{-9}$, and a learning rate that is characterized by a fixed number of warmup steps and given by

$$lr = d_{model}^{-0.5}.min(step\_num^{-0.5}, step\_num * warmup\_steps^{-0.5}) \tag{5}$$

where d_model is the embedding dimension (positional). At the training stage, in order to avoid overfitting, a dropout layer is used for both the feed-forward networks and the attention-mechanism, for the encoder as well as the decoder. A loss function based on sparse categorical cross entropy between the predicted and actual target sequences is minimized. The model was trained using TensorFlow 2.1 and python 3.7 for 60 epochs. For generating the parse-trees and extracting grammar-based features, we use the Natural Language ToolKit (NLTK) 3.4.5 library. The molecular datasets were processed using the 2019 release of RDKit library.

6 Results and Discussions

6.1 Performance measures

In order to evaluate the performance of our approach, we consider four different measures that capture different aspects of the model performance, namely – the BLEU (Bilingual Evaluation Understudy)

score [43] which is a standard metric used for the evaluation of a given translation against the reference translation; the accuracy of prediction computed by identifying perfect matches between the predicted and the actual product (the strictest possible measure for evaluation); syntactic validity of the predicted outputs by determining if the predicted molecules SMILES string is *in grammar* i.e. could be parsed by the given grammar; and character-based similarity[2] between the actual and the predicted SMILES strings that measures the similarity between substructures within the given set of strings

The above four measures capturing the performance of GO-PRO on the test-set of Jin's UPSTO dataset is summarized in Table 4.

Table 4: Results on the test set for Jin's USPTO dataset

BLEU	accuracy	valid fraction	similarity
93.2%	80.1%	99.0%	95.8%

In order to further understand the model performance on this dataset, we look at the split of similarity scores across the reactions in the dataset across three bins with similarity scores of more than 0.95, 0.85, and 0.75, as presented in Table 5.

Table 5: Similarity scores on the test set for Jin's USPTO dataset

similarity ≥ 0.95	similarity ≥ 0.85	similarity ≥ 0.75
84.4%	90.3%	93.4%

Based on the above results, a few conclusions about the efficiency of the proposed approach could be drawn. First, only 1% of the predicted reaction outcomes resulted in invalid SMILES strings that could not be parsed by the given grammar. This indicates that the transformer-model has learnt the underlying SMILES grammar almost with perfection from the reaction encoding strategy. Second, the BLEU score and similarity values suggest that the predicted products are very similar to the actual products of the reaction. This is not trivial since the reactants, especially in organic chemistry reactions, often give rise to products that are significantly different from each one of them after only a few elementary transformations involving addition, substitution, and elimination reactions between different groups. This is further established through Table 5 where the splits indicate that over 90% of the predicted products share a similarity of more than 0.85 with the actual product. Third, the top-1 accuracy of over 80% on the test set is indicative of the fact that the model has really discovered the complex transformations occurring in chemical reactions subject to the reactions conditions. A comparison of the model performance with human chemists presented in the following section validates this claim.

6.2 Comparison with human organic chemists

In this section, we report the performance of our approach on the dataset of 80 reactions used for benchmarking against human organic chemists used in [19]. The test set contains 80 randomly chosen reactions from Jin's USPTO dataset, 10 from each of the 8 category of reaction templates, categorized based on their frequency of occurrence. The comparison of the model accuracy with the average performance of the human chemists across various reaction template bins is presented in Figure 4. The performance measures for the model on this dataset are summarized in Table 6.

Clearly, the model outperforms the chemists for each of the reaction template bins with 100% prediction accuracies for the second and third categories of reaction templates. Even for the increasingly rare reactions represented by the following bins, the model achieves an accuracy of over 70% except for the last two bins where the performance is comparable to the human chemists. Moreover, as is evident from Table 6, 100% of the predictions made by GO-PRO on this dataset correspond to valid

---

[2]computed using the SequenceMatcher routine in python that matches the longest contiguous matching sub-sequence that does not contain any unwanted (or junk) elements

SMILES strings, again reinforcing the advantages of a grammar ontology-based encoding strategy for reactions.
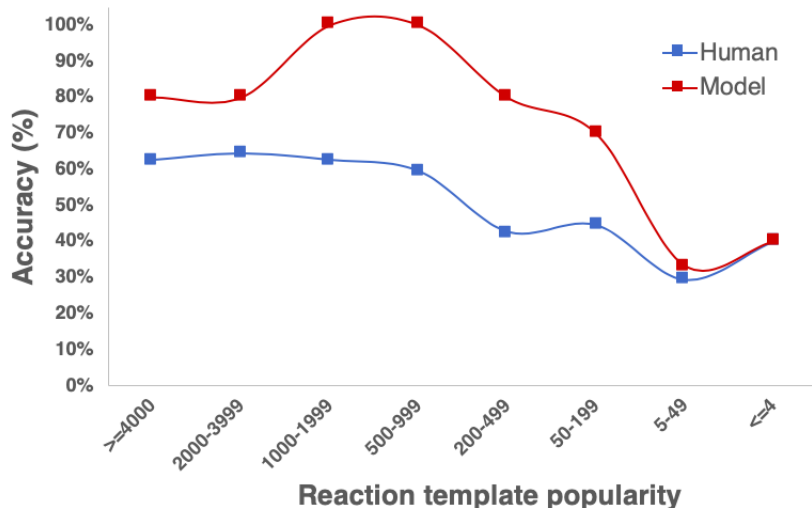

Figure 4: Prediction accuracy of the model and the average accuracy of human chemists versus reaction template popularity

Performance measure	(in %)
BLEU	93.2
accuracy	72.9
valid fraction	100.0
similarity score	94.4

Table 6: Performance measures for model on the human chemists dataset

Figure 5 visualizes some of the incorrect predictions made by our model on this dataset. We observe that even when the predictions were inaccurate, the predicted products were very similar to the actual product of the reaction – based on their structural forms in Figure 5 and also based on the BLEU and similarity scores from Table 6.

6.3   Comparison with other works

The current state of the art model in the reaction prediction literature is the molecular transformer model [27]. Though the overall accuracy obtained using our approach does not outperform the best model, our model achieves an accuracy of over 80% just by using a fraction of the training parameters characterizing the model used in other works. A comparison of the top-1 accuracies and the number of parameters characterizing the seq2seq model used in other works is presented in Table 7.

Table 7: Comparison of top-1 accuracies (in %) reported in other works involving seq2seq models

Model	# parameters	Jin's USPTO
Molecular transformer[27]	12 M	88.6
S2S [25]	30 M	80.3
GO-PRO[3]	**5 M**	**80.1**

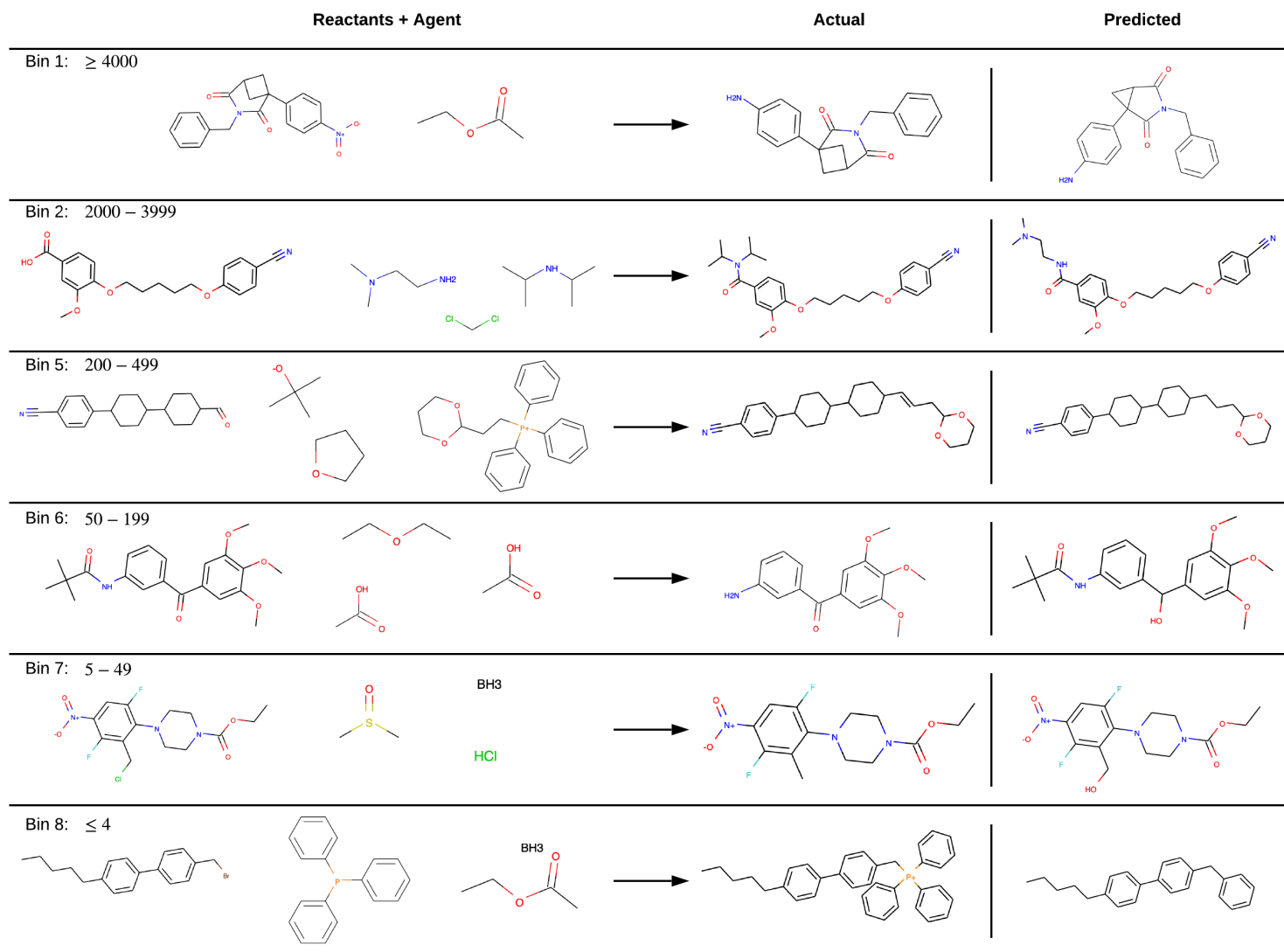[3]using only 3 reactants and 1 agent for predicting the major product of the reaction

Figure 5: Some of the reactions in the human chemists dataset that were predicted incorrectly by our model. Even the incorrect predictions share a structure very similar to the actual product of the reaction. The bin popularity along with the their frequency of appearance in the database is indicated for each reaction.

Based on this, we claim that our grammar-based approach significantly aids the model in learning transformations that occur in a chemical reaction by explicit incorporation of the relationships between the constituent tokens in a SMILES string. A significantly fewer number of parameters also inherently implies that the model does not have the capability to *memorize* the entire training set and therefore, overcomes overfitting during the model training stage – making it more robust and generalizable in practice. The following are the advantages of using the proposed grammar-based representation for reaction prediction

- the grammar-based representation explicitly encodes the structural information for the given molecule in a hierarchical manner with constituency relationships mapped between different components in the SMILES string

- the proposed (more systematic) representation incurs less strain on the model and consequently requires significantly lesser complexity in the model architecture, letting the model focus on discovering the underlying transformations in a chemical reaction

- owing to the grammar-based representation, the model learns to predict the product based on the same grammar, and hence, the output SMILES strings are more likely to be syntactically valid

It is imperative to note here that a relatively lower accuracy in our model could be attributed to certain assumptions and approximations made during the model building stage. First, the model was trained on only three reactants and one agent, discarding all the other molecules involved in the reaction. Moreover, even among these four molecules, those that were not *in-grammar* were dropped while encoding the reaction. Second, the SMILES grammar that we used here does not include metallic ions, certain metallic catalysts, and inorganic elements, limiting the coverage of the training dataset. Third, molecules with representations of over 300 were discarded and reactions were truncated if the left-hand side representation was longer than 600. Owing to these limitations/approximations, the model is constrained and has restrictive predictive capabilities that affects the prediction accuracies. However, these could be overcome by relaxing these constraints which although does not result in an increase in the model parameters, increases the model training time significantly due to longer sequences. In addition, implementing a beam search would increase the accuracy of the predictions further as demonstrated in other similar works using seq2seq models.

7 Conclusions

In this paper, we present an approach for exploiting the grammar underlying the SMILES representation for molecules as an ontology in the reaction prediction framework. We have shown that grammar ontology-based representations offer certain advantages inherent to the reaction prediction task. First, they reduce the strain on the model incurred while modeling relationships between individual tokens in a text-based molecular representation (such as SMILES) by encoding such relationships explicitly in the input and target sequences. Second, they overcome over-parameterization in complex machine learning architectures typically used in the reaction prediction tasks as observed through the significantly reduced number of training parameters in our proposed architecture. Third, such representations ensure syntactic validity of the molecular representations predicted as outcomes of chemical reactions, taking us a step closer towards constraining the model to predict synthesizable molecules.

The proposed approach results in 99.0% of the predictions to be syntactically valid with an overall accuracy of 80.1% using a model characterized by 5M parameters. In contrast, the current state of the art in reaction prediction achieves an accuracy of 88.6% using a model characterized by 12M parameters, significantly higher than the number of parameters used in our model. The proposed approach, GO-PRO, has outperformed the average accuracy of human organic chemists across both common as well as infrequent class of reaction-templates, with 100% of the predicted molecules being syntactically valid SMILES strings. Based on these results, we conclude that context-free grammars (CFGs) could be exploited to develop efficient ontologies for reaction prediction frameworks that encode reactions hierarchically, reflecting the peculiar characteristics of molecular transformations inherent to a chemical reaction. Moreover, the accuracies could be further improved by implementing a beam-search approach so that the ground truth could be compared against a set of possible reaction outcomes. We therefore claim that ontologies that incorporate prior structured-information about the constituents would significantly aid the machine learning models used in such applications.

Though there are numerous benefits to using the proposed grammar-based representation, there are certain limitations. First, a grammar-based approach cannot efficiently incorporate reaction conditions such as temperature, pressure, heating/cooling conditions, and inorganic or metallic catalysts. Second, the current framework does not have the ability to predict outcomes of reactions with multiple products and makes an inherent assumption that the reaction under consideration is a single-product reaction. We plan to address these limitations in our future work by incorporating additional reaction condition information such as temperature, pressure, and (inorganic) catalysts, and performing retrosynthetic reaction prediction using a similar grammar ontology-based framework.

Acknowledgements

Appendix

The SMILES grammar used in this works is a subset of the official OpenSMILES [4] specification [41] and comprises 80 production rules with 24 non-terminals symbols specifying the different structural components of a SMILES string. All the production rules for the grammar used in our work are summarized in Table 8. Similar to the grammar production rules used in [31] for molecule optimization, additional production rules are included to mark the beginning and end of a SMILES string. The first and the last production rules, SMILES $\longrightarrow$ CHAIN and NOTHING $\longrightarrow$ NONE, are used to signify the start and end of a SMILES string. Analogously, they correspond to the <START> and <END> tokens in natural language processing that mark the beginning and the end of sentence, respectively.

Remark 3: The listed grammar rules below were sufficient to parse the SMILES strings that largely correspond to organic molecules in the given databases and hence, the unused production rules corresponding to inorganic compounds in the official OpenSMILES grammar were removed.

Table 8: SMILES grammar used in GO-PRO

S.No	Production rules
1	SMILES $\longrightarrow$ CHAIN
2	ATOM $\longrightarrow$ BRACKET_ATOM | ALIPHATIC_ORGANIC | AROMATIC_ORGANIC
3	ALIPHATIC_ORGANIC $\longrightarrow$ B | C | N | O | S | P | F | I | Cl | Br
4	AROMATIC_ORGANIC $\longrightarrow$ c | n | o | s | p
5	BRACKET_ATOM $\longrightarrow$ [BAI]
6	BAI $\longrightarrow$ ISOTOPE SYMBOL BAC | SYMBOL BAC | ISOTOPE SYMBOL | SYMBOL
7	BAC $\longrightarrow$ CHIRAL BAH | BAH | CHIRAL
8	BAH $\longrightarrow$ HCOUNT BACH | BACH | HCOUNT
9	BACH $\longrightarrow$ CHARGECLASS | CHARGE | CLASS
10	SYMBOL $\longrightarrow$ ALIPHATIC_ORGANIC | AROMATIC_ORGANIC | ELEMENT_SYMBOLS
11	ISOTOPE $\longrightarrow$ DIGIT | DIGIT DIGIT | DIGIT DIGIT DIGIT
12	DIGIT $\longrightarrow$ 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8
13	CHIRAL $\longrightarrow$ @ | @@
14	HCOUNT $\longrightarrow$ H | H DIGIT
15	CHARGE $\longrightarrow$ - | - DIGIT | - DIGIT DIGIT | + | + DIGIT | + DIGIT DIGIT
16	BOND $\longrightarrow$ - | = | # | / | \\
17	RINGBOND $\longrightarrow$ DIGIT | BOND DIGIT
18	BRANCHED_ATOM $\longrightarrow$ ATOM | ATOM RB | ATOM RB BB
19	RB $\longrightarrow$ RB RINGBOND | RINGBOND
20	BB $\longrightarrow$ BB BRANCH | BRANCH
21	BRANCH $\longrightarrow$ (CHAIN) | (BOND CHAIN)
22	CHAIN $\longrightarrow$ BRANCHED_ATOM | CHAIN BRANCHED_ATOM | CHAIN BOND BRANCHED_ATOM
23	CLASS $\longrightarrow$ DIGIT
24	ELEMENT_SYMBOLS $\longrightarrow$ H
25	NOTHING $\longrightarrow$ NONE

---

[4]http://opensmiles.org/spec/open-smiles-2-grammar.html

References

[1] Marwin HS Segler and Mark P Waller. Neural-symbolic machine learning for retrosynthesis and reaction prediction. *Chemistry–A European Journal*, 23(25):5966–5971, 2017.

[2] Kangjie Lin, Youjun Xu, Jianfeng Pei, and Luhua Lai. Automatic retrosynthetic pathway planning using template-free models, 2019.

[3] John S. Schreck, Connor W. Coley, and Kyle J. M. Bishop. Learning retrosynthetic planning through simulated experience. *ACS Central Science*, 5(6):970–981, 2019.

[4] Philippe Schwaller, Riccardo Petraglia, Valerio Zullo, Vishnu H. Nair, Rico Andreas Haeuselmann, Riccardo Pisoni, Costas Bekas, Anna Iuliano, and Teodoro Laino. Predicting retrosynthetic pathways using transformer-based models and a hyper-graph exploration strategy. *Chem. Sci.*, 11:3316–3325, 2020.

[5] Venkat Venkatasubramanian, King Chan, and James M Caruthers. Computer-aided molecular design using genetic algorithms. *Computers & Chemical Engineering*, 18(9):833–844, 1994.

[6] Kyle V Camarda and Costas D Maranas. Optimization in polymer design using connectivity indices. *Industrial & Engineering Chemistry Research*, 38(5):1884–1892, 1999.

[7] Daniel C Elton, Zois Boukouvalas, Mark D Fuge, and Peter W Chung. Deep learning for molecular design—a review of the state of the art. *Molecular Systems Design & Engineering*, 4(4):828–849, 2019.

[8] Logan Ward, Muratahan Aykol, Ben Blaiszik, Ian Foster, Bryce Meredig, James Saal, and Santosh Suram. Strategies for accelerating the adoption of materials informatics. *MRS Bulletin*, 43(9):683–689, 2018.

[9] Dezhen Xue, Prasanna V Balachandran, John Hogden, James Theiler, Deqing Xue, and Turab Lookman. Accelerated search for materials with targeted properties by adaptive design. *Nature communications*, 7:11241, 2016.

[10] Tarak K Patra, Venkatesh Meenakshisundaram, Jui-Hsiang Hung, and David S Simmons. Neural-network-biased genetic algorithms for materials design: Evolutionary algorithms that learn. *ACS combinatorial science*, 19(2):96–107, 2017.

[11] Bryan R Goldsmith, Jacques Esterhuizen, Jin-Xun Liu, Christopher J Bartel, and Christopher Sutton. Machine learning for heterogeneous catalyst design and discovery. 2018.

[12] Zheng Li, Siwen Wang, Wei Shan Chin, Luke E Achenie, and Hongliang Xin. High-throughput screening of bimetallic catalysts enabled by machine learning. *Journal of Materials Chemistry A*, 5(46):24131–24138, 2017.

[13] Benjamin Sanchez-Lengeling and Alán Aspuru-Guzik. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365, 2018.

[14] Motoaki Nishijima, Takuya Ootani, Yuichi Kamimura, Toshitsugu Sueki, Shogo Esaki, Shunsuke Murai, Koji Fujita, Katsuhisa Tanaka, Koji Ohira, Yukinori Koyama, et al. Accelerated discovery of cathode materials with prolonged cycle life for lithium-ion battery. *Nature communications*, 5(1):1–7, 2014.

[15] Jessica Vamathevan, Dominic Clark, Paul Czodrowski, Ian Dunham, Edgardo Ferran, George Lee, Bin Li, Anant Madabhushi, Parantu Shah, Michaela Spitzer, et al. Applications of machine learning in drug discovery and development. *Nature Reviews Drug Discovery*, 18(6):463–477, 2019.

[16] Hanyu Gao, Thomas J Struble, Connor W Coley, Yuran Wang, William H Green, and Klavs F Jensen. Using machine learning to predict suitable conditions for organic reactions. *ACS central science*, 4(11):1465–1476, 2018.

[17] Jarosław M Granda, Liva Donina, Vincenza Dragone, De-Liang Long, and Leroy Cronin. Controlling an organic synthesis robot with machine learning to search for new reactivity. *Nature*, 559(7714):377–381, 2018.

[18] Marwin H. S. Segler and Mark P. Waller. Modelling chemical reasoning to predict and invent reactions. *Chemistry - A European Journal*, 23(25):6118–6128, Jan 2017.

[19] Connor W. Coley, Regina Barzilay, Tommi S. Jaakkola, William H. Green, and Klavs F. Jensen. Prediction of organic reaction outcomes using machine learning. *ACS Central Science*, 3(5):434–443, 2017. PMID: 28573205.

[20] Jennifer N. Wei, David Duvenaud, and Alán Aspuru-Guzik. Neural networks for the prediction of organic chemistry reactions. *ACS Central Science*, 2(10):725–732, 2016. PMID: 27800555.

[21] Matthew A. Kayala and Pierre Baldi. Reactionpredictor: Prediction of complex chemical reactions at the mechanistic level using machine learning. *Journal of Chemical Information and Modeling*, 52(10):2526–2540, 2012. PMID: 22978639.

[22] Connor W Coley, Wengong Jin, Luke Rogers, Timothy F Jamison, Tommi S Jaakkola, William H Green, Regina Barzilay, and Klavs F Jensen. A graph-convolutional neural network model for the prediction of chemical reactivity. *Chemical science*, 10(2):370–377, 2019.

[23] Wengong Jin, Connor W. Coley, Regina Barzilay, and Tommi Jaakkola. Predicting organic reaction outcomes with weisfeiler-lehman network, 2017.

[24] Juno Nam and Jurae Kim. Linking the neural machine translation and the prediction of organic chemistry reactions, 2016.

[25] Philippe Schwaller, Theophile Gaudin, David Lanyi, Costas Bekas, and Teodoro Laino. "found in translation": Predicting outcomes of complex organic chemistry reactions using neural sequence-to-sequence models, 2017.

[26] Bowen Liu, Bharath Ramsundar, Prasad Kawthekar, Jade Shi, Joseph Gomes, Quang Luu Nguyen, Stephen Ho, Jack Sloane, Paul Wender, and Vijay Pande. Retrosynthetic reaction prediction using neural sequence-to-sequence models. *ACS Central Science*, 3(10):1103–1113, 2017. PMID: 29104927.

[27] Philippe Schwaller, Teodoro Laino, Théophile Gaudin, Peter Bolgar, Christopher A. Hunter, Costas Bekas, and Alpha A. Lee. Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction. *ACS Central Science*, 5(9):1572–1583, Aug 2019.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.

[29] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, 1988.

[30] N. Chomsky. Three models for the description of language. *IRE Transactions on Information Theory*, 2(3):113–124, Sep. 1956.

[31] Matt J. Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. 2017.

[32] Thomas R Gruber. Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, 43(5-6):907–928, 1995.

[33] Jan Morbach, Andreas Wiesner, and Wolfgang Marquardt. Ontocape—a (re) usable ontology for computer-aided process engineering. *Computers & Chemical Engineering*, 33(10):1546–1556, 2009.

[34] Leaelaf Hailemariam and Venkat Venkatasubramanian. Purdue ontology for pharmaceutical engineering: Part I. Conceptual framework. *Journal of Pharmaceutical Innovation*, 5(3):88–99, oct 2010.

[35] Leaelaf Hailemariam and Venkat Venkatasubramanian. Purdue ontology for pharmaceutical engineering: Part II. Applications. *Journal of Pharmaceutical Innovation*, 5(4):139–146, 2010.

[36] Xiaoming Zhang, Chongchong Zhao, and Xiang Wang. A survey on knowledge representation in materials science and engineering: An ontological perspective. *Computers in Industry*, 73:8–22, 2015.

[37] Howard J Feldman, Michel Dumontier, Susan Ling, Norbert Haider, and Christopher WV Hogue. Co: A chemical ontology for identification of functional groups and semantic comparison of small molecules. *FEBS letters*, 579(21):4685–4691, 2005.

[38] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, USA, 1st edition, 2000.

[39] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. 2015.

[40] D. M. Lowe. Patent reaction extraction: downloads, 2014.

[41] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, 1988.

[42] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[43] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.