Received: 15 September 2020 | Revised: 22 November 2020 | Accepted: 20 December 2020

DOI: 10.1002/aic.17190

TRIBUTE TO FOUNDERS: KEITH GUBBINS ISSUE.
PROCESS SYSTEMS ENGINEERING

AIChE JOURNAL

Predicting chemical reaction outcomes: A grammar ontology-based transformer framework

Vipul Mann | **Venkat Venkatasubramanian**

Department of Chemical Engineering, Columbia University, New York, New York, 10027

Correspondence
Venkat Venkatasubramanian, Department of Chemical Engineering, Columbia University, New York, NY 10027.
Email: venkat@columbia.edu

Funding information
Center for the Management of Systemic Risk (CMSR), Columbia University, New York, NY 10027

Abstract

Discovering and designing novel materials is a challenging problem as it often requires searching through a combinatorially large space of potential candidates, typically requiring great amounts of effort, time, expertise, and money. The ability to predict reaction outcomes without performing extensive experiments is, therefore, important. Toward that goal, we report an approach that uses context-free grammar-based representations of molecules in a neural machine translation framework. This 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 hierarchically incorporates rich molecular-structure information, ensures syntactic validity of predictions, and overcomes over-parameterization in complex machine learning architectures. We achieve an accuracy of 80.1% (86.3% top-2 accuracy) and 99% syntactic validity of predictions on a standard reaction dataset. Moreover, our model is characterized by only a fraction of the number of training parameters used in other similar works in this area.

KEYWORDS
artificial intelligence, computational chemistry (reaction modeling), context-free grammar, computational chemistry (organic reactions), neural machine translation

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,[1-4] molecular structure and property optimization,[5-7] discovering new materials[8-10] 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, Segler and Waller[18] use

knowledge graphs for representing chemical reactions and formulate the reaction prediction task as a search for missing links in the graph. Coley et al.[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. Wei et al.[20] used a graph-convolutional neural network (CNN) 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 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 Nam and Kim[24] that formulated the reaction prediction task as a translation problem modeled using gated-recurrent units 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 Liu et al.[26]; and more recently Schwaller et al.[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, Kusner et al.[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 CFG 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 reaction transformations 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 CFGs 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

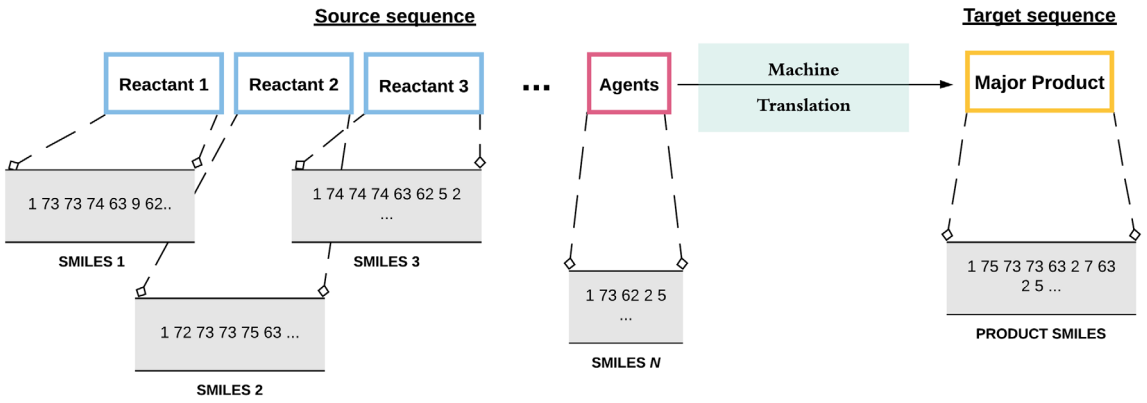

FIGURE 1 Modeling a chemical reaction prediction task as a machine translation problem [Color figure can be viewed at wileyonlinelibrary.com]

rules corresponding to each SMILES string therefore becomes their representation in this framework as shown in Figure 1.

An important contribution of this work is to represent the molecules using a formal SMILES-grammar as proposed in Kusner et al.[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 CFG 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

The most widely used formal grammar is the CFG and was formalized in Chomsky.[30] A CFG 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 CFG 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 $A \rightarrow \beta$, where A is non-terminal and β is a string of symbols from the set $(\Sigma \cup N)^*$
- S: a designated start symbol and a member of N

Typical English grammar rules comprise sentence level constructions ($S \rightarrow$ NP VP, $S \rightarrow$ VP), the noun phrase (Det $\rightarrow$ NP), the verb phrase (VP $\rightarrow$ Verb, VP $\rightarrow$ 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 CFG 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 CFG. These components for the SMILES representations are presented in the website: http://opensmiles.org/spec/open-smiles-2-grammar.html, that could be applied sequentially to

generate the grammar-based parse tree 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 CFG introduced in the 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

In order to motivate the grammar representations proposed in our framework, we consider methyl ethylene (propene) and cyclopropane with SMILES string representations as CC=C and C1CC1, respectively. The parse tree structures corresponding to the two strings are as in Figures 2 and 3, respectively. The grammar representation for each of these molecules correspond to the sequence of production rules extracted when these structures are parsed in a bottom-up left-corner strategy as highlighted in their respective schematics.

Consider the parse tree for propene given in Figure 2. This 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 toward 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 transformations between the reaction space and 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 toward 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.

TABLE 1 Representative SMILES grammar

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

Abbreviation: SMILES, simplified molecular-input line-entry system.

3.2 | Transformers

The transformer architecture was proposed recently in Vaswani et al.[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 Bahdanau et al.[39] combined with positional embedding for encoding sequential information. An overview of the transformer architecture as proposed in Vaswani et al.[28] is presented in Figure 4.

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, ..., x_n to a continuous latent-space representation z_1, z_2, ..., z_n. Given $\mathbf{z}$, the decoder generates the output sequence y_1, y_2, ..., y_n

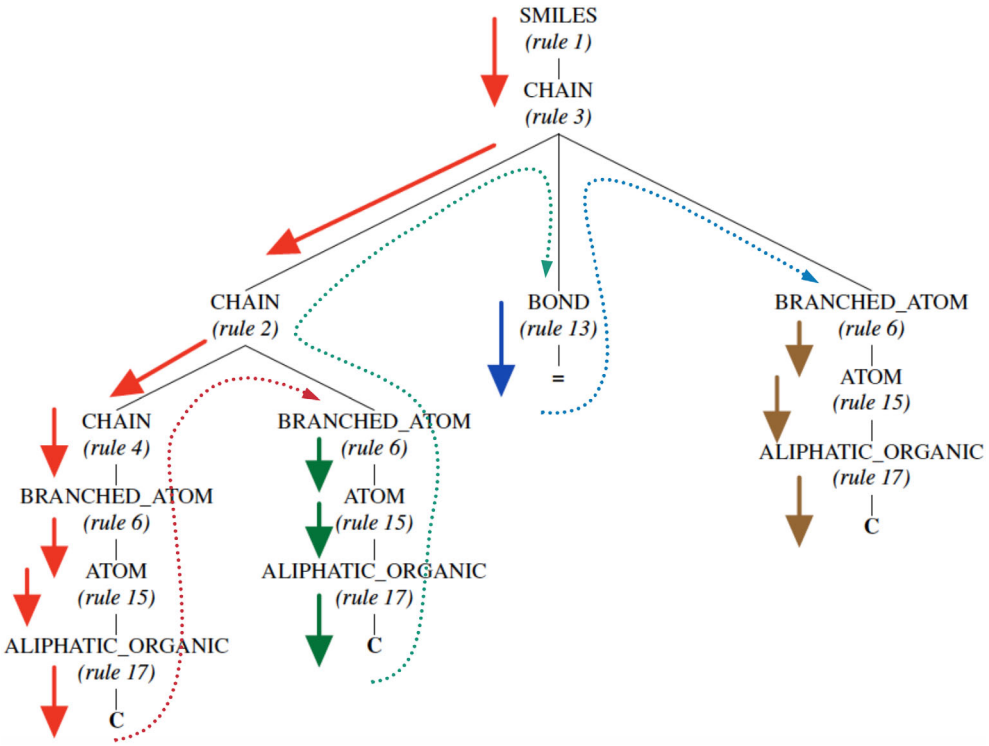

FIGURE 2 The parse-tree obtained for propene with the simplified molecular-input line-entry system (SMILES) representation as `CC=C` using the representative grammar in Table 1. The sequence of production rule indices obtained while parsing the above tree corresponds to the grammar representation and is given as 1, 3, 2, 4, 6, 15, 17, 6, 15, 17, 13, 6, 15, 17 [Color figure can be viewed at wileyonlinelibrary.com]

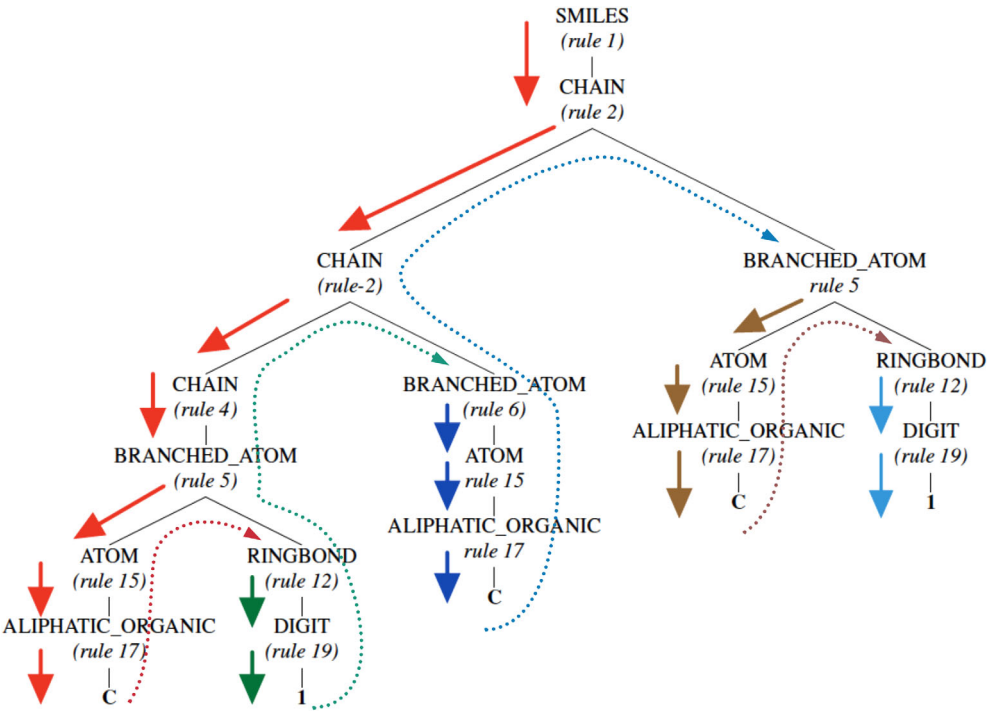

FIGURE 3 Another motivating example representing the parse-tree structure obtained for cyclopropane with simplified molecular-input line-entry system (SMILES) representation as `C1CC1`. The equivalent grammar representation is given as 1, 2, 2, 4, 5, 15, 17, 12, 19, 6, 15, 17, 5, 15, 17 12, 19 [Color figure can be viewed at wileyonlinelibrary.com]

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-headed attention over the output of the encoder.

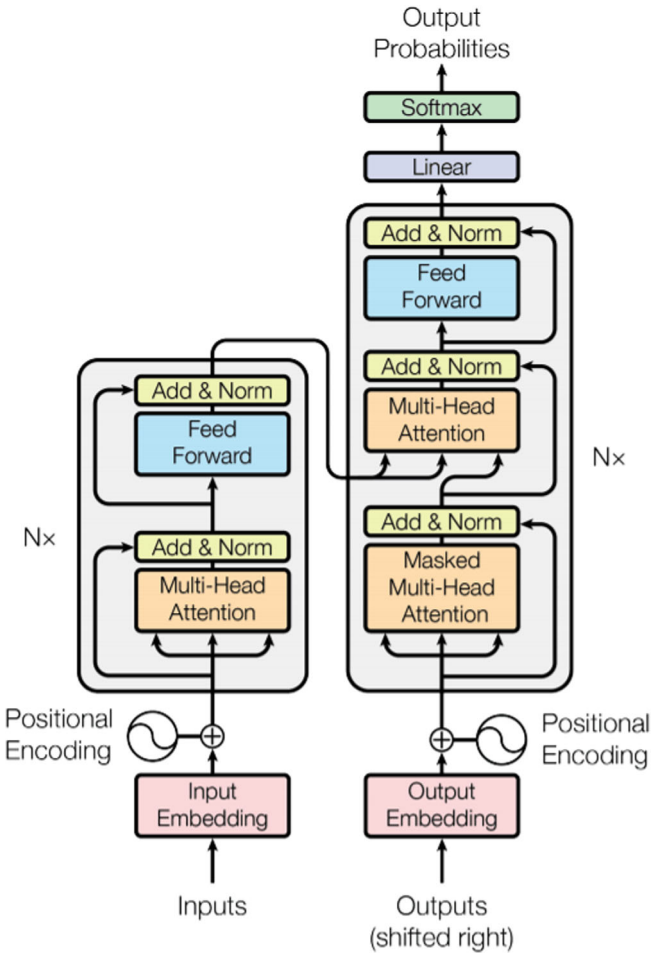

FIGURE 4 The encoder–decoder model architecture of a transformer. The left-half corresponds to the encoder whereas the right-half corresponds to the decoder, the positional information is encoded using the positional embedding, and multi-head attention mechanism aids the model in discovering relationships between groups of tokens at different stages [Color figure can be viewed at wileyonlinelibrary.com]

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

$$\vec{p}_{pos,i} = \begin{cases} \sin\left(pos/10{,}000^{2k/d}\right), & \text{if } i = 2k \\ \cos\left(pos/10{,}000^{2k/d}\right), & \text{if } i = 2k+1 \end{cases} \quad (1)$$

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, that is,

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

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 Vaswani et al[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

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

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,

$$\text{MultiHead}(Q,K,V) = \text{Concat}(\text{head}_1,\text{head}_2,\ldots,\text{head}_h)W^O \quad (4)$$

where $\text{head}_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right)$, 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 Coley et al.[22] Lowe's grants database,[40] based on the text mining work done on U.S. 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.

5 | GRAMMAR ONTOLOGY-BASED PREDICTION OF REACTION OUTCOMES

In this section, we describe the proposed approach—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 reactants and agents to the major product of the reaction.

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 nine 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 in the database 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 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 the Section 3.1.2, we only retain molecules that are in grammar, that is, 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,[29] is presented in the Appendix in Table A1. 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 with grammar-representations longer than this 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 in terms of the number of parameters that need to be trained using a given set of data points. The rule index corresponding to the first rule (`SMILES → CHAIN`) marks the beginning of the sentence while the last rule corresponding to (`NOTHING → NONE`) signifies the end of the sentence, similar to the grammar rules proposed for a molecule optimization framework in Kusner et al.[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

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

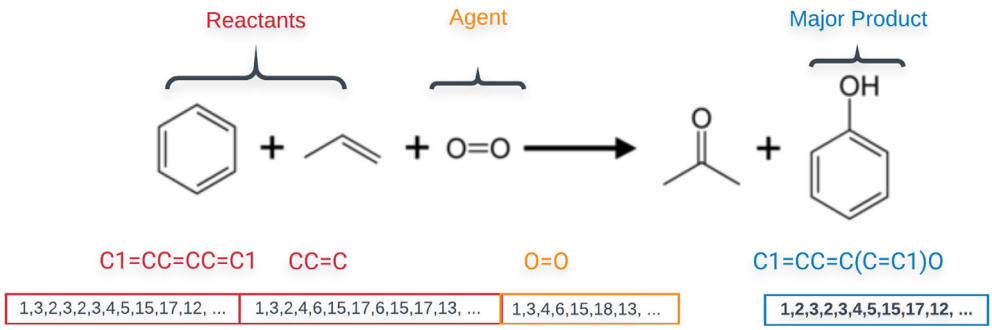

FIGURE 5 An overview of the proposed grammar-based reaction encoding strategy [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 3 Possible and best hyperparameter values for the model architecture described in Figure 4

Hyperparameter	Possible values	Final model
Embedding dimensions	128, 256, 512	256
Attention heads	4, 8, 16	8
Feedforward network units	512, 1,024	512
Number of layers	4, 6	4
Dropout	0.1, 0.2	0.1
Warmup steps	1 k, 4 k, 8 k, 12 k	12 k

are zero-padded until the threshold length. We have fixed the lengths of the vector representations for the left and right hand sides at 600 and 300, respectively. Figure 5 shows an overview of the encoding scheme used in our work (using the representative SMILES grammar in Table 1).

5.3 | Model architecture and training

We work with a transformer architecture described in the Section 3.1 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-B target sequences based on their probabilities. For a beam width of size B, the beam search algorithm decodes a set of B most likely tokens at any stage (based on their conditional probabilities) which are then used to generate the next set of most likely tokens at the next stage until the entire sequence is decoded. Therefore, the output of the beam search algorithm is a set of B most likely sequences which are then used to compute the top-k accuracies, as opposed to a greedy search that only returns the most likely decoded sequence. We, therefore, use the latter approach with a beam width of 3 and compute the top-1, top-2 and top-3 accuracies to evaluate the model performance.

The transformer architecture, like any other machine learning architecture, also consists of several hyperparameters 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 along with their values in the final model architecture. The final model is characterized by ~5 M training parameters.

The model was trained using the Adam optimizer[41] 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

$$lr = d_{\text{model}}^{-0.5}.\min\left(\text{step\_num}^{-0.5}, \text{step\_num} \times \text{warmup\_steps}^{-0.5}\right) \quad (5)$$

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 Bilingual Evaluation Understudy (BLEU) score[42] which is a standard metric used for the evaluation of a given translation against the reference translation; the top-1, top-2, and top-3 accuracies computed by identifying perfect matches between the predictions and the actual product; syntactic validity of the predicted outputs by determining if the predicted molecules SMILES string is *in grammar*, that is, could be parsed by the given grammar; and character-based similarity[1] 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.

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.

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 in Coley et al.[19] The test set contains 80 randomly chosen reactions from Jin's USPTO dataset, 10 from each of the 8 categories 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 6. The performance measures for the model on this dataset are summarized in Table 6.

Clearly, the model outperforms the chemists across 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, 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 SMILES strings, again reinforcing the advantages of a grammar ontology-based encoding strategy for reactions.

Figure 7 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 7 and also based on the BLEU and similarity scores from Table 6.

TABLE 4 Results on the test set of Jin's USPTO dataset computed using the top-1 predictions

BLEU	Top-1 accuracy	Valid fraction	Similarity
93.2%	80.1%	99.0%	95.8%

TABLE 5 Distribution of similarity scores computed on the test set of Jin's USPTO dataset corresponding to the top-1 predictions

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

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 accuracies and the number of parameters characterizing the seq2seq model used in other works is presented in Table 7.

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 representations explicitly encode the structural information for a given molecule in a hierarchical manner with constituency relationships mapped between different components in the SMILES string. This is evident by the contrasting features of a

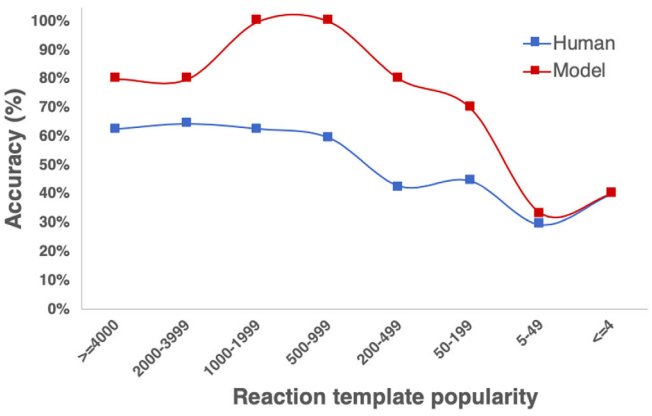

FIGURE 6 Prediction accuracy of the model and the average accuracy of human chemists versus reaction template popularity [Color figure can be viewed at wileyonlinelibrary.com]

parse-tree based grammar-representations (Figures 2 and 3) and their equivalent character-based SMILES string representations described in the Section 3.1.2.

- the proposed (more systematic) representation incurs less strain on the model (in terms of number of training parameters) and consequently requires a significantly less complex model architecture for modeling the underlying transformations in a chemical reaction, as seen through the results in Table 7 where the proposed model is characterized by only a fraction of the training parameters in other works with comparable accuracies.

TABLE 6 Performance measures for model on the human chemists dataset

Performance measure	(in %)
BLEU	93.2
Top-1 accuracy	72.9
Valid fraction	100.0
Similarity score	94.4

- owing to the grammar-based representations, 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. This is validated by the results in Tables 4 and 6 which demonstrate that 99 and 100% of the predictions made by our model (on the test set) are syntactically valid which would have been unlikely without the model learning the underlying grammar production rules.

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

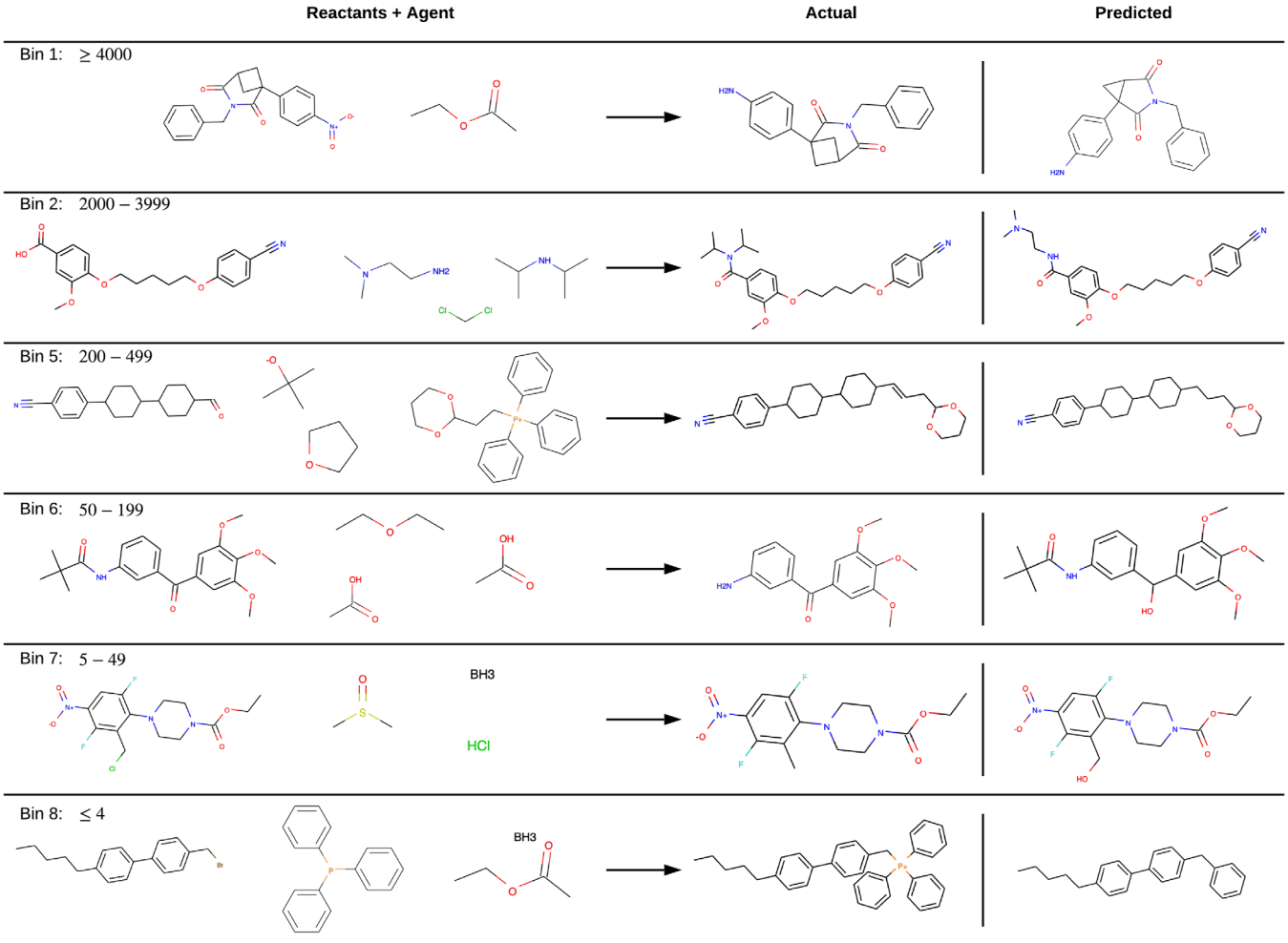

FIGURE 7 Some of the reactions in the human chemists dataset that were predicted incorrectly by our model. Even the incorrect predictions share a structure very similar to the actual product of the reaction. The bin popularity along with the their frequency of appearance in the database is indicated for each reaction [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 7 Comparison of accuracies (in %) reported in other works involving seq2seq models using Jin's USPTO dataset

Model	# parameters	Top-1	Top-2	Top-3
Molecular transformer[27]	12 M	88.6	92.4	93.5
S2S[25]	30 M	80.3	84.7	86.2
GO-PRO[a]	5 M	80.1	86.3	88.7

[a]Using only three reactants and one agent for predicting the major product of the reaction.

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.

7 | CONCLUSIONS

In this paper, we present an approach for exploiting the grammar underlying the SMILES representation of 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 training stage 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 using the underlying grammar. 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 toward 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 5 M parameters. In contrast, the current state of the art in reaction prediction achieves an accuracy of 88.6% using a model characterized by 12 M 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 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.

Although 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 conditions such as temperature, pressure, and (inorganic) catalysts, and performing retrosynthetic reaction prediction using a similar grammar ontology-based framework.

ACKNOWLEDGEMENTS

Venkat Venkatasubramanian would like to offer his sincere gratitude and appreciation to Professor Keith Gubbins for being an indulgent doctoral mentor. Keith was kind enough to let Venkat explore his wide-ranging intellectual interests during his doctoral study, even though Keith knew that Venkat was perhaps spending more time on these interests than on his own thesis topic! In fact, Venkat's research interests in artificial intelligence, neural networks, and cognitive science were formed in 1982 when he was still a PhD student in Keith's group at Cornell. The authors also would like to acknowledge the partial financial support of the Center for the Management of Systemic Risk (CMSR) at Columbia University.

AUTHOR CONTRIBUTIONS

Vipul Mann: Formal analysis; software. **Venkatasubramanian Venkat:** Conceptualization; formal analysis; methodology.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Vipul Mann https://orcid.org/0000-0003-0225-8729
Venkat Venkatasubramanian https://orcid.org/0000-0002-4923-0582

ENDNOTE

* Computed using the SequenceMatcher routine in python that matches the longest contiguous matching sub-sequence that does not contain any unwanted (or junk) elements. http://opensmiles.org/spec/open-smiles-2-grammar.html

REFERENCES

1. Segler MH, Waller MP. Neural-symbolic machine learning for retrosynthesis and reaction prediction. *Chem A Eur J*. 2017;23:5966-5971.
2. Lin K, Xu Y, Pei J, Lai L. Automatic retrosynthetic pathway planning using template-free models. *Chem Sci*. 2020;11:3355-3364.

3. Schreck JS, Coley CW, Bishop KJM. Learning retrosynthetic planning through simulated experience. *ACS Cent Sci*. 2019;5:970-981.
4. Schwaller P, Petraglia R, Zullo V, et al. Predicting retrosynthetic pathways using transformer-based models and a hyper-graph exploration strategy. *Chem Sci*. 2020;11:3316-3325.
5. Venkatasubramanian V, Chan K, Caruthers JM. Computer-aided molecular design using genetic algorithms. *Comput Chem Eng*. 1994; 18:833-844.
6. Camarda KV, Maranas CD. Optimization in polymer design using connectivity indices. *Ind Eng Chem Res*. 1999;38:1884-1892.
7. Elton DC, Boukouvalas Z, Fuge MD, Chung PW. Deep learning for molecular design—a review of the state of the art. *Mol Syst Des Eng*. 2019;4:828-849.
8. Ward L, Aykol M, Blaiszik B, et al. Strategies for accelerating the adoption of materials informatics. *MRS Bull*. 2018;43:683-689.
9. Xue D, Balachandran PV, Hogden J, Theiler J, Xue D, Lookman T. Accelerated search for materials with targeted properties by adaptive design. *Nat Commun*. 2016;7:11241.
10. Patra TK, Meenakshisundaram V, Hung J-H, Simmons DS. Neural-network-biased genetic algorithms for materials design: evolutionary algorithms that learn. *ACS Comb Sci*. 2017;19:96-107.
11. Goldsmith BR, Esterhuizen J, Liu J-X, Bartel CJ, Sutton C. Machine learning for heterogeneous catalyst design and discovery. *AIChE J*. 2018;64:2311-2323.
12. Li Z, Wang S, Chin WS, Achenie LE, Xin H. High-throughput screening of bimetallic catalysts enabled by machine learning. *J Mater Chem A*. 2017;5:24131-24138.
13. Sanchez-Lengeling B, Aspuru-Guzik A. Inverse molecular design using machine learning: generative models for matter engineering. *Science*. 2018;361:360-365.
14. Nishijima M, Ootani T, Kamimura Y, et al. Accelerated discovery of cathode materials with prolonged cycle life for lithium-ion battery. *Nat Commun*. 2014;5:1-7.
15. Vamathevan J, Clark D, Czodrowski P, et al. Applications of machine learning in drug discovery and development. *Nat Rev Drug Discov*. 2019;18:463-477.
16. Gao H, Struble TJ, Coley CW, Wang Y, Green WH, Jensen KF. Using machine learning to predict suitable conditions for organic reactions. *ACS Cent Sci*. 2018;4:1465-1476.
17. Granda JM, Donina L, Dragone V, Long D-L, Cronin L. Controlling an organic synthesis robot with machine learning to search for new reactivity. *Nature*. 2018;559:377-381.
18. Segler MHS, Waller MP. Modelling chemical reasoning to predict and invent reactions. *Chem A Eur J*. 2017;23:6118-6128.
19. Coley CW, Barzilay R, Jaakkola TS, Green WH, Jensen KF. Prediction of organic reaction outcomes using machine learning. *ACS Cent Sci*. 2017;3:434-443.
20. Wei JN, Duvenaud D, Aspuru-Guzik A. Neural networks for the prediction of organic chemistry reactions. *ACS Cent Sci*. 2016;2:725-732.
21. Kayala MA, Baldi P. ReactionPredictor: prediction of complex chemical reactions at the mechanistic level using machine learning. *J Chem Inf Model*. 2012;52:2526-2540.
22. Coley CW, Jin W, Rogers L, et al. A graph-convolutional neural network model for the prediction of chemical reactivity. *Chem Sci*. 2019; 10:370-377.
23. Jin W, Coley CW, Barzilay R., Jaakkola T. Predicting organic reaction outcomes with Weisfeiler-Lehman network. *arXiv:1709.04555*; 2017.
24. Nam J, Kim J Linking the neural machine translation and the prediction of organic chemistry reactions. *arXiv:1612.09529*; 2016.
25. Schwaller P, Gaudin T, Lanyi D, Bekas C, Laino T. "Found in Translation": predicting outcomes of complex organic chemistry reactions using neural sequence-to-sequence models. *Chem Sci*. 2018;9(28):6091-6098.
26. Liu B, Ramsundar B, Kawthekar P, et al. Retrosynthetic reaction prediction using neural sequence-to-sequence models. *ACS Cent Sci*. 2017;3:1103-1113.
27. Schwaller P, Laino T, Gaudin T, et al. Molecular transformer: a model for uncertainty-calibrated chemical reaction prediction. *ACS Cent Sci*. 2019;5:1572-1583.
28. Vaswani A, Shazeer N, Parmar N, et al. In: Guyon I, Luxburg UV, Bengio S, et al., eds. *Advances in Neural Information Processing Systems 30*. Curran Associates Inc.; 2017:5998-6008.
29. Weininger D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J Chem Inf Comput Sci*. 1988;28:31-36.
30. Chomsky N. Three models for the description of language. *IRE Trans Inf Theory*. 1956;2:113-124.
31. Kusner MJ, Paige B, Hernández-Lobato JM Grammar Variational Autoencoder; 2017,
32. Gruber TR. Toward principles for the design of ontologies used for knowledge sharing? *Int J Hum Comput Stud*. 1995;43:907-928.
33. Morbach J, Wiesner A, Marquardt W. OntoCAPE-A (re) usable ontology for computer-aided process engineering. *Comput Chem Eng*. 2009;33:1546-1556.
34. Hailemariam L, Venkatasubramanian V. Purdue ontology for pharmaceutical engineering: part I. conceptual framework. *J Pharm Innov*. 2010;5:88-99.
35. Hailemariam L, Venkatasubramanian V. Purdue ontology for pharmaceutical engineering: part II. Applications. *J Pharm Innov*. 2010;5:139-146.
36. Zhang X, Zhao C, Wang X. A survey on knowledge representation in materials science and engineering: an ontological perspective. *Comput Ind*. 2015;73:8-22.
37. Feldman HJ, Dumontier M, Ling S, Haider N, Hogue CW. CO: a chemical ontology for identification of functional groups and semantic comparison of small molecules. *FEBS Lett*. 2005;579:4685-4691.
38. Jurafsky D, Martin JH. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. 1st ed. Upper Saddle River, NJ, USA: Prentice Hall PTR; 2000.
39. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*; 2014.
40. Lowe DM. Extraction of chemical structures and reactions from the literature (Doctoral thesis). University of Cambridge. https://doi.org/10.17863/CAM.16293; 2012.
41. Kingma DP, Ba J. Adam: a method for stochastic optimization. *arXiv: 1412.6980*; 2014.
42. Papineni K, Roukos S, Ward T, Zhu W-J BLEU: a method for automatic evaluation of machine translation. Paper presented at: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics; 2002:311-318.

APPENDIX

SMILES GRAMMAR

The SMILES grammar used in this works is a subset of the official OpenSMILES* specification[29] 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 A1. Similar to the grammar production rules used in Kusner et a l.[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 → CHAIN` and `NOTHING → 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.

TABLE A1 SMILES grammar used in GO-PRO

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

Abbreviations: GO-PRO, Grammar Ontology-based Prediction of Reaction Outcomes; SMILES, simplified molecular-input line-entry system.

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.