Retrosynthesis Prediction using Grammar-based Neural Machine Translation: An Information-Theoretic Approach

Vipul Mann, Venkat Venkatasubramanian*

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

Abstract

Retrosynthetic prediction is one of the main challenges in chemical synthesis because it requires a search over the space of plausible chemical reactions that often results in complex, multi-step, branched synthesis trees for even moderately complex organic reactions. Here, we propose an approach that performs single-step retrosynthesis prediction using SMILES grammar-based representations in a neural machine translation framework. Information-theoretic analyses of such grammar-representations reveal that they are superior to SMILES representations and are better-suited for machine learning tasks due to their underlying redundancy and high information capacity. We report the top-1 prediction accuracy of 43.8% (syntactic validity 95.6%) and maximal fragment (MaxFrag) accuracy of 50.4%. Comparing our model's performance with previous work that used character-based SMILES representations demonstrate significant reduction in grammatically invalid predictions and improved prediction accuracy. Fewer invalid predictions for both known and unknown reaction class scenarios demonstrate the model's ability to learn the underlying SMILES grammar efficiently.

1. Introduction

One of the important challenges in computational chemistry is the retrosynthetic analysis of desired molecules that satisfy property constraints, subject to the commercial availability of the precursors and the feasibility of the chemical reactions required for their synthesis. The immense interest in this problem over the recent years could be attributed to its practical applications across areas such as drug discovery, synthesis of novel organic compounds, and improvements in

*Corresponding author

Email addresses: vm2583@columbia.edu (Vipul Mann), venkat@columbia.edu (Venkat Venkatasubramanian)

the reactions pathways from a commercial, social, or economic viability standpoint. The industrial applications of retrosynthetic analysis include automobiles, petrochemicals, specialty chemicals, and polymer science, with a great potential to revolutionize the entire industry if the right compound could be synthesized.

Retrosynthetic analysis often involves evaluating many potential candidate reaction pathways and molecules at multiple stages of the reaction, resulting in complex retrosynthesis trees that need to be searched and parsed efficiently. Computational approaches could significantly aid the chemist in solving different aspects of the retrosynthesis problem, such as the graph-theoretic search methodologies for efficient tree traversal to identify feasible reaction pathways, dictionary-based methods to evaluate a large search-space of precursors, and chemistry-driven heuristics to eliminate practically infeasible routes. Multi-step retosynthesis is usually formulated as a fundamentally different problem compared to the single-step retrosynthesis that we study in our work, and often involves either using efficient search techniques combined with one-step forward synthesis models, or using a sequence of single-step reverse transformations informed by chemistry-based heuristics.

One of the first attempts that leveraged computational tools and formalized the retrosynthesis problem was LHASA proposed in [1]. This framework used logic and chemistry rules in the form of heuristics and transformations along with a chemical programming language to solve the retrosynthesis problem. Several subsequent approaches were proposed that utilized rule-based expert-systems [2–11], with a few of them combining network theory to discover that chemistry networks follow scale-free properties, a ubiquitous class of networks reported to be optimal in several other areas [12–15]. However, such approaches were hard to scale beyond interesting prototypes as they required great human effort and expertise to develop [16].

However, in recent years, the massive surge in computational capabilities combined with significant advances in machine learning have resulted in a renewed attack on this problem. This includes approaches that combine neural network models with known chemistry knowledge encoded in the form of reaction templates – e.g., Segler and Waller [17] leveraged neural networks for selecting the reactivity centers and most suitable transformations; Wei et al. [18] predicted reaction types and used Smiles Molecular Arbitrary Target Specification (SMARTS) templates for predicting the likely products given a set of reactants and reagents, and Coley et al. [19] proposed selecting the suitable edit-based transformations in a reaction using reaction templates. Such methods, however, again address only certain limitations of the rules-based systems and the inherent limitation of the lack of their ability to suggest novel chemical reactions and a bias towards the common reaction types still exist.

This is overcome in purely data-driven approaches that use sophisticated machine learning architectures to learn the complex non-linear dynamics of a chemical reaction – both in the forward and the backward directions – primarily by modeling the chemical representations. This includes the neural sequence-to-sequence (or seq2seq) models introduced for the forward reaction prediction in [20] and the retrosynthetic prediction in [21] that formulated the reaction prediction task as a sequence modeling problem. Other recent efforts for the retrosynthesis task include a seq2seq approach combined with a Monte Carlo tree search [22] and various transformer model-based approaches [23–29].

Even though the prediction accuracy has significantly improved due to the increased complexity of model architectures, prior chemistry knowledge in such frameworks is still missing. The incorporation of this knowledge should, in principle, improve the model performance on out-of-sample examples. All previous works in this area use SMILES representations of molecules, treating them as merely character-based strings, except for the recent work by Ucak et al. [30] that used substructure-based representations but suffered from lower prediction accuracy. In our earlier work on the forward prediction problem [24, 25], we demonstrated that incorporating chemical and structural information about molecules ensures that the model learns the underlying chemical transformations with significantly fewer training parameters. As an extension of that work, we propose here a framework for solving the retrosynthesis problem using the rich, SMILES grammar-based representation of molecules and highlight the inherent benefits of such representations – both from an information-theoretic and model performance standpoint.

The rest of the paper is organized as follows: In Section 2, we formally define the retrosynthesis prediction problem as a sequence modeling task in the machine translation framework and present an overview of the methods underlying our work, such as the SMILES grammar, the transformer architecture and the beam search decoding procedure in Section 3. In Section 4, we present an information-theoretic analysis of the proposed grammar-representations and contrast them with the other representations (SMILES and molecular formula) to highlight the differences and quantify the advantages of using the underlying chemical structural information. The standard reaction dataset and the model training aspects of our work are presented in Section 5. The evaluation metrics used for assessing our model's performance, the results on the USPTO 50K reactions dataset, comparison with other works, and the limitations and future work in this direction are presented in Section 6. Finally, the concluding remarks summarizing the major contributions of this work appear in Section 7.

2. Problem formulation and objectives

We formulate the retrosynthesis prediction problem as a sequence modeling task and use a machine translation framework for predicting the precursors for a given target molecule. The objective is to translate a set of input tokens corresponding to the product molecule to an output sequence of tokens corresponding to the precursor molecules. The input sequence may be optionally prepended with an identifier that indicates the reaction class. To allow the model to differentiate between the different precursors (reactants), a separate identifier token is used to indicate the end of the representation of a given precursor and the start of another. This framework is depicted in Figure 1.

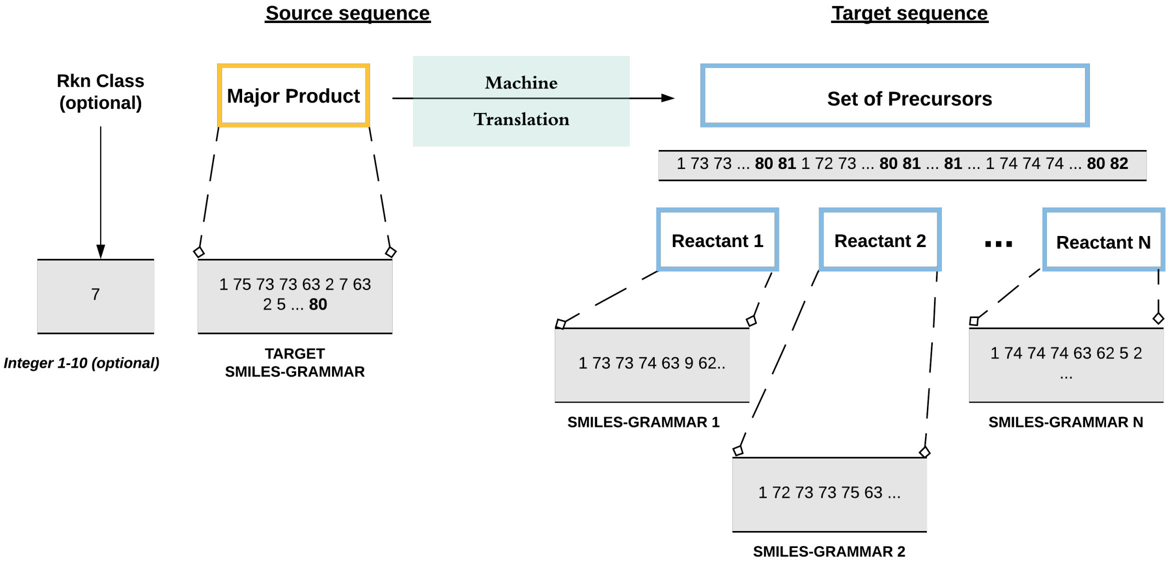


Figure 1: The single-step retrosynthesis prediction problem formulation using machine translation. The reaction class information is optional.

In this framework, the participating product and reactants in a given reaction are represented using their corresponding grammar-based representation described in detail in Section 3.1. The representation starts with the token ‘1’ and ends with the token ‘80’ for all the molecules, the token ‘81’ separates multiple reactants, and the token ‘82’ signifies the end of all the precursor representations. The other identifiers (or tokens) correspond to the sequence of production rules required to obtain the given SMILES string, using the grammar productions described in Table 12 in the Appendix. The sequence modeling task is performed using a transformer model, a state-of-the-art architecture for sequence modeling [31].

3. Methods

In this section, we describe the methods involving our approach, namely the SMILES grammar-based representations used for encoding molecules, the transformer architecture used for the sequence modeling task, and the beam search decoding procedure used for generating a set of most likely target sequences for a given input sequence.

3.1. SMILES grammar

One of the first works that attempted to formalize natural language through context-free grammars (CFGs) was proposed by Noam Chomsky [32] that was based on the idea that a group of words could be thought of as belonging to a constituent unit and that different constituent units could be grouped, hierarchically, to convey a given meaning. Formally, a context-free grammar could be thought of as a set of production rules that define the transformation of a set of non-terminal symbols to terminal symbols that correspond to strings with meaning in the natural language. In addition, there is a designated start symbol that indicates the start of a sentence. Therefore, a CFG consists of the following elements: S, a designated start symbol; Σ, the set of terminal symbols; N, the set of non-terminal symbols; and R, the set of production rules of the form $\text{A} \longrightarrow \beta$ where $\text{A} \in \text{N}$ is non-terminal and $\beta \in \Sigma$ is a terminal symbol.

A similar grammar for the SMILES representation of molecules also exists [33] where the individual tokens in the SMILES string represent the terminal symbols that could be obtained through the sequential application of a set of production rules on the non-terminal symbols. Consider, for example, a subset of the official SMILES grammar presented in Table 1. The equivalent symbols similar to CFG for this grammar are:

- S: `SMILES`

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

Table 1: Reduced 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

We leverage such underlying grammar to assign structure to a given SMILES string and derive from such structures the grammar-based representations. Consider benzene, with the SMILES string representation given by C1=CC=CC=C1. This representation could be obtained by applying the set of production rules in Table 1 sequentially with the corresponding parses-tree shown in Figure 2. The grammar-representation that we work with, originally proposed in our earlier work [24], is obtained by extracting production rules from the parse-tree by parsing it in a bottom-up-left-corner strategy, i.e., starting at the top and going down the left-most branch, then coming back up to parse the immediate right branch, and so on until the entire tree is parsed. The grammar representation thus obtained corresponding to the parse-tree for benzene is given in the figure caption.

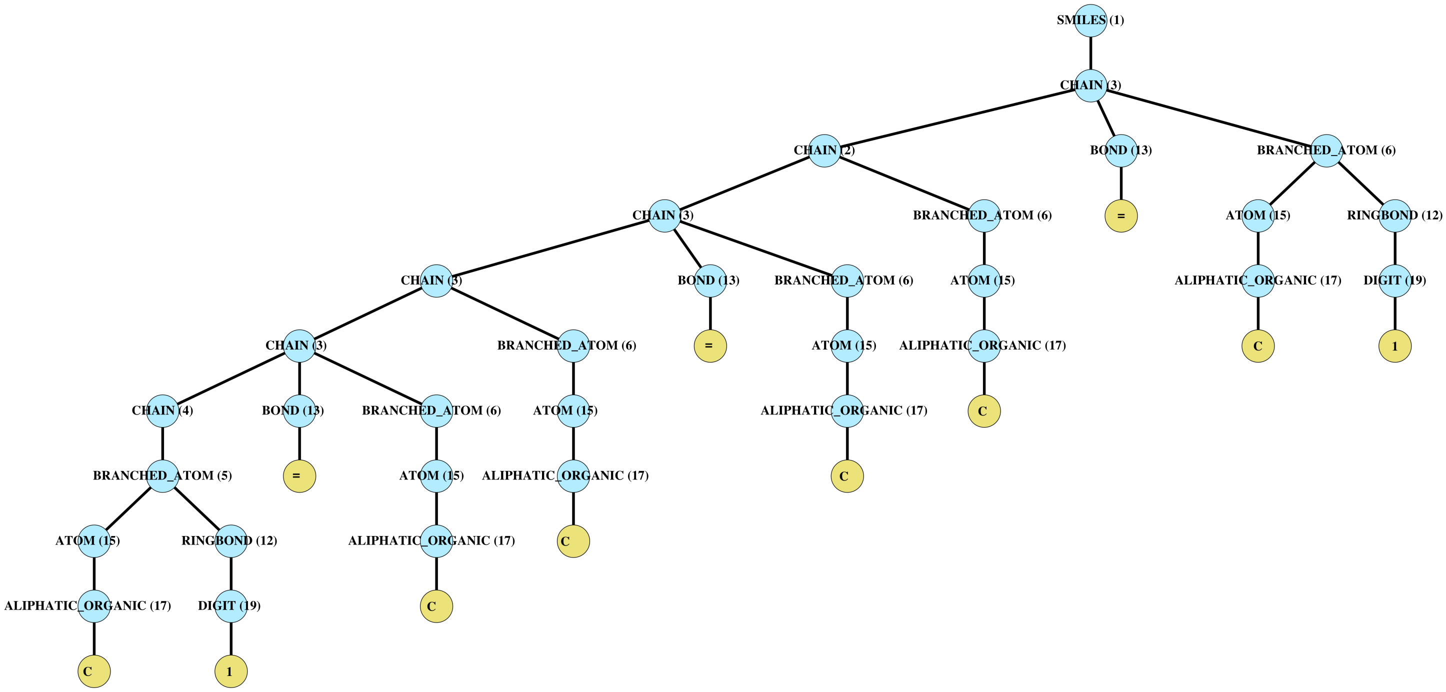


Figure 2: The parse-tree obtained for benzene with SMILES string representation as `C1=CC=CC=C1`. The production rules from Table 1 applied at each stage are indicated next to the non-terminal symbols. Parsing this tree in a bottom-up-left-corner strategy gives rise to the grammar-representation given by: $1, 3, 2, 3, 3, 3, 4, 5, 15, 17, 12, 19, 13, 6, 15, 17, 6, 15, 17, 13, 6, 15, 17, 6, 15, 17, 13, 6, 15, 17, 12, 19$

Clearly, as compared to a purely character-based SMILES string representation consisting merely of the tokens ‘C’, ‘1’, ‘=’, ‘C’, ‘C’, ‘=’, ‘C’, ‘C’, ‘=’, ‘C’, ‘1’, without any additional information conveying the relationships between the tokens, the grammar-based representations are significantly richer, incorporate chemical and structural information, and contain hierarchical information about the underlying chemistry. This is leveraged by the model architecture for modeling the underlying SMILES grammar. We have shown that these representations are more efficient in modeling the underlying chemistry and eliminate overparameterization in complex machine learning architectures [24]. We present an information-theoretic analysis of the grammar representations and the text-based representations in Section 4 to establish the fundamental superiority of the grammar representations compared to other text-based representations such as SMILES.

3.2. Sequence-to-sequence models

We model the reaction prediction problem as a sequence modeling task that involves mapping the input sequence to a sequence of tokens corresponding to the output sequence. This framework has been used in recent years and has shown a significant promise in reaction modeling. We use the state-of-the-art model in this area, known as the transformer framework, proposed in [31].

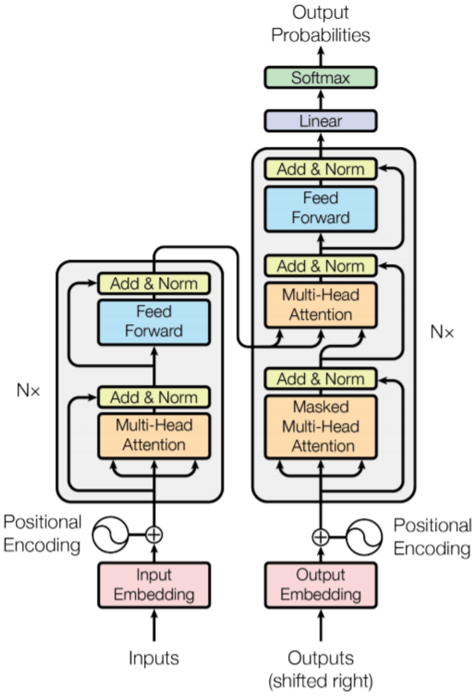


Figure 3: The encoder-decoder model architecture of a transformer as depicted in [31]

The transformer framework, shown in Figure 3, consists of an encoder-decoder architecture where the encoder maps the input sequence to a latent space, and the decoder decodes from the latent space in an autoregressive manner, one element at a time, to give rise to the output sequence. The positional encodings in a transformer encode the position of a given word (or token) in the sequence to a high dimensional vector space, getting rid of recurrent or convolution operations that significantly improved the computational complexity of training the model architecture. These mappings are characterized by sines and cosines of different frequencies, given by

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

An attention mechanism lets the transformer model relationships between groups of words in an input sequence at different stages of the network. The attention-mechanism used in [31] 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 which are used to represent a given word and the corresponding key-value pairs for computing the attention function. The query, key, and value vectors are obtained from the output of the preceding layers in the network. The attention-score 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

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2}$$

where Q, K, and V are the matrices of query, key, and values vectors, respectively. The attention score computed above determines the importance of 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,

$$MultiHead(Q, K, V) = Concat(head_1, head_2, \ldots, head_h)W^O \tag{3}$$

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. The reader is referred to [31] for further details on the transformer model architecture.

3.3. Beam search

In order to generate the output sequences in a transformer framework, a decoding procedure is used that decodes from the latent space in an autoregressive manner, with the current prediction as input while decoding the next token. Therefore, the decoding procedure could either use a greedy strategy that involves selecting the token with the maximum likelihood at each stage for decoding the next token, generating a single most-likely sequence in the end; or on the other hand, it could employ a beam search procedure that decodes a set of top-B tokens at each stage based on their likelihood and return them as the model output. We follow the latter approach for decoding. This allows us to evaluate our model's performance more extensively and compare it with the top-K accuracy reported in other similar works in this area. A schematic of the beam-search decoding procedure used in our work is shown in Figure 4.

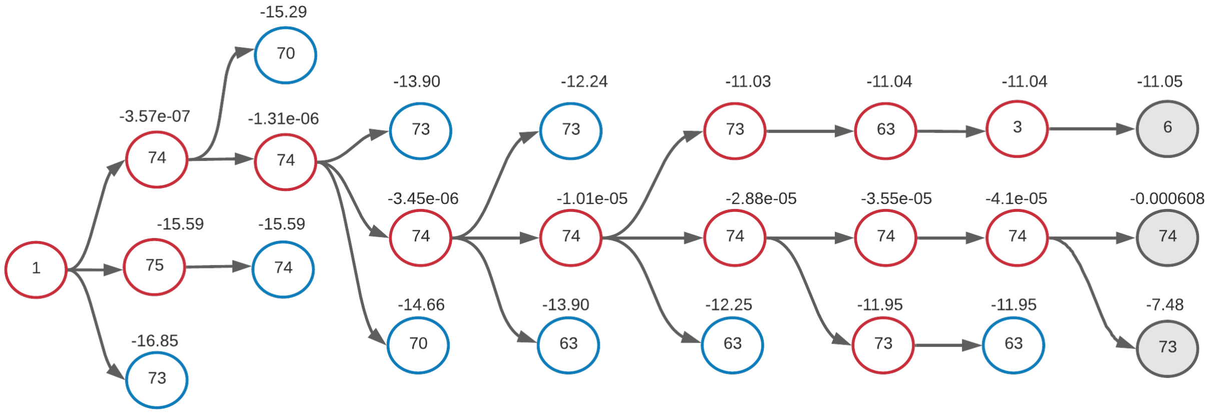


Figure 4: A partially completed beam search output for a beam width of 3 for a reference input. At each stage, the most likely grammar-rules are predicted, that are used as input to decode the next most likely set of grammar-rules, and so on until the entire sequence of grammar-rules corresponding to a given SMILES string is reconstructed. The log-likelihood values are indicated above each node in the schematic.

4. Information-Theoretic Analysis of Chemical Representations

Before discussing the model training aspects, we demonstrate the richness of the proposed grammar-based representations using an information-theoretic framework. We compare the information capacity, information gain, and redundancy characterizing the various symbols-based chemical representations, namely, molecular formula, SMILES, and grammar representations. We first provide a brief overview of the relevant information-theoretic concepts and their intuition in the next section, followed by their application to chemical representations and quantify the superiority of grammar-based representations from an information-theoretic standpoint.

4.1. Shannon Entropy and Information Content

The development and formalization of information theory, mainly by Claude Shannon in [34], offered a mathematical definition of the *amount of information* communicated between any two components or channels of a given system. The primary motivation was the fundamental problem of decoding a source message passing through a noisy channel, either exactly or approximately, at any other point in the communication system. However, the applications and adaptations of it are not limited to communication systems alone but have had far-reaching consequences across most fields of science and engineering.

The Shannon entropy for a given probability distribution $p(x)$ of a random variable x is defined as,

$$H = -\sum_{i=1}^{M} p(x_i) \log_2 p(x_i) \tag{4}$$

where $p(x)$ is the probability mass function of x with M possible values. This is equivalent to the expected value of the Shannon information or self-information of a variable and is measured in units of *bits per symbol*. There is a direct correspondence between the amount of information in a message and the degree of uncertainty that is associated with it. That is, if a system can exist in one of a very large number of possible states, then there is a great amount of uncertainty associated with its state as opposed to another system that can exist only in a handful states. Therefore, the amount of information required is more for the former than the latter.

Consider the two extremes of zero-information content and maximum information content. The Shannon entropy in Equation 4 attains a value of zero when the probability $p(x_i)$ of a x_i attaining a given value is 1 meaning that the outcome or the value that x_i could take is known with complete certainty, and hence, there is no information content (or gain) associated with knowing its value explicitly. On the other hand, when x_i could take any of the possible values with equal probability, i.e., $p(x_i) = 1/M$ where M is the total number of possible values that the symbols in the source message could take, the information content is maximized and is equal to $\log_2 M$. This implies that in such a scenario, specifying the value of a given bit in the sequence would result in the maximum information gain when compared to any other scenario.

The generalization of Equation 4 when several random variables $X_1, X_2, \ldots, X_n$ are present is given by the joint Shannon entropy as,

$$H(X_1, X_2, \ldots, X_n) = -\sum_{x_1, x_2, \ldots, x_n} p(x_1, x_2, \ldots, x_n) \log_2 p(x_1, x_2, \ldots, x_n) \tag{5}$$

The joint entropy in Equation 5 could be interpreted as an information measure corresponding to multiple random variables presented simultaneously. Similarly, the conditional entropy that quantifies the information content of a given random variable X_1 conditioned on a set of other random variable $X_2, X_3, \ldots, X_n$, is given as

$$H(X_1 \mid X_2, X_3, \ldots, X_n) = -\sum_{x_1, x_2, \ldots, x_n} p(x_1, x_2, x_3, \ldots, x_n) \log_2 p(x_1 \mid x_2, x_3, \ldots, x_n) \tag{6}$$

The conditional entropy could be used to measure the information gain when partial information or context of other random variables is known. Equipped with information theory concepts, we now apply these information measures to chemical systems and molecules.

4.2. Information theory and Chemical Representations

Studies in chemical information theory [35] have demonstrated the promise of entropic perspective in chemistry [36–39]. We analyze various chemical representations, namely, the SMILES representations, molecular formulas, and our proposed SMILES grammar-based representations from the perspective of Shannon entropy. We quantify the superiority of certain representations when compared to the others and highlight their inherent benefits when used in machine learning algorithms.

In our framework, we consider the individual tokens in various representations as random variables that contain *bits of information* required to reconstruct a given molecule. The representations are therefore a sequence of random variables, $X_1, X_2, \ldots, X_n$, where n is the length of the representation for a given molecule and X_i could take any of the M possible tokens defined in the vocabulary of the representation. For instance, consider the earlier example of benzene from Section 3.1. The corresponding random variables for each of the three representations is given by,

- Molecular formula (C_6H_6): $X_i^{Mo} \in \{‘C’, ‘6’, ‘H’\}$, where $M = 3, n = 4$
- SMILES ($C1 = CC = CC = C1$): $X_i^{S} \in \{‘C’, ‘1’, ‘=’\}$, where $M = 3, n = 11$
- Grammar[1]$(1, 3, 2, 3, \ldots, 12, 19)$: $X_i^{G} \in \{1, 2, 3, 4, 5, 6, 12, 13, 15, 17, 19\}$, where $M = 11, n = 32$

Defining the random variables and computing their probability distributions over all the molecules in the dataset, we compute the corresponding information measures using Shannon entropy in Equation 4. Since our objective is to quantify the information capacity for an entire representation instead of certain specific molecules, this distribution is computed over all the possible lengths of representations, n, in the dataset. Similarly, the conditional information measure in Equation 6 could be computed using the conditional distribution of random variables based on the co-occurrence matrices (up to a given order) of the random variables in the database. The order indicates the number of random variables under consideration, with $\eta - 1$ conditioned random variables for an order of η. An order $\eta = 1$ corresponds to Shannon entropy (Equation 4), order $\eta = 0$ corresponds to Shannon entropy when the random variables follow a uniform distribution, and orders $\eta > 1$ correspond to conditional entropy with conditioning on $\eta - 1$ random variables (Equation 6).

The Shannon entropy and conditional information measures, presented in Figure 5 and Table 2, are computed (at various orders) using Equations 4 and 6, respectively. The USPTO 50K test set is used to estimate the required (conditional) probability distributions for the three representations (SMILES, grammar, and molecular formula) based on the co-occurrence matrices, conditioned on a given number of tokens according to the order of conditioning. The probability distributions for the random variables are computed using the three representations for all the molecules in the test-set of the USPTO 50K reaction dataset to limit computational requirements, especially for calculating the conditional distributions. We evaluate the maximum conditional distribution up to an order of $\eta = 5$. The molecular formulas are extracted from the SMILES representations of molecules using the ‘rcdk’ library in R.

[1] using the representative grammar in 1

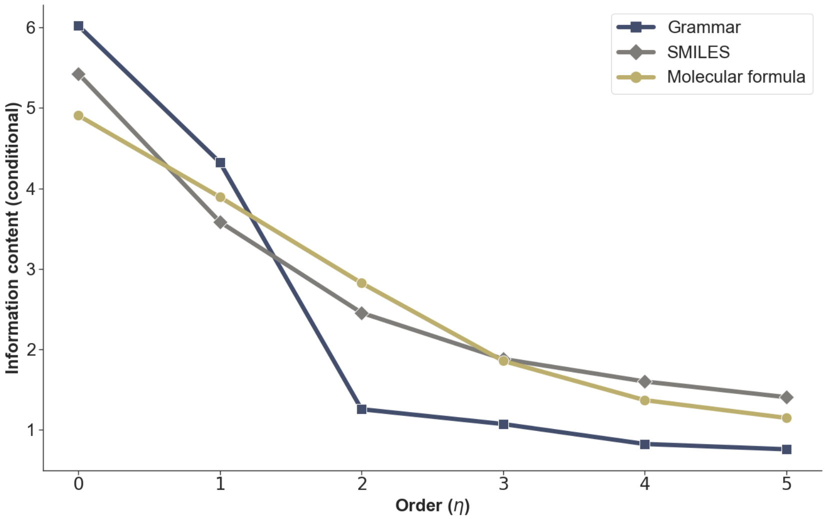


Figure 5: Information content vs order of conditioning (η) for the three representations

Table 2: Information content (i_η) for various orders of conditioning (η) for the three representations

	SMILES	Grammar	Molecular Formula
i_0	5.426	6.022	4.906
i_1	3.583	4.322	3.891
i_2	2.453	1.254	2.823
i_3	1.879	1.070	1.855
i_4	1.599	0.822	1.367
i_5	1.404	0.756	1.146

It follows from our discussion in the earlier section that the maximum information (corresponding to i_0) is achieved when the random variables follow a uniform distribution and all the bits have the same probability $(1/M)$ of taking a given value. Thus, i_0 is independent of the dataset under consideration and is purely a property of the representation that is indicative of its information storing capacity. Based on Figure 5, the grammar-representations have much higher information capacity, followed by the SMILES representation and then the molecular formulas, highlighting the theoretically high information capacity of grammar representations.

When the order of analysis is increased to 1, the information capacity decreases for all the representations, indicating that the underlying probability distributions are far from uniform, with certain values more likely than others. This is expected since in any chemical representation, the identifiers for atoms such as C and H are significantly more likely to occur when compared to others such as F or B. It could be inferred through the probability versus identifier index plot depicted in Figure 6 that the SMILES and molecular formula representations are much more skewed, with a majority of the identifiers occurring much more frequently than the others. On the other hand, the grammar-based representations' identifiers exhibit a much smoother and slower decay, indicating more evenly distributed probabilities for the identifiers. This validates the richness of grammar-representations due to the incorporation of structural-hierarchy, an argument that we made qualitatively in our earlier work [24].

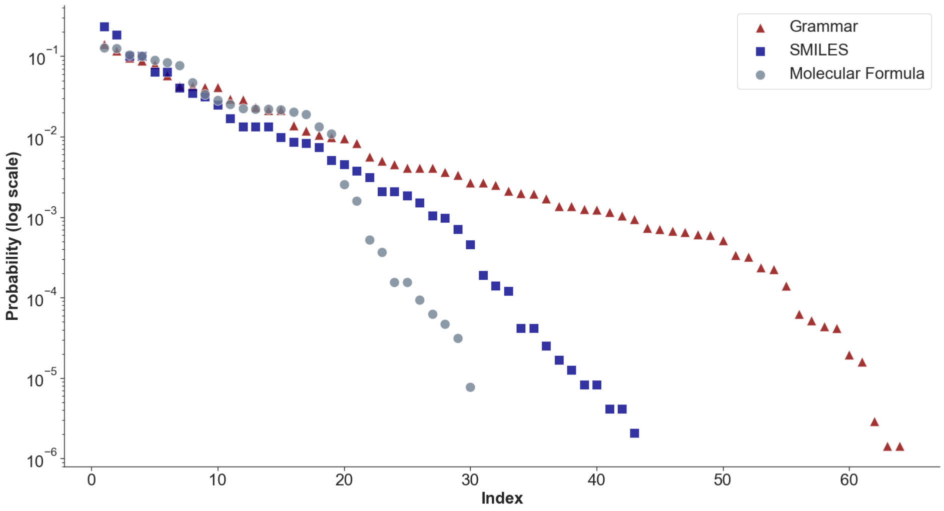


Figure 6: Probability of occurrence of a given token versus the sorted index

As the order of conditioning while computing the information measure is increased to $\eta = 2$, a drastic decrease in the information content is observed for grammar-representations, and the conditional information content remains significantly lesser than the other representations even for higher values of η. This could be attributed to the in-built redundancy in the grammar-representations incorporated by means of a hierarchical sequence of production rules encoded in a molecule's representation. This transforms into lower values of conditional probabilities when an identifier's context in terms of the preceding tokens is known. Qualitatively, this means that when the context of a token is provided, the uncertainty associated with the possible values it could take is much lesser than its equivalent in the SMILES representation and molecular formula-based representations.

It is interesting to also note from Figure 5 that the conditional information content plots seem to intercept twice for the SMILES and molecular formula representations, which could possibly be explained as follows – the first intercept is due to the relative differences in the theoretical information content (i_0) and the actual information content (i_1) computed using the conditional probabilities from the database, indicating that the conditional probabilities at order 1 are much more skewed for the SMILES representation (translating to lower entropy) since it has more tokens that are repeated compared to molecular formulas; and the second intercept at order 3 could be due to the trade-off between the number of tokens and redundancy in the representations where the higher number of tokens for SMILES begin to contribute more towards the conditional entropy (uncertainty) even after a reduction in entropy due to the partial knowledge of the context (preceding tokens). In contrast, the grammar-based representation consistently has the highest theoretical (i_0) and actual (i_1) information content, and the lowest conditional entropy (highest redundancy) beyond order 2. This clearly demonstrates the ability of grammar-based representations in overcoming the associated trade-off between higher number of tokens and redundancy that the SMILES and molecular formula-based representations suffer from.

In summary, the underlying redundancy in grammar-representations, indicated by i_η with $\eta \geq 2$, could be leveraged by machine learning algorithms that model the long and short-range dependencies between tokens in a given sequence, such as the class of sequence-to-sequence models used in our work. In addition, the higher information-storage capacity of these representations, as indicated by i_0 and i_1, implies that they are much richer when compared to the other representations and therefore contain additional *bits of information* that is lacking in the other representations and could be crucial for the adequate differentiation between molecules in the latent space. There are other representations such as International Chemical Identifier (InChI) that are used to represent molecules and performing a similar information-theoretic analysis on such representations would part of our future work in this direction.

5. Data and Model training

We demonstrate our model's performance using a standard retrosynthesis prediction dataset which is a filtered dataset derived from the text extraction work done on US Patents and Trademark Office's (USPTO) database [40] and further classified into ten different reaction classes [41]. The filtered dataset contains only the reactants and products, with the reagent information removed and the SMILES strings canonicalized. Further, similar to [21], the multiple product reactions are split into multiple reactions so that each reaction contains only a single major product. This dataset is referred to as the USPTO 50K dataset in the literature.

In order to use our approach, we encode the SMILES strings corresponding to all the molecules in the database into their equivalent grammar representations as described in Section 3.1. This implies that since we are working with a subset of the official OpenSMILES grammar, certain molecules that are not in grammar are skipped and therefore are not included in the model training stage. Table 3 summarizes the reaction database with the number of reactions that are in grammar along with the train, validation, and test-set splits. Table 4 summarizes the distribution of the various reactions across the 10 reaction classes.

Table 3: An overview of the retrosynthesis dataset used in our work

Dataset	**train**	**valid**	**test**	**total**
USPTO 50K				
with (sanitized) molecules	40,029	5,004	5,004	50,037
in grammar	38,995	4,861	4,861	48,717

Table 4: Distribution of reactions across different reaction classes that are in-grammar

Reaction class	Reaction name	train	valid	test	**total**
1	Heteroatom alkylation and arylation	11,886	1,476	1,478	14,840
2	Acylation and related processes	9,358	1,165	1,169	11,698
3	C – C bond formation	4,324	544	539	5,407
4	Heterocycle formation	710	89	90	889
5	Protections	513	64	62	639
6	Deprotections	6,357	796	789	7,942
7	Reductions	3,607	448	452	4,507
8	Oxidations	629	80	79	788
9	Functional group interconversion (FGI)	1,434	176	180	1,790
10	Functional group addition (FGA)	177	23	23	223

Since the retrosynthesis prediction task involves predicting a set of precursors that could be used for obtaining a given product molecule, we define identifiers that distinguish the various reactant molecules (grammar-representation) from each other and also indicate the end of the set. These two additional tokens convey to the model the separation between various precursors' representations and also the end of the entire set of precursors. The reaction class identifiers are appended at the start of the source (product) molecule's representation while evaluating the model performance under known reaction type scenarios. This additional step is skipped when the model performance is evaluated for the unknown reaction type scenario.. A schematic for this is shown in Figure 7.

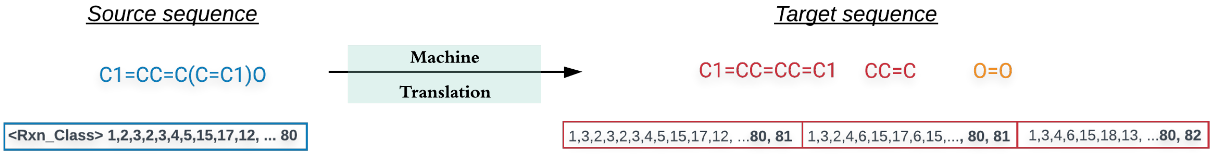


Figure 7: The retrosynthesis reaction encoding strategy used in the machine translation framework. The identifier '80' indicates the end of a given molecule's grammar-representation, '81' indicates the separation between two precursor molecules, and '82' indicates the end of the entire set of precursor molecules. The additional token indicating the reaction type is optional and we report the model performance under both the scenarios with known and unknown reaction classes.

We train the transformer model for this task using a cross-entropy-based loss function that minimizes the sequence-to-sequence translation error. 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 cyclic learning rate schedule that is characterized by a fixed number of warmup steps given by

$$lr = d_{model}^{-0.5} . min(step\_num^{-0.5}, step\_num * warmup\_steps^{-0.5}) \tag{7}$$

where *d_model* is the embedding dimension (positional). At the training stage, to avoid overfitting, a dropout layer is used for both the feed-forward networks and the attention-mechanism, for the encoder and the decoder. A masking approach similar to [43] is used for generating the output SMILES strings from the decoded grammar-representation. A loss function based on sparse categorical cross-entropy between the predicted and actual target sequences is minimized. The possible and the best hyperparameters identified for the model are given in Table 5. The lengths of the input and output representations to the model are fixed at 301 and 900, respectively.

Both the models were trained using TensorFlow 2.1 and python 3.7 for 12 cycles (∼700 epochs). For generating the parse-trees and extracting grammar-based features, we used the Natural Language ToolKit (NLTK) 3.4.5 library. The molecular datasets were processed using the 2019 release of the RDKit library.

Table 5: Possible and best hyperparameter values for the transformer model architecture described in Figure 3

Hyperparameter	**Possible values**	**Final model**
Embedding dimensions	64, 128, 256	256
Attention heads	4, 8, 16	8
Feedforward network units	512, 1024, 2048	512
Number of layers	4, 6	4
Dropout	0.1, 0.2	0.1
Warmup steps	4k, 8k, 12k	8k

6. Results and Discussion

In this section, we define the performance metrics, evaluate the model's performance on the test-set of the USPTO 50K dataset, and benchmark the performance of our approach against other similar works in this area, highlighting the advantages and limitations of this framework.

6.1. Evaluation metrics

We evaluate our model's performance using the following metrics – accuracy, which captures the ability to perfectly predict all the precursor molecules; fractional accuracy, which indicates the fraction of accurately predicted precursors from the set of molecules in the ground truth; and syntactic validity, meaning the percentage of grammatically valid predictions. In addition, we also compute the accuracy of prediction of the Maximal Fragment or MaxFrag [29] indicating the prediction accuracy of the longest reactant involved and report the average BLEU (bilingual evaluation understudy) [44] and similarity scores for this maximal fragment. The BLEU score is a standard metric used for evaluation of the quality of machine-translated texts against a reference translation, and the similarity scores[2] are computed using the similarities between the string substructures of the predictions and the ground truth. These metrics are reported for three example predictions in Figure 8.

6.2. Results on USPTO 50K dataset

The performance evaluation measures computed on the test set of the USPTO 50K dataset are presented in Tables 6 and 8 for the known reaction class scenario and in Tables 7 and 9 for the scenario when reaction classes are not known. We observe from Table 6 that though the top-10 accuracy is 66.6%, the fractional accuracy at 73.7% is much higher and indicates that a major fraction of the ground truth reactants is accurately predicted across reactions. The syntactic validity is as high as 95.6% for the top-1 predictions and 90.4% for the top-10 predictions. The decreasing trend in syntactic validity is expected since as the number of predictions increases, the invalid predictions go up because of the model's susceptibility to decode grammatically invalid strings.

Performance measure	top-1	top-3	top-5	top-10
Accuracy	43.8	57.2	61.4	66.6
Fractional accuracy	53.8	65.4	69.2	73.7
Syntactic validity	95.6	92.8	91.6	90.4

Table 6: Accuracy, fractional accuracy, and syntactic validity on the test set with known reaction class

Performance measure	top-1	top-3	top-5	top-10
Accuracy	32.1	44.3	48.9	54.0
Fractional accuracy	39.6	51.5	56.2	61.8
Syntactic validity	94.9	92.6	91.6	90.3

Table 7: Accuracy, fractional accuracy, and syntactic validity on the test set with unknown reaction class

The similarity scores in Table 8 indicate that the MaxFrag precursor is predicted with a top-10 accuracy of over 70% and a similarity score of over 90%, highlighting the model's ability to correctly identify the characteristics of the most critical molecule (in classical retrosynthesis) with a fairly high degree of accuracy. The corresponding BLEU scores also indicate the good quality of translation that is achieved for the MaxFrag molecule.

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

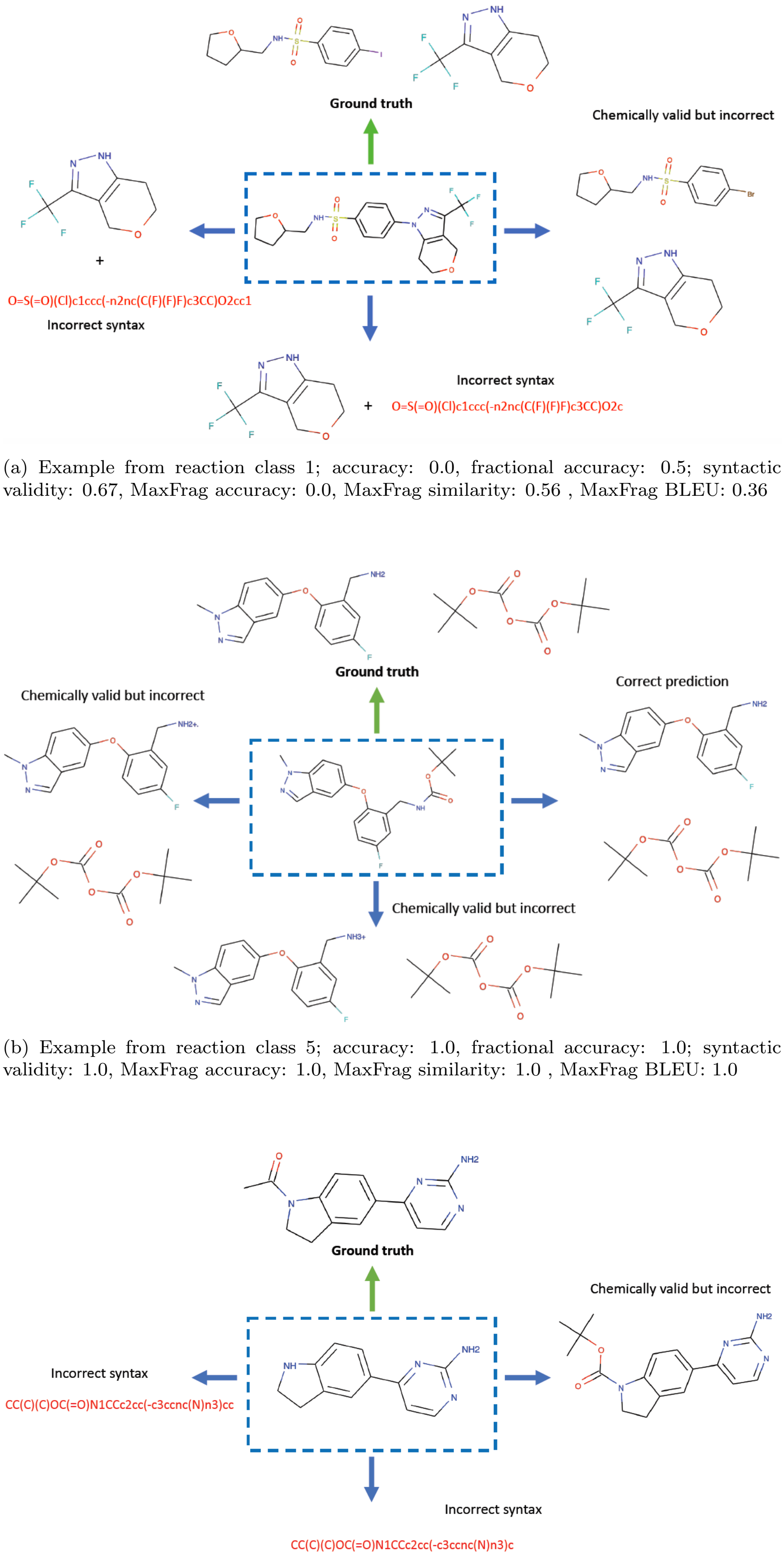
Performance measure	top-1	top-3	top-5	top-10
MaxFrag accuracy	50.4	62.1	65.7	70.2
BLEU score	74.8	83.4	85.2	87.4
Similarity score	80.0	87.2	88.6	90.2

Table 8: MaxFrag accuracy and the corresponding BLEU and similarity scores on the test set with known reaction class

Performance measure	top-1	top-3	top-5	top-10
MaxFrag accuracy	38.1	49.1	53.2	58.4
BLEU score	67.5	76.0	78.3	81.1
Similarity score	75.7	82.2	83.8	85.7

Table 9: MaxFrag accuracy and the corresponding BLEU and similarity scores on the test set with unknown reaction class

Few of the example top-3 predictions along with the prediction inaccuracies and performance metrics are presented in Figure 8.


Ground truth
Chemically valid but incorrect
O=S(=O)(Cl)c1ccc(-n2nc(C(F)(F)F)c3CC)O2cc1
Incorrect syntax
Incorrect syntax
O=S(=O)(Cl)c1ccc(-n2nc(C(F)(F)F)c3CC)O2c
Ground truth
Chemically valid but incorrect
Correct prediction
Chemically valid but incorrect
Ground truth
Chemically valid but incorrect
Incorrect syntax
CC(C)(C)OC(=O)N1CCc2cc(-c3ccnc(N)n3)cc
Incorrect syntax
CC(C)(C)OC(=O)N1CCc2cc(-c3ccnc(N)n3)c

(a) Example from reaction class 1; accuracy: 0.0, fractional accuracy: 0.5; syntactic validity: 0.67, MaxFrag accuracy: 0.0, MaxFrag similarity: 0.56 , MaxFrag BLEU: 0.36

(b) Example from reaction class 5; accuracy: 1.0, fractional accuracy: 1.0; syntactic validity: 1.0, MaxFrag accuracy: 1.0, MaxFrag similarity: 1.0 , MaxFrag BLEU: 1.0

(c) Example from reaction class 6; accuracy: 0.0, fractional accuracy: 0.0; syntactic validity: 0.33, MaxFrag accuracy: 0.0, MaxFrag similarity: 0.89 , MaxFrag BLEU: 0.79

Figure 8: Example top-3 predictions made by our model and their corresponding evaluation metrics indicated in the figure captions

6.3. Performance across reaction classes

In order to further understand the performance of our model across reaction classes, we increase the granularity of the analysis and compute the five metrics– accuracy, fractional accuracy, MaxFrag accuracy, similarity score, and syntactic validity across the 10 reaction classes. The detailed measures of these metrics are summarized in Tables 13, 14, 15, and 16 for the known reaction class scenario, and in Tables 17, 18, 19, and 20 for the unknown reaction class scenario in Appendix 2. The fraction of invalid predictions across the various reaction types for top-10 analysis are presented in Figures 9 and 10.

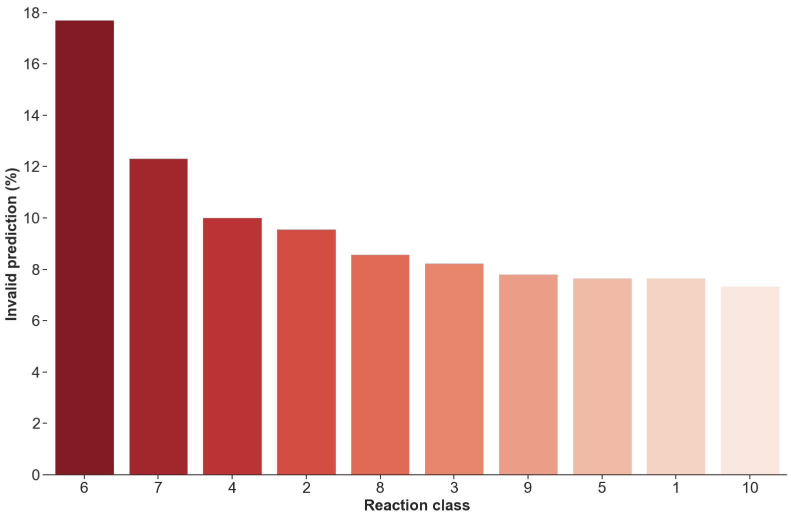


Figure 9: Invalid percentages for Top-10 predictions with known reaction class

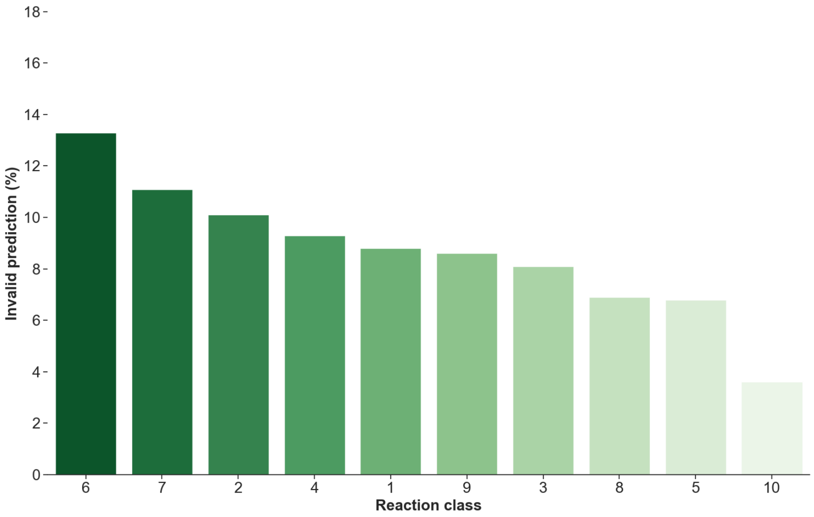


Figure 10: Invalid percentages for Top-10 predictions without reaction class

The above trend indicates that except for reaction class 6 (deprotections) and the surprisingly accurate predictions on reaction class 10 (functional group addition) when the reaction class is unknown, all the reaction types result in nearly the same percentage of invalid predictions. A likely possibility for this observation could be the model learning the underlying grammar, irrespective of the number of samples in each class or the chemical transformations occurring across the different reaction types. This behavior is not trivial since the corresponding top-10 prediction accuracy in Tables 16 and 20 do not follow the same trend across reaction classes. Moreover, the percentage of invalid predictions shows only minor variations across the two scenarios with known and unknown reaction classes. This observation again highlights the ability of our proposed SMILES grammar-based representations to force the model to learn the underlying grammar and consequently generate grammatically correct predictions, irrespective of the other factors. The high percentage error in deprotection reactions could be attributed to several factors that could be specific to the reaction class and could be analyzed through chemistry-driven heuristics that we envision as a hybrid explanation-generation system as a future extension of this work.

6.4. Comparison with other works

Here, we compare the performance of our model against other similar works in this area. One of the first benchmarks in retrosynthesis prediction using seq2seq models on SMILES string representations is by Liu et al. [21]. Their framework is similar to ours in that there are no post-processing of predictions, data augmentation strategies, and model performance-boosting methods used for further improving the model performance – techniques that usually result in improved accuracy custom-fit to a given setting. Our objective is to propose an alternative formulation that is fundamentally different from the other approaches in that it ensures incorporation of chemistry knowledge, forcing the model to learn the underlying SMILES grammar and minimize invalid predictions.

Table 10 compares the prediction accuracy against those reported in Liu et al. We observe that our model improves the prediction accuracy by a margin of $\sim 5\%$ across all the top-N measures and reduces the percentage of invalid predictions by $53\% - 64\%$ when the reaction class is known. We attribute the higher accuracy and the reduced invalid predictions to the grammar-representations that incorporate structural information about the molecules and are characterized by much higher redundancies when compared to SMILES strings as demonstrated using the information-theoretic analysis in Section 4. Figure 11 demonstrates our model's ability to outperform the top-10 accuracy reported in Liu et al. across reaction classes, often by a significant margin. For completeness, we also compare out models' performance with that reported in Liu et al. when the reaction class is unknown. Since they did not evaluate the model under this setting, we use the implementation of [22] that evaluated the performance of this model with unknown reaction class. This is of interest for retrosynthetic planning under certain scenarios where no chemistry information about the target molecule is known apriori. A comparison of the accuracy reported under this scenario is presented in Table 11. We report the detailed class-wise results for both the models (with and without reaction class information) in Appendix 2.

Table 10: Comparison with other similar works involving purely seq2seq models and USPTO 50K dataset with known reaction classes

Model	Top-N measure (with reaction class)			
	accuracy (%) | invalid (%)			
	1	3	5	10
Liu Seq2Seq [21]	37.4 | 12.2	52.4 | 15.3	57.0 | 18.4	61.7 | 22.0
Our work	43.8 | 4.4	57.2 | 7.2	61.4 | 8.4	66.6 | 9.6

Table 11: Comparison with other similar works involving purely seq2seq models and USPTO 50K dataset with unknown reaction classes

Model	Top-N measure (without reaction class)			
	accuracy (%) | invalid (%)			
	1	3	5	10
Liu Seq2Seq [3]	28.3 | -	42.8 | -	47.3 | -	52.8 | -
Our work	32.1 | 5.1	44.3 | 7.4	48.9 | 8.4	54.0 | 9.7

[3] as implemented in [22]; the invalid fractions were not reported for this model

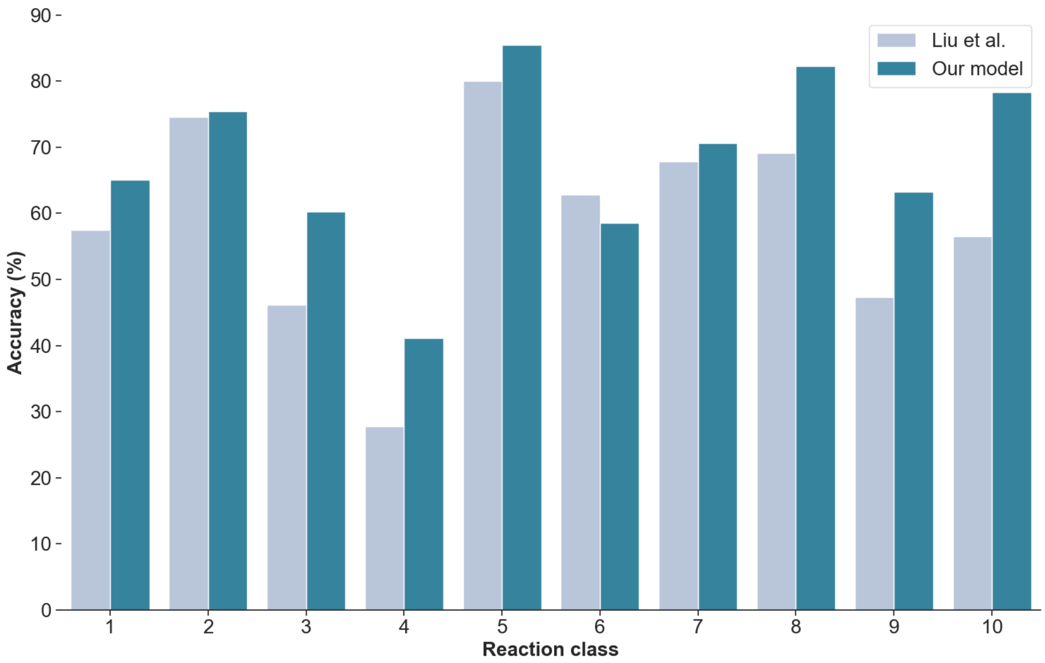


Figure 11: Comparison of top-10 accuracies across different reaction classes

As mentioned earlier, it is possible to achieve even higher prediction accuracy through additional performance boosting techniques as demonstrated in the following studies. Zheng et al. [45] used an additional transformer model that takes as input the output of another transformer model to correct the invalid predictions. Tetko et al [29] proposed data augmentation strategies that significantly increased the size of the dataset used for building a transformer model for retrosynthesis. Karpov et al. [26] used model ensembling, snapshot learning methods, and increasing beam search temperature to improve the model performance. Lin et al. [22] used averaging of model weights and combination with Monte Carlo Tree Search (MCTS) strategies for proposing retrosynthesis routes. The accuracy for such augmented (template-based and template-free) models vary significantly and the top-1 accuracy could be as high as 65%. However, we emphasize here that the objective of this work is not to pursue the state-of-the-art but to highlight the benefits of incorporating prior chemistry knowledge into such black-box models. We have shown that incorporation of this knowledge translates to higher accuracy and fewer invalid predictions when compared to purely black-box models using the same framework. Applying such data augmentation and transfer-learning strategies to boost the model performance further would be the future extension of our work which we conjecture would further improve the model accuracy significantly.

Finally, we would like to highlight here that building models that leverage as much prior chemistry knowledge as possible would be more reliable, acceptable, and explainable as compared to purely data-driven, black-box models that completely disregard known underlying chemistry. Such prior chemistry knowledge could be in the form of information about molecules (grammar, molecular-graph, or structure-based representations), possible reactions (reaction class information, molecular descriptors), model architecture and workflow (that mimic expert chemists), and other similar approaches utilizing deeper integration of first-principles with machine learning-based models.

7. Conclusions

Retrosynthesis analysis is a challenging problem since it involves predicting the precursors with limited information, searching a combinatorially large number of possible synthesis pathways, and approximating an often complex multi-step analysis as a single-step prediction problem. Naturally, incorporating additional information about the reaction or the molecules involved would be of considerable use given the complexity of the task and the limited information often present for making the predictions. Towards that goal, we have proposed grammar-based representations of molecules that incorporate chemical and structural information extracted from their SMILES string representations. We have shown in our earlier work [24] that such representations successfully overcome over-parameterization in models for the forward reaction prediction. Here, we have quantified the superiority of SMILES grammar-based representations compared to the character-based SMILES representations from an information-theoretic standpoint. We have shown that such representations have higher information capacity captured by the Shannon entropy computed for molecules in the USPTO 50K dataset. Moreover, the conditional entropy measures highlighted the higher redundancy built-in to these representations, making them better-suited for machine learning architectures.

The performance of our model reinforced the above observations. We report the top-1 prediction accuracy of 43.8% and syntactic validity of 95.6% as opposed to 37.4% and 87.8%, respectively, reported in Liu et al. We have shown that not only does our model outperform the aggregate statistics reported in Liu et al., the performance of our model across the various reaction classes is much better. An interesting observation is that owing to the grammar representations, our model results in nearly the same percentage of invalid predictions across reaction classes – independent of reaction type, the class-wise number of reactions in the training set, and the known or unknown reaction class scenarios. Moreover, the MaxFrag similarity, which could be as high as 90%, indicates that the model predicts the major precursors required for synthesis fairly accurately. The future extension of our work would involve solving the multi-step retrosynthesis problem and incorporating additional contextual information about the reactions into the same framework.

Acknowledgments

This work was supported in part by the Center for the Management of Systemic Risk (CMSR) at Columbia University.

As the corresponding author, I take this opportunity to express my sincere appreciation and gratitude to Professor George Stephanopoulos for his unwavering support and encouragement throughout my career. As an outsider in the PSE community, lacking the well-known PSE pedigrees of my distinguished colleagues, I recall finding the start of my academic start particularly challenging. Furthermore, I was among the very small minority of researchers exploring the use of artificial intelligence (AI) in chemical engineering in the 1980s, an area that was generally considered a lost cause in those early years. Despite these twin challenges, if I seem to have survived the trials and tribulations, it is due to the invaluable mentoring assistance from Professor George Stephanopoulos. It seems apropos to acknowledge and record my debt to him here in this special issue in his honor.

Appendix 1

The SMILES grammar used in this work is the same as that used in our previous work on the forward prediction problem [24]. This grammar 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 12. The first and the last production rules, SMILES ⟶ CHAIN and NOTHING ⟶ NONE, are additional rules included signifying the start and end of a SMILES string, which is analogous to the <START> and <END> tokens in natural language processing marking the beginning and the end of sentences, respectively.

Table 12: SMILES grammar used in GO-PRO [24]

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 ⟶ CHARGECLASS | 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)

Table 12: SMILES grammar used in GO-PRO [24]

S.No	Production rules
22	`CHAIN` ⟶ `BRANCHED_ATOM` | `CHAIN BRANCHED_ATOM` | `CHAIN BOND BRANCHED_ATOM`
23	`CLASS` ⟶ `DIGIT`
24	`ELEMENT_SYMBOLS` ⟶ `H`
25	`NOTHING` ⟶ `NONE`

Appendix 2

The detailed results capturing the model performance for the five metrics – accuracy, fractional accuracy, syntactic validity, maximal fragment (MaxFrag) accuracy and maximal fragment (MaxFrag) similarity are reported here. Tables 13, 14, 15, and 16 present the results for the top-1, top-3, top-5, and top-10 predictions, respectively, when the reaction class is known. Tables 17, 18, 19, and 20 present the results for the top-1, top-3, top-5, and top-10 predictions, respectively, when the reaction class is unknown.

Scenario 1: Reaction class known

Top-1 measure	Reaction class									
	1	2	3	4	5	6	7	8	9	10
Accuracy	40.9	52.2	37.7	26.7	66.1	35.4	50.4	69.6	38.3	56.5
Fractional accuracy	54.9	67.0	49.9	39.4	81.5	35.4	50.4	75.3	45.0	71.7
Syntactic validity	96.7	95.8	96.4	94.1	97.6	90.9	95.2	97.1	98.8	97.8
MaxFrag accuracy	50.3	61.3	44.9	37.8	83.9	35.4	50.4	77.2	44.4	60.9
MaxFrag similarity	82.0	86.2	78.6	71.2	91.8	68.0	79.5	89.0	78.4	82.9

Table 13: The top-1 accuracy, fractional accuracy, MaxFrag accuracy and MaxFrag similarity scores across the reaction classes (in %)

Top-3 measure	Reaction class									
	1	2	3	4	5	6	7	8	9	10
Accuracy	54.3	66.6	51.4	31.1	80.6	49.9	61.3	77.2	51.7	73.9
Fractional accuracy	66.3	77.4	62.5	49.4	89.5	49.9	61.3	82.9	57.8	80.4
Syntactic validity	94.5	92.6	93.9	91.8	94.4	86.3	89.6	94.4	95.6	91.9
MaxFrag accuracy	61.0	72.1	58.3	46.7	90.3	49.9	61.3	84.8	58.9	73.9
MaxFrag similarity	87.0	91.5	84.3	80.3	98.4	81.0	89.2	96.0	88.0	89.4

Table 14: The top-3 accuracy, fractional accuracy, MaxFrag accuracy and MaxFrag similarity scores across the reaction classes (in %)

Top-5 measure	Reaction class									
	1	2	3	4	5	6	7	8	9	10
Accuracy	59.3	70.9	54.5	36.7	83.9	53.9	64.4	79.7	56.7	73.9
Fractional accuracy	70.7	80.9	66.0	53.9	91.1	53.9	64.4	84.8	61.7	80.4
Syntactic validity	93.4	91.6	92.6	90.9	92.6	84.0	88.9	93.2	94.3	91.6
MaxFrag accuracy	65.2	75.7	61.0	50.0	93.5	53.9	64.4	87.3	62.2	73.9
MaxFrag similarity	88.5	92.2	85.6	81.2	98.5	83.3	90.3	99.2	88.2	89.2

Table 15: The top-5 accuracy, fractional accuracy, MaxFrag accuracy and MaxFrag similarity scores across the reaction classes (in %)

Top-10 measure	Reaction class									
	1	2	3	4	5	6	7	8	9	10
Accuracy	65.1	75.4	60.3	41.1	85.5	58.6	70.6	82.3	63.3	78.3
Fractional accuracy	75.5	84.2	70.5	58.3	91.9	58.6	70.6	86.7	67.8	82.6
Syntactic validity	92.3	90.4	91.8	90.0	92.3	82.3	87.7	91.4	92.2	92.7
MaxFrag accuracy	69.8	79.4	65.7	53.3	93.5	58.6	70.6	88.6	67.8	78.3
MaxFrag similarity	90.1	93.5	87.8	82.9	98.6	85.0	92.0	98.9	90.5	91.8

Table 16: The top-10 accuracy, fractional accuracy, MaxFrag accuracy and MaxFrag similarity scores across the reaction classes (in %)

Scenario 2: Reaction class unknown

Top-1 measure	Reaction class									
	1	2	3	4	5	6	7	8	9	10
Accuracy	32.2	39.9	25.8	10.0	35.5	26.7	35.8	38.0	16.1	60.9
Fractional accuracy	42.6	52.2	33.9	15.0	44.4	26.8	35.8	40.5	22.5	63.0
Syntactic validity	95.4	94.8	96.1	92.4	96.4	92.0	94.4	96.8	96.3	100.0
MaxFrag accuracy	40.7	48.9	33.0	11.1	50.0	26.9	35.8	41.8	21.7	60.9

Top-1 measure	Reaction class									
	1	2	3	4	5	6	7	8	9	10
MaxFrag similarity	76.7	80.6	72.9	57.3	79.2	70.7	74.3	83.0	72.4	81.6

Table 17: The top-1 accuracy, fractional accuracy, MaxFrag accuracy and MaxFrag similarity scores across the reaction classes (in %)

Top-3 measure	Reaction class									
	1	2	3	4	5	6	7	8	9	10
Accuracy	43.5	54.8	37.1	15.6	43.5	39.9	48.0	48.1	23.9	69.6
Fractional accuracy	53.5	65.7	46.0	21.1	53.2	39.9	48.0	51.3	30.3	71.7
Syntactic validity	93.2	92.7	93.8	90.0	94.8	89.4	92.0	95.5	93.3	97.7
MaxFrag accuracy	50.3	60.7	44.2	17.8	58.1	39.9	48.0	51.9	28.9	69.6
MaxFrag similarity	82.5	86.7	78.3	65.1	85.1	79.2	83.1	87.2	78.2	89.0

Table 18: The top-3 accuracy, fractional accuracy, MaxFrag accuracy and MaxFrag similarity scores across the reaction classes (in %)

Top-5 measure	Reaction class									
	1	2	3	4	5	6	7	8	9	10
Accuracy	48.5	59.8	41.7	17.8	46.8	44.2	51.5	48.1	30.0	69.6
Fractional accuracy	58.7	70.6	51.5	27.8	58.9	44.2	51.5	51.9	35.3	71.7
Syntactic validity	92.3	91.4	93.3	90.5	94.0	88.0	90.6	94.6	92.8	96.4
MaxFrag accuracy	54.6	64.7	48.8	24.4	64.5	44.2	51.5	51.9	32.8	69.6
MaxFrag similarity	83.8	88.2	80.1	68.0	89.1	81.5	84.8	87.8	78.8	88.8

Table 19: The top-5 accuracy, fractional accuracy, MaxFrag accuracy and MaxFrag similarity scores across the reaction classes (in %)

Top-10 measure	Reaction class									
	1	2	3	4	5	6	7	8	9	10
Accuracy	53.2	64.4	49.0	20.0	58.1	50.1	55.5	57.0	34.4	69.6
Fractional accuracy	64.8	74.4	59.1	31.7	71.8	50.1	55.5	59.5	40.6	73.9
Syntactic validity	91.2	89.9	91.9	90.7	93.2	86.7	88.9	93.1	91.4	96.4
MaxFrag accuracy	59.8	68.6	55.8	30.0	79.0	50.1	55.5	60.8	37.8	69.6
MaxFrag similarity	85.4	89.4	82.3	71.9	92.9	83.9	87.3	88.7	80.1	88.2

Table 20: The top-10 accuracy, fractional accuracy, MaxFrag accuracy and MaxFrag similarity scores across the reaction classes (in %)

References

1. Pensak, D. A. & Corey, E. J. in (ACS Publications).

2. Salatin, T. D. & Jorgensen, W. L. Computer-assisted mechanistic evaluation of organic reactions. 1. Overview. *The Journal of Organic Chemistry* **45,** 2043–2051 (1980).

3. Corey, E., Long, A. K., Greene, T. W. & Miller, J. W. Computer-assisted synthetic analysis. Selection of protective groups for multistep organic syntheses. *The Journal of Organic Chemistry* **50,** 1920–1927 (1985).

4. Jorgensen, W. L. *et al.* CAMEO: a program for the logical prediction of the products of organic reactions. *Pure and Applied Chemistry* **62,** 1921–1932 (1990).

5. Satoh, H. & Funatsu, K. SOPHIA, a knowledge base-guided reaction prediction system-utilization of a knowledge base derived from a reaction database. *Journal of chemical information and computer sciences* **35,** 34–44 (1995).

6. Satoh, K. & Funatsu, K. A novel approach to retrosynthetic analysis using knowledge bases derived from reaction databases. *Journal of chemical information and computer sciences* **39,** 316–325 (1999).

7. Chen, J. H. & Baldi, P. No electron left behind: a rule-based expert system to predict chemical reactions and reaction mechanisms. *Journal of chemical information and modeling* **49,** 2034–2043 (2009).

8. Law, J. *et al.* Route designer: a retrosynthetic analysis tool utilizing automated retrosynthetic rule generation. *Journal of chemical information and modeling* **49,** 593–602 (2009).

9. Gothard, C. M. *et al.* Rewiring chemistry: Algorithmic discovery and experimental validation of one-pot reactions in the network of organic chemistry. *Angewandte Chemie International Edition* **51,** 7922–7927 (2012).

10. Szymkuć, S. *et al.* Computer-Assisted Synthetic Planning: The End of the Beginning. *Angewandte Chemie International Edition* **55,** 5904–5937 (2016).

11. Segler, M. H. & Waller, M. P. Modelling chemical reasoning to predict and invent reactions. *Chemistry–A European Journal* **23,** 6118–6128 (2017).

12. Mann, V., Sivaram, A., Das, L. & Venkatasubramanian, V. Robust and efficient swarm communication topologies for hostile environments. *Swarm and Evolutionary Computation* **62,** 100848 (2021).

13. Biłozor, A., Kowalczyk, A. M. & Bajerowski, T. Theory of scale-free networks as a new tool in researching the structure and optimization of spatial planning. *Journal of Urban Planning and Development* **144,** 04018005 (2018).

14. Zhang, Y., Li, Y., Zhou, Y. & Ma, J. Optimal link rewiring strategy for transport efficiency on scale-free networks with limited bandwidth. *International Journal of Modern Physics C* **31,** 2050033 (2020).

15. Xu, S., Xia, Y. & Ouyang, M. Effect of resource allocation to the recovery of scale-free networks during cascading failures. *Physica A: Statistical Mechanics and its Applications* **540,** 123157 (2020).

16. Venkatasubramanian, V. The promise of artificial intelligence in chemical engineering: Is it here, finally? *AIChE Journal* **65,** 466–478 (2019).

17. Segler, M. H. & Waller, M. P. Neural-symbolic machine learning for retrosynthesis and reaction prediction. *Chemistry–A European Journal* **23,** 5966–5971 (2017).

18. Wei, J. N., Duvenaud, D. & Aspuru-Guzik, A. Neural networks for the prediction of organic chemistry reactions. *ACS central science* **2,** 725–732 (2016).

19. Coley, C. W., Barzilay, R., Jaakkola, T. S., Green, W. H. & Jensen, K. F. Prediction of organic reaction outcomes using machine learning. *ACS central science* **3,** 434–443 (2017).

20. Nam, J. & Kim, J. Linking the neural machine translation and the prediction of organic chemistry reactions. *arXiv preprint arXiv:1612.09529* (2016).

21. Liu, B. *et al.* Retrosynthetic Reaction Prediction Using Neural Sequence-to-Sequence Models. *ACS Central Science* **3.** PMID: 29104927, 1103–1113 (2017).

22. Lin, K., Xu, Y., Pei, J. & Lai, L. Automatic retrosynthetic route planning using template-free models. *Chemical Science* **11,** 3355–3364 (2020).

23. Zheng, S., Rao, J., Zhang, Z., Xu, J. & Yang, Y. Predicting retrosynthetic reactions using self-corrected transformer neural networks. *Journal of Chemical Information and Modeling* **60,** 47–55 (2019).

24. Mann, V. & Venkatasubramanian, V. Predicting chemical reaction outcomes: A grammar ontology-based transformer framework. *AIChE Journal* **67,** e17190 (2021).

25. Mann, V. & Venkatasubramanian, V. *A Formal Grammar-Based Machine Learning Approach for Predicting Reaction Outcomes* in *2020 Virtual AIChE Annual Meeting* (2020).

26. Karpov, P., Godin, G. & Tetko, I. V. *A transformer model for retrosynthesis* in *International Conference on Artificial Neural Networks* (2019), 817–830.

27. Duan, H., Wang, L., Zhang, C., Guo, L. & Li, J. Retrosynthesis with attention-based NMT model and chemical analysis of “wrong” predictions. *RSC Advances* **10,** 1371–1378 (2020).

28. Schwaller, P. *et al.* Predicting retrosynthetic pathways using transformer-based models and a hyper-graph exploration strategy. *Chemical Science* **11,** 3316–3325 (2020).

29. Tetko, I. V., Karpov, P., Van Deursen, R. & Godin, G. State-of-the-art augmented NLP transformer models for direct and single-step retrosynthesis. *Nature communications* **11,** 1–11 (2020).

30. Ucak, U. V., Kang, T., Ko, J. & Lee, J. Substructure-based neural machine translation for retrosynthetic prediction. *Journal of Cheminformatics* **13,** 4. ISSN: 1758-2946 (2021).

31. Vaswani, A. *et al.* in *Advances in Neural Information Processing Systems 30* (eds Guyon, I. *et al.*) 5998–6008 (Curran Associates, Inc., 2017).

32. Chomsky, N. Three models for the description of language. *IRE Transactions on Information Theory* **2,** 113–124. ISSN: 2168-2712 (Sept. 1956).

33. Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences* **28,** 31–36 (1988).

34. Shannon, C. E. A Mathematical Theory of Communication. *Bell System Technical Journal* **27,** 379–423 (1948).

35. Bonchev, D. & Trinajstić, N. Chemical information theory: Structural aspects. *International Journal of Quantum Chemistry* **22,** 463–480 (1982).

36. Chandler, J. An Introduction to the Foundations of Chemical Information Theory. Tarski–Lesniewski Logical Structures and the Organization of Natural Sorts and Kinds. *Information* **8,** 15. ISSN: 2078-2489 (Jan. 2017).

37. Graham, D. J. Information and Organic Molecules: Structure Considerations via Integer Statistics. *Journal of Chemical Information and Computer Sciences* **42.** PMID: 11911689, 215–221 (2002).

38. Nalewajski, R. F. & Parr, R. G. Information Theory Thermodynamics of Molecules and Their Hirshfeld Fragments (2001).

39. Nalewajski, R. F. & Parr, R. G. Information theory, atoms in molecules, and molecular similarity. *Proceedings of the National Academy of Sciences* **97,** 8879–8882. ISSN: 0027-8424 (2000).

40. Lowe, D. M. *Extraction of chemical structures and reactions from the literature* PhD thesis (University of Cambridge, 2012).

41. Schneider, N., Stiefl, N. & Landrum, G. A. What's what: The (nearly) definitive guide to reaction role assignment. *Journal of chemical information and modeling* **56,** 2336–2346 (2016).

42. Kingma, D. P. & Ba, J. *Adam: A Method for Stochastic Optimization* 2014.

43. Kusner, M. J., Paige, B. & Hernández-Lobato, J. M. Grammar Variational Autoencoder (2017).

44. Papineni, K., Roukos, S., Ward, T. & Zhu, W.-J. *BLEU: a method for automatic evaluation of machine translation* in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics* (2002), 311–318.

45. Zheng, S., Rao, J., Zhang, Z., Xu, J. & Yang, Y. Predicting Retrosynthetic Reactions Using Self-Corrected Transformer Neural Networks. *Journal of Chemical Information and Modeling* **60.** PMID: 31825611, 47–55 (2020).