Language Models for Predicting Organic Synthesis Procedures

Mantas Vaškevičius * and Jurgita Kapočiūtė-Dzikienė

Posted Date: 25 November 2024

doi: 10.20944/preprints202411.1807.v1

Keywords: deep learning; large language model; organic synthesis; synthesis procedure; machine learning; artificial intelligence

Article

Language Models for Predicting Organic Synthesis Procedures

Mantas Vaškevičius * and Jurgita Kapočiūtė-Dzikienė

Department of Applied Informatics, Vytautas Magnus University, LT-44404 Kaunas, Lithuania
* Correspondence: mantas.vaskevicius@vdu.lt

Featured Application: Leveraging fine-tuned large language models, this study enhances the prediction of efficient chemical synthesis procedures, saving time and resources in laboratory workflows. The findings highlight the potential of tailored AI approaches in transforming organic synthesis through intelligent, data-driven methods.

Abstract: In optimizing organic chemical synthesis, researchers often face challenges in efficiently generating viable synthesis procedures that conserve time and resources in laboratory settings. This paper systematically analyzes multiple approaches to efficiently generate synthesis procedures for a wide variety of organic synthesis reactions, aiming to decrease time and resource consumption in laboratory work. We investigated the suitability of different sizes of BART, T5, FLAN-T5, molT5, and classic sequence-to-sequence transformer models for our text-to-text task and utilized a large dataset prepared specifically for the task. Experimental investigations demonstrated that a fine-tuned molT5-large model achieves a BLEU score of 47.75. The results demonstrate the capability of LLMs to predict chemical synthesis procedures involving 24 possible distinct actions, many of which include various parameters like solvents, reaction agents, temperature, duration, solvent ratios, and other specific parameters. Our findings show that only when the core reactants are used as input, the models learn to correctly predict what ancillary components need to be included in the resulting procedure. These results are valuable for AI researchers and chemists, suggesting that curated datasets and large language model fine-tuning techniques can be tailored for specific reaction classes and practical applications. This research contributes to the field by demonstrating how deep-learning-based methods can be customized to meet the specific requirements of chemical synthesis, leading to more intelligent and resource-efficient laboratory processes.

Keywords: deep learning; large language model; organic synthesis; synthesis procedure; machine learning; artificial intelligence

1. Introduction

Organic chemistry is at the core of multiple industries and plays a pivotal role in the contemporary world. This field involves synthesizing chemicals by carefully choosing and mixing reactants and auxiliary components, such as catalysts, while also considering environmental and temporal factors like reaction temperature and length. Following synthesis, the compound is purified using methods like distillation, crystallization, chromatography, or extraction to ensure purity, with the method selected based on the compound's unique properties. Concise instructions or procedures, commonly referred to by chemists, outline the choice of solvents, precise temperature conditions, and reaction duration, which are critical in achieving the desired product with high yield and purity. The task of delineating the optimal procedure for a given combination of reactants typically necessitates a series of iterative experiments involving various chemical reagents and conditions. Chemists utilize scientific databases such as SciFinder and Reaxys [1] and peer-reviewed literature to access detailed information on previous experiments and methods for creating known compounds. However,

 doi:10.20944/preprints202411.1807.v1

creating new compounds or using new combinations of reactants poses significant challenges because there are no established protocols.

The emerging field of deep learning (DL) offers a promising solution by potentially predicting synthesis procedures in cases where traditional resources are unavailable. Recent advancements in DL have highlighted its efficacy in modeling chemical properties and reaction dynamics [2-6]. Interestingly, large language models have revolutionized various fields of artificial intelligence, demonstrating capabilities in natural language processing, text generation, and even specialized domains like chemistry. The rapid progress in LLM development has led to increasingly sophisticated models capable of understanding and generating complex, domain-specific content with high accuracy [7]. In the field of organic chemistry, LLMs present a promising path for predicting synthesis procedures, potentially simplifying the process of developing new compounds and optimizing existing ones [8]. By framing chemistry as a text-to-text problem [9], we leverage the power of language models to interpret and generate chemical language, offering a novel approach to challenges in the field. Viewing chemistry as a text-to-text problem involves representing chemical reactions and processes as sequences of text, allowing them to be processed by language models in a manner similar to natural language. This innovative approach enables the application of natural language processing techniques to chemical problems, to solve tasks such as reaction prediction, retrosynthesis planning, and procedure generation by treating chemical formulas, reactions, and methodologies as a form of specialized language that can be learned and generated by appropriately trained models. Large language models (LLMs) trained on diverse datasets can help in understanding and generating chemical language with precision.

Organic synthesis encompasses a wide range of reaction types, including hydrogenation [10], oxidation [11], reduction [12], condensation [13], nucleophilic substitution [14], electrophilic addition [15] and redox reactions [16]. More specialized processes like the Suzuki coupling [17], Diels-Alder reaction [18], Aldol condensation [19], Wittig reaction [20] and Grignard reaction [21] enable chemists to synthesize increasingly complex molecules with precision and diversity. Across a variety of reactions, factors such as solvent choice, reaction temperature, and the use of catalysts influence yields, selectivity, and rates [22]. With the increasing complexity of molecules, especially with multiple functional groups, accurate and efficient synthetic methodologies are vital. However, identifying a synthetic pathway is insufficient, because a comprehensive experimental procedure is needed for each synthesis. The development of a chemical synthesis procedure involves detailed knowledge of the specific sequence of actions required, such as adding chemicals and performing the synthesis along with accurately defining their optimal conditions, like temperature, solvent choices, and atmospheric requirements. In classical chemistry, empirical knowledge and iterative testing have guided the determination of optimal synthesis conditions and steps [23]. The challenge is devising procedures for novel or minimally investigated reactions, due to a dependence on pre-existing databases. Recent advancements in DL for molecule generation also present new challenges and opportunities for chemists. These methods facilitate the discovery of compounds within previously underexplored chemical domains. Chemists need to create specific synthesis plans and procedures for practical laboratory applications based on these discoveries. The transition to machine learning (ML) provides a more effective method for analyzing chemical data, improving prediction for unexplored reactions. The task does not have many established conventions compared to challenges like language translation or text classification. The lack of established standards for creating synthesis methods requires the creation of flexible and adaptable methodologies. While synthesis procedure generation is not extensively studied, a transformer-based sequence-to-sequence model has been shown to attain a BLEU score of 54.7 through text-based representations by IBM [8]. Our methodology is a variant of the published methods and is closer to the research demonstrating the effectiveness of the FLAN-T5 model for the generation of esterification synthesis reactions, which attains a BLEU score of 51.82 [24]. Our paper aims to expand on the available knowledge and explore LLM applicability for the prediction of synthesis instructions for a large variety of organic synthesis types. Specifically, we assess the performance of LLMs by fine-tuning them with a curated dataset to predict a precise sequence of actions and parameters. Such a strategy offers an initial predictive

 doi:10.20944/preprints202411.1807.v1

means to scientists to optimize reaction conditions in advance and simplify the experimental process. This method seeks to enhance the efficiency of chemical synthesis, potentially reducing both the time and resources used in laboratory work. To enhance the state-of-the-art in generating synthesis procedures, our research presents several pivotal contributions:

- **Creation of the novel dataset.** We compiled a unique dataset encompassing multiple reaction classes (~931,350 reactions in total). This dataset forms the foundation for the exploration of synthesis procedure prediction.
- **Selective input strategy for enhanced predictive modeling.** A key feature of our methodology is the intentional omission of ancillary compounds from the input to our models; we provide only the reactants and products, purposely leaving out any agents not specific to the reaction, such as gases, solvents, and catalysts. This approach adds complexity to the model's task, as it cannot rely solely on the given inputs to predict all necessary steps and parameters, including those related to ancillary compounds. However, this limitation results in outputs that are particularly useful for researchers, especially in the context of novel compounds for which detailed synthetic documentation is absent. In such cases, usually, the reactants and products are often the only known factors before laboratory experiments.
- **LLMs suitability exploration and comparison.** We investigated the suitability of different sizes of BART, T5, FLAN-T5, molT5, and classic sequence-to-sequence transformer models suitable for our text-to-text task. In the experiments, we first tested pre-trained models and then used configurations with randomly initialized models, allowing us to assess the impact of pre-trained weights. Additionally, we assessed the model's performance using the default tokenizer versus a fine-tuned version on our dataset, aiming to discern the impact of specialized tokenization on the overall effectiveness of our text-to-text tasks. The paper presents results and discusses the predictive capabilities of the optimal models, which were trained on the largest currently available dataset specifically prepared for the task.
- **Open source.** We also open source the dataset and model for the academic community, providing a valuable resource that encourages further research and development in the field of chemical synthesis procedure generation.
- **Standardization of synthesis procedure parameters.** The dataset features a high level of standardization of reaction agent and solvent names, temperature, and duration expressions. The models can approximate the ancillary compounds without relying on specific input and the generated outputs are more accurate.
- **Extended parameter prediction.** Models are capable of predicting more parameters of actions (specifically the *purify* step) than what is available at the time of publishing.

2. Related Work

Deep learning methods have notably enhanced a variety of tasks within the chemistry domain [25]. The development of transformer architecture has been particularly influential across several areas of chemical research and applications.

2.1. Molecular Generation

In the field of drug discovery, models like MolBERT [26] and ChemGPT [27] have effectively utilized NLP techniques to facilitate analysis and insights in chemical research. Trained with databases such as ZINC [28], BARTSmiles [29] and Chemformer [30] have demonstrated the capacity of DL to enhance molecular generation tasks. These models use large-scale chemical databases to generate novel molecular structures, potentially accelerating the drug discovery process. Recent advancements in molecular generation have significantly expanded the capabilities of computational chemistry. GraphINVENT [31] uses a graph-based approach for generating molecules with desired properties, while HierG2G [32] uses graph-to-graph translation for targeted molecular design. The integration of reinforcement learning in models like MolGPT [33] allows for multi-objective optimization in molecule generation. Foundation models like MolFormer [34] demonstrate the potential of transfer learning in tackling diverse molecular design challenges efficiently across various chemical spaces and applications.

 doi:10.20944/preprints202411.1807.v1

2.2. Reaction and Property Prediction

T5Chem [35] has been trained on large-scale datasets to refine the predictive accuracy for complex chemical phenomena. The nach0 model, an encoder-decoder LLM pre-trained on scientific texts and molecular strings, has shown remarkable versatility in chemical and biological tasks, capable of producing both molecular and textual outputs [36]. These models excel in predicting chemical reactions and properties, offering valuable insights into synthetic chemistry and materials science. Large language models (LLMs) have been increasingly applied to various scientific domains beyond traditional text-based tasks. The utility of LLMs has been demonstrated in material science by creating a structured dataset for solar cell materials and using these models to predict their performance [37]. AttentiveFP [38] improved molecular property prediction using graph neural networks, while RetroPrime [39] enhanced retrosynthetic prediction with a transformer approach. Multi-task learning models like ChemBERTa-2 [40] have demonstrated the potential of transfer learning across diverse chemical prediction tasks, collectively advancing our understanding and predictive capabilities in chemical research and applications.

2.3. Synthesis Planning and Autonomous Experimentation

Graph transformer models like RetroExplainer [41] employ a DL-guided approach for molecular assembly, offering an efficient and highly scalable solution to design synthesis pathways. The development of G2Gs (Graph-to-Graph models) has further revolutionized retrosynthesis planning. For instance, RetroGNN [42] utilizes graph neural network architecture to predict retrosynthetic steps with high accuracy, considering both local and global molecular features. Additionally, the MEGAN model [43] has shown promise in predicting multi-step reaction outcomes, capturing intricate reaction pathways and helping chemists anticipate unexpected side products. ChemLLM is also a specialized large language model designed specifically for chemical tasks, including reaction prediction and synthesis planning [44]. A significant leap in LLM applications is the development of multi-LLM agents capable of independently designing, planning, and conducting scientific experiments [45]. Large language models can also be leveraged for predictive chemistry, highlighting the potential of these models to predict chemical reactions and syntheses with increasing accuracy [46]. This advancement highlights the potential of LLMs in driving autonomous discoveries in chemistry, potentially revolutionizing how research is conducted. Building on this foundation, ChemCrow [47] demonstrates how LLMs can be integrated with specialized chemical tools to perform complex reasoning tasks and design experiments.

2.4. Synthesis Planning and Autonomous Experimentation

The task of synthesis procedure generation does not have many established conventions compared to challenges like language translation, text classification or even retrosynthesis route planning. The lack of established standards for creating synthesis methods requires the creation of flexible and adaptable methodologies. While synthesis procedure generation is not extensively studied, a transformer-based sequence-to-sequence model has been shown to attain a BLEU score of 54.7 through text-based representations by IBM [8].

3. Formal Definition of the Tasks

The paper addresses a generative text-to-text task where the aim is to transform a given source chemical reaction into a structure reaction procedure. We denote $r = (r_1, r_2, \ldots, r_n)$, where each r_i is a *reactant* | *reactant* >> *product* in SMILES notation and $p = (p_1, p_2, \ldots, p_m)$, where p_i is a structured reaction procedure in English language. This target description is in a structured, machine-readable format that outlines the specific steps, conditions, and parameters relevant to the reaction at hand. We define R as the domain of all potential chemical reactions in SMILES notation, and P as the domain encompassing all target procedure descriptions in a structured, machine-readable format. We introduce Θ, an ML algorithm that must learn a mapping function, $\phi(R) \rightarrow P$, which translates a source reaction description into its corresponding target procedure description.

 doi:10.20944/preprints202411.1807.v1

The objective for Θ is to learn the approximation (noted ϕ) of this mapping function using a training dataset $D_R \subset R$. This dataset comprises pairs where each source reaction description, r, is paired with its target procedure description, p, in the structured format. The effectiveness of the learned function ϕ is then tested against a separate testing dataset, $D_T \subset R$, containing reaction descriptions unseen during training. The mapping function ϕ is implemented using a transformer-based sequence-to-sequence model, chosen for its proven effectiveness in handling complex text generation tasks. This architecture allows for efficient processing of the input SMILES notation (sequence of characters) and generation of structured output procedures (text). The model is trained using a cross-entropy loss function, which is well-suited for sequence generation tasks. The model's efficacy is measured by the accuracy of its predictions compared to the target procedures, utilizing a predefined objective evaluation metric. For evaluation, we employ multiple metrics including BLEU score to assess the fluency and accuracy of the generated procedures, and a custom metric that measures the structural consistency of the output with respect to the expected format.

4. The Data

In the preparation of our dataset, we utilized a compilation of synthesis procedures extracted from USPTO and EPO patents spanning from 1971 to 2022. Patents from the USPTO and EPO are detailed documentation that covers a variety of chemical processes and synthesis methods. The organized format of patents, with comprehensive descriptions of experimental methods, compounds, outcomes, and parameters, offers a uniform structure for extracting and converting data. Patents provide a simpler way to access chemical techniques, but the process of obtaining information is intricate. That is why we utilize an already processed dataset, named EPO/USPTO, which was created using a new methodology outlined in a study that utilizes a mix of ML algorithms, software tools, and specifically prepared scripts [48]. The processing consists of two main tasks: first, categorizing patent paragraphs to precisely recognize chemical procedures, and second, transforming these procedures into a structured and machine-readable format. Three examples are presented in Table 1 (original and structured format examples).

Table 1. Three examples of original and structured chemical procedure sentences and three examples of input and output from the PRP-931k dataset.

Original format	Structured format
In 1.5 ml of N,N-dimethylformamide was dissolved 150 mg of ethyl 2-[6-chloro-2-(2,4-difluorophenylamino)-5-fluoronicotinoyl]acetate, and 70 mg of thiophenol and 60 mg of triethylamine were added thereto, after which the resulting mixture was subjected to reaction at room temperature for 1 hour.	ADD N,N-dimethylformamide (1.5 ml); ADD ethyl 2-[6-chloro-2-(2,4-difluorophenylamino)-5-fluoronicotinoyl]acetate (150 mg); ADD thiophenol (70 mg); ADD triethylamine (60 mg); STIR for 1 hour at room temperature.
The organic layer was separated, washed successively with 10 ml of water and 10 ml of saturated aqueous sodium chloride solution, and then dried over anhydrous magnesium sulfate.	PHASESEPARATION; COLLECTLAYER organic; WASH with water (10 ml); WASH with saturated aqueous sodium chloride solution (10 ml); DRYSOLUTION over anhydrous magnesium sulfate.
Inputs from PRP-931k dataset	**Outputs from PRP-931k dataset**
CC(C)CCON=O|O=C1CCc2ccc(I)cc21>>O=C1/C(=N/O)Cc2ccc(I)cc21	MAKESOLUTION with $R2$ and DCM and CH3OH; ADD SLN; SETTEMPERATURE 0° C; MAKESOLUTION with $R1$ and DCM; ADD SLN over 30 min; STIR for 3 h at 25° C; CONCENTRATE; ADD Diethyl ether; FILTER keep precipitate; YIELD $P1$.

 doi:10.20944/preprints202411.1807.v1

CCCC[Sn](CCCC)(CCCC)c1ccncc1| Cc1ccc(S(=O)(=O)n2cc(I)c3c (NC4CC4)nc(Cl)nc32)cc1>> Cc1ccc(S(=O)(=O)n2cc(- c3ccncc3)c3c(NC4CC4)nc(Cl)nc32)cc1	MAKESOLUTION with $R2$ and Dioxane; ADD SLN; ADD $R1$; ADD Pd(PPh3)4; STIR for 18 h at 100° C; ADD water; ADD Ethyl acetate; PHASESEPARATION; COLLECTLAYER organic; DRYSOLUTION over Na2SO4; CONCENTRATE; PURIFY; YIELD $P1$.

The steps are essential, but the dataset mentioned earlier is not directly suitable for our task and needs additional processing. The data section outlines the steps required to prepare the dataset used in our study, referred to as *PRP-931k*. The abbreviation PRP stands for Patent Reaction Procedures, indicating that the extracted instances include both reactions and procedures. The *931k* represents the dataset's total size, intended for discerning potential larger versions in the future. The final dataset is unique compared to the commonly utilized USPTO-50k [49] or USPTO-MIT [50] datasets as it contains reactants and products in SMILES format, along with simplified and machine-readable synthesis procedures.

The refined dataset contains input and output instances. The inputs are represented as single lines of text, denoting reactants and products in SMILES notation, whereas the outputs describe a series of synthesis procedure steps and their respective parameters. Three examples from the dataset are presented in Table 1. This notation system simplifies the representation of molecular structures by using a linear text string, making it easy to encode and decode intricate molecular geometries, functional groups, and connectivity. It enables a clear representation of molecular structures, reducing the chance of misunderstanding or mistakes in the synthesis planning process. Additionally, when SMILES notation is utilized as input for predictive models, it allows for the direct implementation of data-driven methods in chemical synthesis without requiring complex translations or conversions of molecular representations.

The output procedure is composed of actions that are constrained and presented in a methodical and simple manner: a single word denotes the action, followed by its parameters. The actions (24 in total) are: *Add, Stir, Concentrate, Yield, MakeSolution, Wash, CollectLayer, Purify, Filter, DrySolution, Extract, SetTemperature, Reflux, PH, PhaseSeparation, DrySolid, Quench, Wait, Recrystallize, Partition, Degas, Triturate, Microwave, Sonicate*. The schema used for action, parameter naming, and formatting was first introduced by D. M. Lowe [51] and later improved by IBM. The input SMILES notation requires case sensitivity, with aromatic atoms represented in lowercase and aliphatic atoms in uppercase. The output is also case-sensitive to enable differentiation between action names, compound names, abbreviations, and parameters such as temperature and duration. Computer code can read procedures for analysis and potentially utilize it in various robotic synthesizers. Consequently, the syntax is strict, and even minor spelling mistakes in the words render the entire procedure incorrect.

4.1. Dataset Preparation

The *PRP-931k* dataset has been prepared by utilizing a larger EPO/USPTO dataset [48] of chemical reactions and their related procedures. The dataset undergoes several stages of refinement to make it appropriate for our task.

For input, we combine reactants and products from the EPO/USPTO dataset, which provides reactants in SMILES format. We created a curated list of 252 undesirable compounds, including inorganic substances, organic solvents, acids, bases, and complexes, to filter out ancillary substances from the input. This list, available in our project repository, enhances the dataset beyond the initial removals in the EPO/USPTO reactant column. Importantly, these filtered compounds are not eliminated from the reaction but are included in the output procedure text. This approach simulates real-world scenarios where scientists lack complete knowledge of required solvents and agents when synthesizing novel chemicals. By creating this learning environment, we aim to mirror the challenges researchers face, making the work more difficult but ultimately more beneficial by offering additional valuable information. We used RDKit to standardize SMILES notation for all compounds. Reactants

 doi:10.20944/preprints202411.1807.v1

and products are concatenated into a single line, using "|" to separate compounds and ">>" between reactants and products. While similar tasks often use a dot (".") for separation, we opted for the bar symbol to avoid confusion, as dots can represent disconnected compounds in SMILES notation ("[Li+].[OH-]" for lithium hydroxide). We removed duplicate input instances, including variations where reactant order was swapped. Instances with less than 5 total atoms were discarded and validated as incorrectly mapped reactions. We also identified cases with missing reactants by calculating the average atom count ratio between reactants and products. Ratios of 2.5 or higher typically indicate missing compounds. An expert-determined threshold of 2.3 was used to remove 20,000 instances, resulting in a cleaner dataset despite the removal of some correct instances. This approach is effective only when ancillary substances are absent from the reaction notation.

In the first stage of output processing, we performed common checks and initial cleaning. Quantities of added compounds were removed as they are typically calculated separately. Valid procedures must meet specific requirements: they cannot contain actions like *FollowOtherProcedure*, *OtherLanguage*, or *InvalidAction*. Some actions (e.g., Concentrate, Filter) may appear without parameters, while others require them. The action names were designed during procedure formalization to accurately represent the original natural language procedure. We aimed to maintain the original framework without altering procedure actions, either removing instances entirely or leaving actions unchanged. Our approach prioritizes retaining the formalized output format, removing instances with mistakes rather than modifying procedures. This decision is supported by limited research in the field and allows for future tracking of conceptual changes in action name inference, definitions, and syntax.

Reactants and products need to be tokenized. To avoid the need for complex IUPAC or SMILES notation in the output, which would require accurate generation each time, we denote reactants as *$R1$*, *$R2$*, …, *RN*, and products as *$P1$*, *$P2$*, …, *PN*. Only compounds related to the input are tokenized. The mapping between SMILES and IUPAC naming is accomplished by the same process which allows the dataset to have two formats in the first place. A vocabulary of IUPAC to SMILES names is available alongside the EPO/USPTO dataset mentioned earlier. The tokenization does not necessitate copying the reaction compounds to the resulting procedure and, most importantly, avoids potential errors in the notation.

There are generally four types of parameters: compounds, solvents, duration, and temperature. Standardization is a substantial challenge for this task, but it is essential to address all stages of the process. The supervised machine-learning methods benefit from unified naming conventions because they reduce the complexity of synthesis procedure generation and improve the quality of the output. Regarding the chemistry, the task remains unchanged. Standardization benefits the user by ensuring consistent output, eliminating the need for additional interpretation steps when analyzing variations of the same compound in automated processes. Actions were converted into a formalized format and a significant portion of unnecessary text from the original procedure was removed, but the specific notation of parameters was not modified. In the case of temperature, for instance, the same temperature point can be found in multiple formats: “15 degrees”, “15 °C”, “15 C”, “15 °c”, “15 °”, “15 Celsius”, “10-20 °C”, etc. This is true for all other parameters; the specific notation or format is simply copied over when converting from the original procedure text. While the methodology to standardize temperature and duration can be relatively straightforward, ancillary compounds and solvents pose a challenge due to the high number of possible names and their variants. To address these standardization problems, we have employed heuristic algorithms and manual inspection to create vocabularies for all four types of parameter types. The processes are briefly described in Table 2.

Table 2. Description of standardization processes for different actions.

Action names	
Add, MakeSolution	Compound name variants were standardized.

 doi:10.20944/preprints202411.1807.v1

Degas, DrySolution, Extract, Partition, Purify, PH, Quench, Recrystallize, Triturate, and Wash	Names of solvents have been converted to consistent formats.
Heat, Add, SetTemperate	Temperature notations were converted to Celsius.
Heat, Wait, Reflux	Duration notations were converted to a machine-readable format. we have used "min" for minutes, "h" for hours, "d" for days and "week" for weeks.

If the standardization format provided in certain cases is not satisfactory for researchers, the values may be modified as needed. The processing of the dataset can be redone since all material is freely available online, allowing for flexibility and customization according to specific requirements.

4.2. PRP-931k Dataset Overview

A separate subset of 1,000 instances has been reserved to serve as a gold standard, with each instance manually inspected. The expert corrected all naming and standardization mistakes and removed a few instances with missing key reactant inputs. The gold standard subset will be used as a second testing dataset. The main goal of utilizing such a dataset is to simulate a real-world scenario where the inputs are curated by a human. The 1,000 instances are always kept separate and therefore we consider the remaining 931,350 instances as the main dataset for our task. The dataset (931,350 instances) has been split into training (909,665 instances), validation (2,000 instances), and testing (20,000 instances) subsets.

Detailed PRP-931k dataset input analysis. The input character sequences vary in range from 15 to 256 characters in SMILES notation. The average input lengths were consistent across subsets: training (92.68 characters), validation (93.49 characters), and testing (92.79 characters). In total, the dataset has reactions involving single reactants (388,086 instances), two reactants (502,070 instances), and three reactants (400,057 instances). Reactions with 4 or more reactants are rare and the maximum number of reactants is 6. To clarify, single reactant reactions can include halogen exchange [52], reduction, oxidation, deacetylation [53], dehalogenation, and others that require either common additional agents or inorganic compounds, which are included in the procedure description (output). Most inputs contain a single reaction product (919,690 instances) while some contain two (11,599 instances). A convenient way of inspecting the dataset reactions is to analyze a TMAP [54] generated by combining our dataset with the Schneider 50k data set [49], which contains 50,000 annotated reactions that are distributed evenly over 50 reaction classes (10 super-classes). A TMAP generated from DRFP fingerprints [55] displays reactions grouped according to their super-class (Figure 1). All colors other than green indicate the super-class, while green indicates instances from our dataset. Green instances are distributed to cover most spaces where the annotated reactions are present. Additionally, several regions are exclusively green, indicating distinct groups not present in the 50k dataset. Throughout the training process, the models are exposed to a diverse range of reactions, crucial for ensuring versatility in their predictive capabilities. Although a more detailed examination is possible, such analysis is beyond the scope of the broad overview provided in this paper for our deep-learning task.

 doi:10.20944/preprints202411.1807.v1

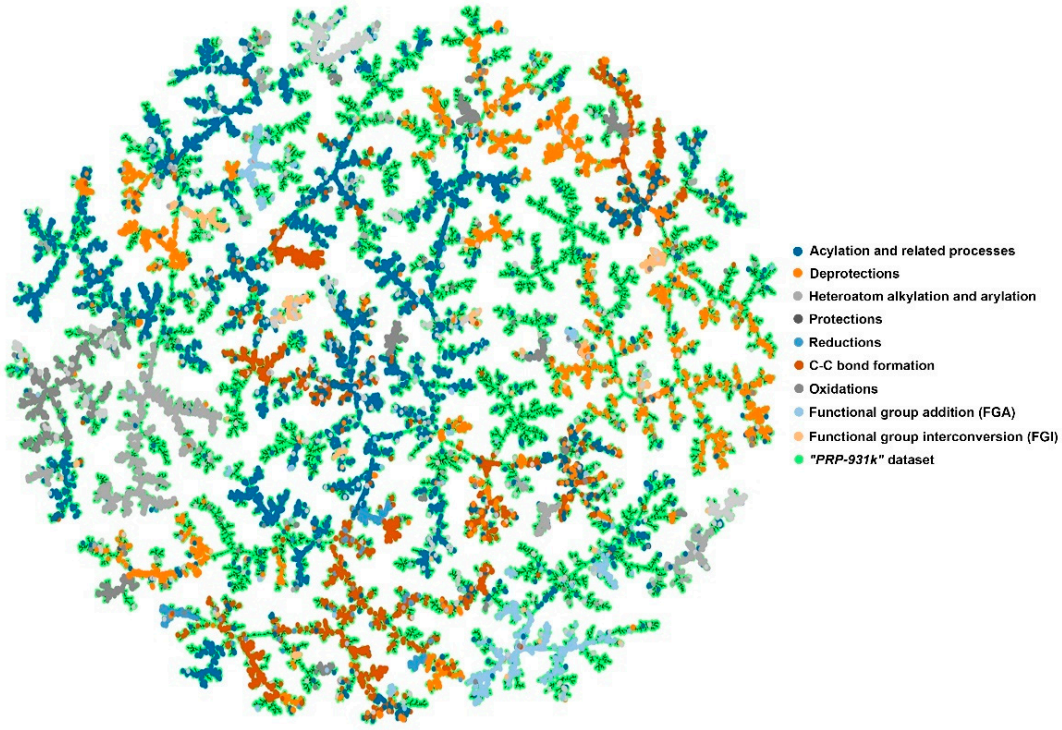

Figure 1. TMAP graph with Schneider 50k and PRP-931k combined.

Detailed PRP-931k dataset output analysis. The output character sequences range from 21 to 600 characters in length. The average input lengths were consistent across subsets: training (238.7 characters), validation (239.9 characters), and testing (238.3 characters). The synthesis procedures ranged from a minimum of 4 actions to a maximum of 29. The average action counts of the procedures across the subsets were also uniform: training (12.3 actions), validation (12.4 actions), and testing (12.3 actions), with a minimum action number of 2 and a maximum of 39 overall. A histogram visually represents the distribution of procedure lengths by number of actions (Figure 2). The histogram resembles a normal distribution with the majority of instances concentrated between 6 and 20 actions.

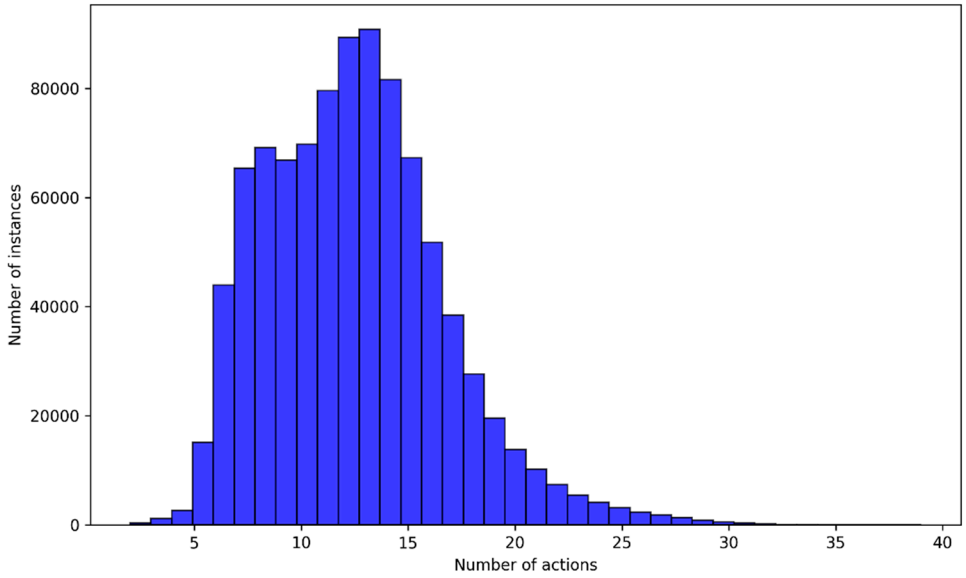

Figure 2. Histogram of the output procedure lengths by number of actions.

 doi:10.20944/preprints202411.1807.v1

The most common actions are *Add* (28.71% of all actions), *Stir* (9.27%), *Concentrate* (8.97%), *Yield* (8.24%), and *MakeSolution* (6.57%). The dataset is highly imbalanced in terms of actions; however, this is the feature of organic synthesis procedures mined from patent applications. Actions like *Partition* (0.39%), *Microwave* (0.20%), or *Sonicate* (0.02%) are used less often and therefore comprise only a small percentage of all actions. The full breakdown of action numbers can be found in Table 3.

Table 3. Breakdown of action numbers in the PRP-931k dataset.

Action name	Count	% of total	Action name	Count	% of total
Add	3282648	28.71	Reflux	112076	0.97
Stir	1061384	9.27	PH	104997	0.91
Concentrate	1028446	8.97	PhaseSeparation	93140	0.82
Yield	944022	8.24	DrySolid	88974	0.77
MakeSolution	757612	6.57	Quench	87872	0.77
Wash	696935	6.08	Wait	64040	0.56
CollectLayer	563969	4.91	Recrystallize	55498	0.48
Purify	553010	4.84	Partition	44888	0.39
Filter	522336	4.56	Degas	35813	0.32
DrySolution	514443	4.48	Triturate	31885	0.28
Extract	394015	3.42	Microwave	23091	0.20
SetTemperature	391722	3.44	Sonicate	2400	0.02

The action *Add* has a single parameter which is a name of what compound should be added. The most common added names are tokens for reactants and the "SLN" token, which stands for "solution" and usually follows the *MakeSolution* action. The most common non-token values are solvents like water, ethyl acetate, DCM, and methanol, and agents like TEA (Triethanolamine), and DIPEA (N,N-Diisopropylethylamine). 32,095 unique names can be added via this action in our current dataset, of which 646 occurred more than 100 times. The *MakeSolution* action can include multiple compounds and features solvents like THF, DCM, DMF, methanol, and water as the most frequently combined components besides the reactant tokens. 16,121 unique names can be combined, of which 359 occurred more than 100 times. Common work-up actions (*Wash*, *Extract*, *Purify*, *DrySolution*) feature solvents as parameters together with specific numeric notation. For example, the action Wash is most often done with brine, water, or sodium bicarbonate. Extract action is primarily done with ethyl acetate, dichloromethane, diethyl ether, or chloroform (51 solvents in total with frequency higher than 100). The *Purify* action features three parameters: solvent system, an indicator of whether the purification uses a gradient, and the ratio (two ratios – for the start and the end if the gradient is used). About 66.5% of all purification instances do not use gradient and use simple solvent systems for chromatographic purification, like ethyl acetate/hexane, methanol/dichloromethane, and acetonitrile/water (62 combinations in total with frequency higher than 100). Other instances do use a gradient for purification and feature common gradient ratio transitions, such as 100%:0% to 0%:100%, 100%:0% to 10%:90%, and 100%:0% to 50%:50%. Actions may also feature familiar inorganic drying agents, such as Na2SO4, MgSO4, MgSO2, K2CO3, and NaSO4 for solution drying (*DrySolution* action). The temperature and duration of synthesis can be controlled by *Stir*, *SetTemperature*, *Reflux*, or *Wait* actions. The *Stir* action has temperature and duration as the parameters. Some of the most common temperature points are 25° C (room temperature), 0° C, 80° C, or 100° C, while in terms of duration, we found 16 hours, 1 hour, 2 hours, 30 min, and 3 hours to be frequent, however, there is a large variety of possible time and temperature points (in total 555 for the duration and 439 for temperature). Action *SetTemperature* is used to designate the required temperature point usually after some operations have already taken place. The *Reflux* action does not specify temperature but features a duration parameter and is similar to the *Stir* action duration parameter.

PRP-931k dataset augmentation. The input was formed by combining the reactants with the products. The order of reactants is arbitrary and depends on the sequence in which they are entered. To improve the model adaptability, the instance order (input) of reactant 1 and reactant 2 can be

 doi:10.20944/preprints202411.1807.v1

rearranged by switching their positions to reactant 2, followed by reactant 1. The tokens that denote the reactants in the output also switch places to match the new order of input. The same process is relevant to the products and their tokens, however, there is a low number of reactions featuring more than one product. We have augmented the training subset with the new instances by permuting compounds in the dataset for all instances whenever feasible. The process results in an augmented training dataset with 1,535,423 instances and it has been used for our experiments. Models are trained to predict as accurately as they can regardless of the sequence in which different compounds are presented to them.

5. ML Approaches

Our study has examined the following generative ML approaches to assess their effectiveness in predicting steps and parameters of synthesis methods:

Seq2seq transformer model-based approach. The transformer model uses an encoder-decoder architecture adapted from the classic transformer framework [56]. For the implementation, the OpenNMT-py library [57] designed for efficient neural machine translation and sequence-to-sequence modeling tasks was utilized. The models are not pre-trained and we have utilized the "*TransformerBaseSharedEmbeddings*" model configuration and architecture from the OpenNMT library which has about 45 million parameters. SentencePiece tokenizers have been trained for vectorization of both input and output [58]. To investigate the effect of vocabulary size on the model's performance, we experimented with several transformer models labeled seq2seq_1000, seq2seq_3000, and seq2seq_8000 corresponding to vocabulary sizes of 1,000, 3,000, and 8,000, respectively. The models were trained with a learning rate of 0.001 and a batch size of 1024 which are used for neural machine translation [59].

BART model-based approach. The BART model, developed by Facebook AI [60], represents a significant advancement in transformer-based architecture. BART integrates attributes of both encoder and decoder mechanisms from the transformer architecture. It utilizes the Byte-Pair Encoding (BPE) tokenizer [61], which segments text into a set of subwords or tokens. Each token is then transformed into detailed context vectors, the length of which varies in accordance with the model's configuration. BART by Facebook has been trained on various natural language datasets, such as SQuAD [62], NMLI [63], and ELI5 [64]. The model employs a 512-length context vector and has 139 million parameters (BART-base model) in total. The fine-tuned tokenizer is created by fine-tuning the BART tokenizer on our data. Compared to T5 and FLAN-T5, which have smaller versions of the model, Facebook AI BART does not. Three more configurations were constructed to investigate the effects of model size on performance. An extra-small (BART-XS), small (BART-S), and medium (BART-M) sized models with sizes 18, 24, and 40 million parameters, respectively, were defined. The only difference in terms of testing is that there are no pre-trained versions of these configurations. The architecture parameters for each can be found in the project's repository (https://github.com/Mantas-it/Chem_Procedure_Prediction).

T5 model-based approach. T5 was developed by Google AI [65] and it is a transformer-based model, primarily designed for natural language processing tasks. T5 (Text-to-Text Transfer Transformer) differs from other models by redefining all NLP tasks as a text-to-text problem, treating both the input and output as text sequences, which facilitates the training process for various tasks. This model utilizes a distinctive method of pre-training on a vast corpus using a self-supervised objective, which involves predicting the masked text span to improve its comprehension and creation abilities. T5 is available in various sizes, ranging from small to extra-large, providing versatility in usage based on available resources and project requirements. The T5 model utilizes the SentencePiece tokenizer to segment words into subwords or tokens. Each token is then transformed into context vectors of different lengths based on the scale of the model. The model itself was pre-trained on datasets, such as GLUE [66], SuperGLUE [67], SQuAD, CoLA [68], and CB [69]. The model we use employs a 512-length context vector and has 60 million parameters (T5-small) or 223 million parameters (T5-base).

 doi:10.20944/preprints202411.1807.v1

FLAN-T5 model-based approach. FLAN-T5 (text-to-text transformer), created by Google [70], is an advanced version of the original T5 model which is focused on improving language model performance by including the capability to follow instructions. FLAN-T5 is proficient at comprehending and carrying out various text-based instructions due to pre-training on an extensive dataset that includes tasks presented in an instruction-based format. The model maintains the fundamental structure of T5, which includes using a tokenizer to break down text into smaller subwords or tokens for analysis. The tokenizer in FLAN-T5 is optimized to efficiently process instruction-heavy inputs, enabling the model to effectively interpret and execute the detailed commands present in the training data. The model is pre-trained on 1,836 fine-tuning tasks by combining four mixtures from previous studies: Muffin (80 tasks) [71], T0-SF (193 tasks) [72], and CoT [73]. The model employs a 512-length context vector and has 60 million parameters (FLAN-T5-small) or 223 million parameters (FLAN-T5-base).

molT5 model-based approach. The molT5 model is a specialized modification of the T5 framework that specifically targets the field of molecular science [74]. MolT5 is designed for chemoinformatics, converting chemical compound representations into a text format. This allows for text-to-text processing for applications like property prediction, synthesis planning, and molecule production. molT5 utilizes the T5 architecture to analyze a large dataset of chemical structures and characteristics, incorporating a comprehension of chemical context and interactions. The model was pre-trained using the C4 "Colossal Clean Crawled Corpus" [75] and Chemformer's 100 million SMILES dataset [30]. MolT5's adaptability and efficiency make it potentially valuable for our task due to its knowledge of chemical notation. The molT5 model is one of the few pre-trained models in the chemoinformatics field that can generate natural text. Several models can generate molecules; however, they typically only employ SMILES notation for output, making them unsuitable for our text-to-text task. Tokenizers can be adjusted for specific datasets; however, this often requires considerable changes to the pre-trained model, resulting in the loss of pre-training benefits. The model employs a 512-length context vector and has 76 million (molT5-small), 248 million (molT5-base), and 783 million parameters (molT5-large).

Models BART, T5, and FLAN-T5 were set up in three configurations: (1) a pre-trained tokenizer and a pre-trained model (denoted as *PP*) (2) a pre-trained tokenizer and a randomly initialized model (denoted as *PC*) (3) fine-tuned tokenizer and a randomly initialized model (denoted as *CC*). The molT5 model was exclusively trained using a pre-trained tokenizer and model to evaluate the impact of being pre-trained on chemical information on its performance for our task. The molT5 is a fine-tuned T5 model, so randomly initializing the model would only replicate tests already being conducted with the T5 model. The BART, T5, and FLAN-T5 models underwent fine-tuning with a learning rate of 0.0005, while the molT5 model was fine-tuned at a slightly higher learning rate of 0.001. Optimal performance for BART, T5, and FLAN-T5 was achieved between 18,000 and 21,000 steps, whereas for molT5, it ranged from 14,000 to 21,000 steps. In most instances, an effective batch size of 512 was utilized, contributing to the attainment of these results. All models underwent training and testing with Hugging Face's library [76] in a GPU-enabled environment.

6. Results

The experiments involved training and testing with the dataset (as outlined in Section 4) utilizing LLMs-based ML approaches described in Section 5. The models were evaluated by comparing the actual values from the testing dataset with the predicted values. The BLEU score metric was utilized. The BLEU score [77], originally developed to assess the quality of machine translation, is now an important metric in various NLP tasks for evaluating the quality of generated text. It provides a quantitative assessment of the similarity between the produced text and the reference text and is used for reaction procedure evaluation [8]. We also provide the ROUGE-L metric (Recall-Oriented Understudy for Gisting Evaluation - Longest Common Subsequence) [78]. These metrics are frequently utilized to assess the quality of generated text in comparison to reference texts.

 doi:10.20944/preprints202411.1807.v1

The BLEU score is derived by computing the geometric mean of adjusted n-gram ($N = 4$) precision (P_n) along with a brevity penalty (BP) to address the challenge of shorter generated texts in comparison to the reference:

$$BLEU = BP \times exp(\sum \left(\frac{\log(Pn)}{N}\right)) \quad (1)$$

where N represents the highest order of n-grams being analyzed, while the brevity penalty (BP) is determined as follows:

$$BP = \begin{cases} 1 & if\ c > r \\ e^{(1-r/c)} & if\ c \leq r \end{cases} \quad (2)$$

where r represents the combined length of the reference texts, while c represents the combined length of the created texts.

ROUGE-L score evaluates the longest common subsequence (LCS) between the generated text and the reference text, considering both recall and precision. The ROUGE-L score is computed in the following manner:

$$ROUGE - L = \frac{(1 + \beta^2) \times LCS\ Precision \times LCS\ Recall)}{(\beta^2 \times LCS\ Precision + LCS\ Recall)} \quad (3)$$

The weighting factor $\beta=1$ adjusts the importance of precision compared to recall. LCS Precision is the proportion of the length of the LCS to the length of the generated text, whereas LCS Recall is the proportion of the length of the LCS to the length of the reference text.

We also compute the normalized Levenshtein similarity [79] between each pair of reference and predicted answers, assessing similarity by accounting for the minimum edits needed to change one string into the other. This similarity is then evaluated against predefined thresholds (50% and 75%) to calculate the proportion of predictions reaching or surpassing each level of similarity. For example, the 75% threshold score indicates the proportion of sentence pairs with a normalized Levenshtein similarity of at least 0.75 or more.

Random and majority methods were chosen to determine a starting point for the trained models. The random method selects a random output from the training dataset, whereas the majority method produces the most frequently occurring sentence in the training dataset. Random and majority baselines achieve BLEU scores of 18.50 and 31.87, respectively. We also established a benchmark using the k-nearest neighbor algorithm (with $k=1$) as the reference point due to its simplicity, effectiveness, and suitability for chemical tasks. To vectorize the reactions and the input we have utilized DRFP fingerprints which are designed for chemical reactions. The methods yield a BLEU score of 25.32 with the Euclidean distance [80]. All test results are subsequently compared against these benchmarks to assess the performance of the fine-tuned models. If results exceed these baseline levels, it will indicate that the trained models have enough predictive capacity to be considered as possible solutions for our issue.

The results are presented in Table 4 of the testing subset for BLEU, ROUGE-L, normalized Levenshtein similarity match (with 50% and 75% thresholds), and BLEU for a manually checked gold standard subset of 1,000 instances (column name "BLEU g1k"). Due to relatively small differences in metric scores and a higher number of model configurations, we have opted not to present the results in a graph. The results table contains scores highlighted in bold for their particular type of model and in red, for best scores overall.

Table 4. Results of the testing dataset.

Name	BLEU	ROUGE-L	50% match	75% match	BLEU g1k
seq2seq_1000	39.61	51.15	30.99%	2.67%	41.16
seq2seq_3000	**43.81**	53.93	39.06%	4.03%	**45.29**
seq2seq_8000	43.45	53.92	38.77%	4.09%	44.42
BART-XS-PC	42.26	52.48	34.39%	3.10%	43.38
BART-XS-CC	43.99	53.98	38.28%	4.03%	45.32

 doi:10.20944/preprints202411.1807.v1

BART-S-PC	42.88	53.77	37.92%	4.08%	44.08
BART-S-CC	44.50	54.79	40.28%	4.88%	45.86
BART-M-PC	43.57	54.02	38.56%	4.48%	45.43
BART-M-CC	45.21	55.00	40.58%	5.27%	46.70
Bart-base-PP	45.12	55.36	42.27%	5.46%	46.05
Bart-base-PC	44.24	54.54	39.76%	4.58%	45.88
Bart-base-CC	**45.91**	56.00	43.06%	6.46%	**47.06**
T5-small-PP	**44.91**	53.76	38.05%	3.79%	45.41
T5-small-PC	44.79	54.47	39.63%	4.78%	46.50
T5-small-CC	43.74	54.34	38.93%	4.92%	45.40
T5-base-PP	44.63	54.14	39.12%	3.82%	46.21
T5-base-PC	44.69	54.37	39.16%	5.42%	**46.52**
T5-base-CC	43.39	53.87	37.59%	4.61%	45.17
FLAN-T5-small-PP	44.93	54.80	40.92%	4.20%	46.01
FLAN-T5-small-PC	45.05	54.32	39.66%	4.83%	45.91
FLAN-T5-small-CC	43.47	54.05	38.60%	4.34%	45.13
FLAN-T5-base-PP	45.62	55.11	41.72%	4.52%	46.76
FLAN-T5-base-PC	45.37	54.77	39.88%	6.21%	46.08
FLAN-T5-base-CC	**46.43**	56.02	42.74%	**7.91%**	**47.70**
molT5-small-PP	44.54	53.61	37.80%	3.65%	45.77
molT5-base-PP	46.38	55.51	43.30%	4.70%	47.79
molT5-large-PP	**47.75**	**56.74**	**45.73%**	6.46%	**48.52**

7. Discussion

Compared to the random and majority baseline scores (random - 31.87, majority - 18.50 BLEU), all evaluated model types with varying sizes and tokenizers demonstrate superior performance and predictive ability. In nearly all cases, models with a greater number of parameters, which are base-type models, demonstrate better performance. Models with a greater number of parameters should be able to interpret complicated input patterns and produce more precise predictions. The T5 model is an exception in this instance, while the other models have very comparable scores. The *T5-base-CC* always produces the lowest scores within its category, unlike other types where the *CC* (custom tokenizer and a randomly initialized model) configuration is typically better. The result might be related to the T5's unique architecture and hyperparameters, which may limit effective training with customized tokenizers. The results of classic transformer neural networks used in our text-to-text model closely match those of other smaller models such as *FLAN-T5-small* and *molT5-small*. The vocabulary size impacts the performance of the simple transformer model, with the best scores achieved using a vocabulary of 3000. However, the overall scores are not distinctive. When comparing BLEU and BLEU g1k scores, it is apparent that all BLEU g1k scores are higher due to the correct inputs provided to the model, as the gold standard testing dataset has fewer errors compared to the basic testing dataset. When key reactants are missing, the predictions may be significantly different from the ground truth which is reflected in the score. This implies that the model's performance in real-world scenarios could be slightly better than what the BLEU scores indicate.

When comparing the *PP* (pre-trained tokenizer and pre-trained model) with the *PC* (pre-trained tokenizer and randomly initiated model), we find either very similar performance or a reduction in scores. We believe that the pre-trained model's exposure to a vast amount of information during its first training phase enhances its capacity to interpret and analyze new information. In contrast, a model that starts with a randomly initialized state lacks this fundamental knowledge, which could restrict its ability to efficiently adjust to and understand the input. The *CC* (custom tokenizer and randomly initiated model) is usually the best option and achieves better scores than other configurations. Interestingly, a fine-tuned tokenizer on specific data can have a very positive effect and allow models to achieve better results than pre-trained models. Models that are pre-trained (T5, FLAN-T5, and BART) on a variety of texts that do not include domain-specific information are not

 doi:10.20944/preprints202411.1807.v1

particularly useful for our task. However, molT5 is pre-trained on chemical data and demonstrably performs well for our task, achieving top scores in two out of five metrics.

Three models stand out in terms of metric scores: *molT5-large-PP*, and *molT5-base-PP* and *FLAN-T5-base-CC*. No single model stood out as the best because none excelled in all metrics. The *molT5-large-PP* model performs best in terms of BLEU (47.75), ROUGE-L (56.74), 50% Levenshtein match (45.73%) and BLEU g1k (48.52). The highest scores of 75% Levenshtein match (7.91%) metric are achieved by the *FLAN-T5-base-CC* model (pre-trained molT5 model was fine-tuned). The *molT5-base-PP* configuration is not able to achieve a top score, however, it is close to the models mentioned earlier and is one of the few models that also achieves a BLEU g1k score over 47.

Observing the pattern that models with more parameters and the *CC* configuration perform better, we have also trained and tested the *FLAN-T5-large* (783M parameters) and *BART-large* (406M parameters) using the setup as mentioned above. The *BART-large-CC* did not converge despite different hyper-parameter setups and achieved a BLEU score of 21.11, which is below the baseline scores. The *FLAN-T5-large-CC* scored 41.81 BLEU with the testing dataset and most likely overfit, since the result is lower than most tested methods. Poor performance can be explained by the fact that these models are significantly bigger and the training dataset is not large enough to train models effectively. In previous works, our group has also conducted experiments with OpenAI's fine-tuned GPT-3.5-turbo using a single type of reaction (about 1,000 instances). The performance of such large LLMs is yet to be seen when using the entire *PRP-931k* dataset, but we have found that a trained *FLAN-T5-base* model was more effective for this specific chemistry-related task in our previous works. This suggests that large language models can present challenges and provide limited results in fine-tuning a computationally expensive task (text-to-text).

Metric scores alone do not provide a definitive indication of the best model, the quality of generated procedures, or the criteria for assessing future methodologies. We compared the top three models using additional metrics and analyzed the results to develop an in-depth assessment. The following metrics are tailored to each task and are only used to obtain a better understanding of the results generated by various models:

- **Ancillary compound coverage (ACC).** Procedures frequently require supplementary materials in addition to the core reactants, which are commonly known as ancillary compounds. The *Add* and *MakeSolution* actions can introduce additional materials into the reaction, such as solvents, inorganic substances, or reagents. The *ACC* metric is determined by compiling a list of all ancillary compounds involved in a reaction (from the *Add* and *MakeSolution* actions) and comparing how many of them are found in the predicted procedure. *ACC* is expressed in a formula 4, where A_p is the set of ancillary compounds in the predicted procedure, A_{gt} is the set of ancillary compounds in the ground truth procedure.

$$ACC = \frac{|A_p \cap A_{gt}|}{|A_{gt}|} * 100\% \quad (4)$$

For example, if the ground truth instance has THF, HBTU, and DIPEA added to the procedure, while the predicted are THF, HBTU, and NaOH, the coverage score would be 66.7% because THF and HBTU were correctly selected out of the three in the ground truth procedure. The metric aims to assess how well the models predict relevant materials that should be added to the reaction mixture because often the choice of the components suggests a specific reaction mechanism. However, it was difficult to assess how the score should be impacted if the predicted procedure contained other materials than the ground truth (NaOH in the example), and therefore no penalty was given in such cases. Also, the action *Add* appears not just at the beginning but may be present at various stages of the workup process. This means that if the predicted procedure chooses different methods for workup, which may not be incorrect, the ACC score will be lower. In about 3-4% of cases, where no *Add* or *MakeSolution* actions are present, this metric is not applicable. We calculate an average *ACC* score using the testing dataset (size: 20,000 instances). The scores were as follows: *molT5-large-PP* - 41.08%, *molT5-base-PP* - 31.61% and *FLAN-T5-base-CC* - 36.66%. Surprisingly, the model *BART-base-CC* which had the lowest metrics out of the top four models achieved the second-highest score of 40.37%. Other smaller models had scores of about 28%–32%, while most base models had scores of

 doi:10.20944/preprints202411.1807.v1

33%–38%. Notably, *molT5-base-PP* has achieved the highest score with our gold standard dataset but scored low with *ACC*, leading us to believe that it excels in capturing and applying the foundational chemical knowledge encoded in its training data for other steps, but it may not be as adept at incorporating ancillary compounds into its predictions. Alternatively, it may simply predict alternative agents or solvents, which may be more suitable in some cases. Generally, the scores of about 40% are within the expected range for the task in its current state. One factor that contributes to lower scores is the uncertainty with which certain agents could be chosen. For example, in one case, the base for the reaction is Cs2CO3 (ground truth), and models predict K2CO3, which is an acceptable prediction considering that Na2CO3 could possibly be used. Similar cases are present in organic chemistry, where a set of substances may be suitable in one case, especially solvents, and the ACC metric does not account for that.

- **Workup action and solvent coverage (WASC).** Workup actions are a set of procedures designed to isolate the desired product and alter its state in order to typically yield a solid product. Actions that require solvents for these manipulations are *Extract, Quench, Recrystallize, DrySolution,* and *Purify*. To assess the model's ability to accurately produce specific steps and determine the Workup Action and Solvent Coverage (WASC) score, compile a list of all the mentioned action names and solvents in a reaction. Then, compare this list with the predicted procedure to identify the matches. For example, if the ground truth instance has: Extract with ethyl acetate, DrySolution over MgSO4, while the predicted procedure has: Extract with ethyl acetate, DrySolution over Na2SO4, the coverage score would be 50% because only the extraction was correctly predicted with its parameter (solvent). WASC is expressed in formula 5, where W_p is the set of actions with compounds in the predicted procedure, W_{gt} is the set of actions with compounds in the ground truth procedure.

$$WASC = \frac{|W_p \cap W_{gt}|}{|W_{gt}|} * 100\% \quad (5)$$

The top three models achieve scores of around 30%: *FLAN-T5-base-CC* (34.00%, highest overall), *molT5-large-PP* (32.89%) and *molT5-base-PP* (31.52%). The scores for all the smaller models are around 28%. The *FLAN-T5-base-CC* model performs best and produces satisfactory results, given the strict nature of the metric. It should be noted that not all procedures contain workup actions, so this metric can only be used when the ground truth and the predicted instance both have at least one step (about 75%–80% of our testing subset). To measure whether it is challenging to predict the action steps without parameters, we can exclude the solvents and materials to perform similar calculations. We observe that the scores are considerably higher: *FLAN-T5-base-CC* (75.36%), *molT5-base-PP* (78.84%), and *molT5-large-PP* (78.23%). Such results once again reiterate the fact that predicting solvents or inorganic compounds for these actions is challenging since the options can be flexible. However, it also highlights the difficulty of measuring the scores, since it can be difficult to be certain whether an incorrect prediction was made or the degree to which some solvent is not suitable for an operation.

- **Average reaction temperature error (RTE).** In order to analyze the accuracy of reaction temperature prediction, we measure the average reaction temperature error. This is done by identifying the temperature point designated by the Stir action. If several options are possible, the highest (in absolute terms). Temperatures below zero are rare, but possible. The highest temperature point is chosen because it is assumed to be the temperature the reaction mixture must reach. The problem of simply picking temperature points randomly from the reaction procedure is that the *Stir* action might be mentioned in the beginning with 25° C as the temperature point and for a short duration (to add relevant reactants, for example) only to later be mentioned again with a temperature point of 140° C for the actual synthesis. Also, *SetTemperature* is an action used only to control the temperature of the reaction mixture, without any other implicit actions. It is often employed in the workup stage to adjust the temperature back to room temperature or to specific points (-40° C or -78° C), especially during crystallization processes. The action *Reflux* is the third option for designating the reaction temperature, but it is not explicitly written, and the reaction should be conducted at the boiling point of the solvent medium. Due to the reasons above, our experiment considered only the Stir action, which

 doi:10.20944/preprints202411.1807.v1

pertains to about 70% of the reactions in our testing dataset. *RTE* is calculated using formula 6, where n is number of samples, t_p – predicted temperature and t_g – ground truth.

$$RTE = \frac{1}{n}\sum |t_p - t_g| \quad (6)$$

The average temperature of reactions was around 45° C and the range of temperatures was from -100° C to 300° C. The reaction temperature average error for the top models was: *FLAN-T5-base-CC* (22.51), *molT5-base-PP* (24.54), and *molT5-large-PP* (19.04). The *BART-base-CC* (18.39) model scored best, and among the tested cases, it had an average error of 5° C in 54% of cases, which is a high degree of accuracy. Acceptable temperature ranges may vary based on the procedure's end goal and the compounds' stability. In our case, we can observe that there is room for improvement, but we must recognize that the source training procedures do not necessarily present the optimal temperature points for the reaction. It is at least assumed that the temperature points are good enough for the reaction to take place in the first place. This means that while temperature predictions may be too high or too low, it does not mean that the reaction will not occur. There might be cases where an optimal temperature point is predicted, and there also might be cases where the suggestions are incompatible with the compounds or the mechanisms of synthesis.

- **Average synthesis duration error (SDE).** Lastly, to improve the understanding of synthesis duration prediction accuracy, we analyzed the average errors in synthesis duration. The synthesis duration encompasses the combined durations of both *Stir* and *Reflux* actions within the procedure. All the time that is given by the actions is converted to hours and added together. We calculate the average error using the testing dataset where it is available (about 90–95% of the testing dataset). *SDE* is calculated using formula 7, where n is number of samples, d_p – predicted temperature and d_g – ground truth.

$$SDE = \frac{1}{n}\sum |d_p - d_g| \quad (7)$$

The average duration of reactions was around 13.3 hours and the range of duration was from 5 minutes to 10 days. The *SDE* for the top models was: *FLAN-T5-base-CC* (10.04), *molT5-base-PP* (10.26), and *molT5-large-PP* (10.04). The average error of about 10 hours is quite high, and there certainly is room for improvement. The *BART-base-CC* had the top score by a small margin (9.98, best overall) and an average error of less than 3 hours in 38% of all suitable instances and less than 1 hour in 23.5% of cases. Generally, the length of the synthesis is variable because it depends on the nature of the chemicals, the selection of solvents, the reaction temperature, and the scale of the synthesis. In addition, the end of the synthesis operations is usually monitored using analytical equipment; therefore, if incorrect predictions are made, the operation's duration can be adjusted accordingly. In fact, regular or automated reaction analysis may render the duration prediction irrelevant. Additionally, we should acknowledge that if the predicted temperature is higher and appropriate for the reaction, the reaction rate will naturally increase, necessitating an adjustment to the duration. However, our analysis of the four additional metrics reveals that future research should focus on addressing the synthesis duration, given the higher level of errors compared to other aspects of the procedure.

Analysis of common and uncommon metrics can be difficult because of the specificity of reaction procedures. To illustrate some of the challenges, two examples in Table 5 are given from the gold standard subset of the *BART-base-CC* predictions with the matching ground truth. In the first example, we see a shorter ground truth procedure than the prediction. The prediction itself is similar in the beginning and correctly adds HATU, DIPEA, and DMF into one mixture. It incorrectly predicts the duration but also includes a workup sequence (extraction, washing, drying, and filtration). When comparing the two with most metrics, we should see lower scores; however, in practical terms, the core of the ground truth procedure has been predicted correctly. If the workup is not needed, it simply can be skipped. If it is needed, the predicted version suggests a possible path. In example two, the inverse is true, and the ground truth includes a longer workup. The core of the reaction is quite accurate; only the *Purify* action is done after the synthesis is complete. It is important to address this aspect of chemical reaction procedure prediction in future research to have greater control over prediction generation. While the length of the generation can be affected by the length penalty

 doi:10.20944/preprints202411.1807.v1

constants, it is not necessarily the solution since the length is not what matters most. For example, the action *Purify* calls for chromatographic purification, which can be used in place of an entire sequence of workup actions. Also, certain laboratories might not have the required equipment for the predicted sequence of actions, so a generated alternative ending of the procedure could also be very beneficial for scientists, which currently is not possible. In addition, some actions do not require parameters, like *Purify*, but it would be useful to generate them. It would be an improvement if the dataset could be manipulated in such a way that it does not compromise the model's capability and allows it to generate these additional parameters.

Table 5. Two examples of *BART-base-CC* predictions.

***BART-base-CC* predictions**	**Ground truth**
MAKESOLUTION with $R1$ and $R2$ and HATU and DIPEA and DMF; ADD SLN; STIR for 16 h at 25° C; ADD water; EXTRACT with Ethyl acetate 3 x; COLLECTLAYER organic; WASH with brine; DRYSOLUTION over Na2SO4; FILTER keep filtrate; CONCENTRATE; PURIFY; YIELD $P1$.	MAKESOLUTION with $R1$ and $R2$ and HATU and TEA and DMF; ADD SLN; STIR for 3 h at 25° C; PURIFY; YIELD $P1$.
MAKESOLUTION with $R1$ and THF; ADD SLN; ADD TBAF/THF; STIR for 1 h at 25° C; CONCENTRATE; PURIFY gradient 0:1-1:9 CH3OH:DCM; YIELD $P1$.	MAKESOLUTION with $R1$ and THF; ADD SLN; ADD TBAF; ADD THF; STIR for 20 min at 0° C; EXTRACT with NaHCO3; EXTRACT with Ethyl acetate; WASH with brine; DRYSOLUTION over MgSO4; FILTER keep filtrate; CONCENTRATE; PURIFY gradient 1:49-1:19 CH3OH:DCM; YIELD $P1$.

When conducting error analysis case by case, we observe that most of the reactions (about two-thirds) fall into the category of being close to the ground truth but with differences in temperature and/or duration, sequence of component addition, and solvent or ancillary substance choice. Table 6 illustrates three such cases (*molT5-large-PP* model). It does not mean that the procedures are not valid, some of the discrepancies may hinder the synthesis process, while others could enhance the rate or efficiency.

Table 6. Three predicted and ground truth instances. The main differences are highlighted.

***molT5-large-PP* predictions**	**Ground truth**
MAKESOLUTION with $R1$ and DCM and TFA; ADD SLN; STIR for 2 h at 25° C; CONCENTRATE; YIELD $P1$.	MAKESOLUTION with $R1$ and DCM; ADD SLN; ADD TFA; STIR for 1 h at 25° C; CONCENTRATE; YIELD $P1$.
MAKESOLUTION with $R1$ and CH3OH; ADD SLN; ADD Pd-C under N2; STIR for 12 h at 25° C; PURIFY 1:9 CH3OH:chloroform; FILTER keep filtrate; WASH with CH3OH; CONCENTRATE; PURIFY; YIELD $P1$.	MAKESOLUTION with $R1$ and CH3OH; ADD SLN; ADD Pd-C; STIR for 16 h at 25° C under H2; FILTER keep filtrate; CONCENTRATE; YIELD $P1$.
MAKESOLUTION with $R2$ and Ethyl acetate; ADD SLN; ADD $R1$; ADD HOBT; ADD EDC; ADD TEA; STIR for 16 h at 25° C; ADD water; PHASESEPARATION; COLLECTLAYER organic; WASH with NaHCO3; WASH with brine; DRYSOLUTION; YIELD $P1$.	MAKESOLUTION with $R2$ and DCM; ADD SLN; ADD $R1$; ADD EDAC; ADD HOBT; ADD TEA; STIR for 16 h at 25° C; ADD DCM; WASH with water; WASH with brine; DRYSOLUTION over Na2SO4; CONCENTRATE; PURIFY; YIELD $P1$.

In conclusion, while the additional metric scores (ACC, WASC, RTE, and SDE) that we have calculated provide some insight into the quality of the predicted procedures and allow us to generally

 doi:10.20944/preprints202411.1807.v1

rank the performance of models, it is still unclear which models are best. The main difficulty is the evaluation of the procedures, and the only conclusive way to ensure the quality of the generated sequences is laboratory testing. Such testing, however, can be extensive and require significant resources. We found three models that are top performers on different metrics, and the best suggestion is for users to find the most suitable for their particular use case. In general, it is interesting that LLMs can be used to train models for specific chemical problems. One of the contributors to the success is a relatively large and clean dataset. The notation and naming normalization help models learn better and avoid instances where the same chemicals may have several versions, which may all look like different chemicals to the model. Models that are trained on a variety of chemical synthesis classes may provide a starting point for fine-tuning tasks with curated small datasets where the goal is a precise generation of the procedure for one or more reaction types. In the end, models that can recognize various chemical reaction types and can consequently generate a rich procedure for chemists or automatic systems are necessary for automation overall in chemistry.

8. Conclusions

The paper evaluates and compares various deep-learning-based algorithms that convert SMILES molecular notation into rich procedural texts of organic synthesis reactions. We have developed a novel dataset for this purpose, which is now publicly accessible online. Our research encompasses a comparison of BART, T5, FLAN-T5, molT5, and classic seq2seq transformers all of different sizes and configurations. Most of the models were investigated under three configurations: (1) with both a pre-trained tokenizer and model (denoted as *PP*); (2) a pre-trained tokenizer with a model of randomly initialized weights (denoted as *PC*) (3) a fine-tuned tokenizer with a model of randomly initialized weights (denoted as *CC*). Experimental investigation revealed three top performers: *FLAN-T5-base-CC*, *molT5-base-PP*, and *molT5-large-PP*. The *molT5-large-PP* model performs best in terms of BLEU (47.75), ROUGE-L (56.74), 50% Levenshtein match (45.73%) and BLEU g1k (48.52). The highest scores of 75% Levenshtein match (7.91%) metric are achieved by the *FLAN-T5-base-CC* model (the pre-trained molT5 model was fine-tuned). The *molT5-base-PP* configuration is not able to achieve top scores, however, it performed competitively close to the aforementioned top models. The results demonstrate the capability of language models to predict chemical synthesis procedures involving 24 possible distinct actions, many of which include various parameters like solvents, reaction agents, temperature, duration, solvent ratios, and other specific parameters. We demonstrate that only when the core reactants are used as input, models learn to correctly predict what ancillary components need to be included in the resulting procedure. These insights are valuable for AI researchers and chemists, suggesting that curated datasets and language model fine-tuning techniques can be tailored for specific reaction classes and practical applications. We anticipate that future research will focus on (1) investigation of specific methods to control the generation procedures with additional parameters for enhanced outcome specificity and sequence control; (2) augmentation of the dataset to encompass a broader range of organic reactions to increase model predictive capabilities, and (3) conducting real-world laboratory validations to assess the practical applicability and reliability of our models. Our goal is to refine these models further based on feedback from empirical trials in real-world settings.

Author Contributions: Conceptualization, M.V. and J.K.-D.; methodology, M.V. and J.K.-D.; software, M.V.; validation, M.V. and J.K.-D.; data curation, M.V.; writing—original draft preparation, M.V.; writing—review and editing, J.K.-D.; visualization, M.V.; supervision, J.K.-D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Datasets and code can be found at (https://github.com/Mantas-it/Chem_Procedure_Prediction).

 doi:10.20944/preprints202411.1807.v1

Conflicts of Interest: authors declare no conflicts of interest.

References

1. Goodman, J. Computer Software Review: Reaxys. *Journal of Chemical Information and Modeling* **2009**, *49*, 2897–2898. https://doi.org/10.1021/ci900437n.
2. Dou, B.; Zhu, Z.; Merkurjev, E.; Ke, L.; Chen, L.; Jiang, J.; Zhu, Y.; Liu, J.; Zhang, B.; Wei, G.-W. Machine Learning Methods for Small Data Challenges in Molecular Science. *Chemical Reviews* **2023**, *123*, 8736–8780. https://doi.org/10.1021/acs.chemrev.3c00189.
3. Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. Neural Message Passing for Quantum Chemistry. *arXiv* **2017**. https://doi.org/10.48550/ARXIV.1704.01212.
4. Mater, A. C.; Coote, M. L. Deep Learning in Chemistry. *Journal of Chemical Information and Modeling* **2019**, *59*, 2545–2559. https://doi.org/10.1021/acs.jcim.9b00266.
5. Shilpa, S.; Kashyap, G.; Sunoj, R. B. Recent Applications of Machine Learning in Molecular Property and Chemical Reaction Outcome Predictions. *The Journal of Physical Chemistry A* **2023**, *127*, 8253–8271. https://doi.org/10.1021/acs.jpca.3c04779.
6. Wiercioch, M.; Kirchmair, J. DNN-PP: A Novel Deep Neural Network Approach and Its Applicability in Drug-Related Property Prediction. *Expert Systems with Applications* **2023**, *213*, 119055. https://doi.org/10.1016/j.eswa.2022.119055.
7. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q.; Zhou, D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *arXiv* **2022**. https://doi.org/10.48550/ARXIV.2201.11903.
8. Vaucher, A. C.; Schwaller, P.; Geluykens, J.; Nair, V. H.; Iuliano, A.; Laino, T. Inferring Experimental Procedures from Text-Based Representations of Chemical Reactions. *Nature Communications* **2021**, *12*. https://doi.org/10.1038/s41467-021-22951-1.
9. Schwaller, P.; Gaudin, T.; Lányi, D.; Bekas, C.; Laino, T. "Found in Translation": Predicting Outcomes of Complex Organic Chemistry Reactions Using Neural Sequence-to-Sequence Models. *Chemical Science* **2018**, *9*, 6091–6098. https://doi.org/10.1039/c8sc02339e.
10. Allen, R. R. Hydrogenation: Principles and Catalysts. *Journal of the American Oil Chemists' Society* **1968**, *45*. https://doi.org/10.1007/bf02655520.
11. Hansen, E. C.; Perkins, R.; Ripin, D. H. B. Oxidations. *Practical Synthetic Organic Chemistry* **2020**, 513–562. https://doi.org/10.1002/9781119448914.ch10.
12. Ruggeri, S. G.; Caron, S.; Dubé, P.; Ide, N. D.; Wiglesworth, K. E. P.; Ragan, J. A.; Yu, S. Reductions. *Practical Synthetic Organic Chemistry* **2020**, 455–511. https://doi.org/10.1002/9781119448914.ch9.
13. Surya, K. De.; Condensation Reaction. *Applied Organic Chemistry* **2020**, 69–109. https://doi.org/10.1002/9783527828166.ch2
14. Ameri, A. M. Principles of Nucleophilic Substitution. *American International Journal of Cancer Studies* **2019**, *1*, 11–18. https://doi.org/10.46545/aijcs.v1i1.48.
15. Best, K. T.; Li, D.; Helms, E. D. Molecular Modeling of an Electrophilic Addition Reaction with "Unexpected" Regiochemistry. *Journal of Chemical Education* **2017**, *94*, 936–940. https://doi.org/10.1021/acs.jchemed.6b00488.
16. Liu, J.; Lu, L.; Wood, D.; Lin, S. New Redox Strategies in Organic Synthesis by Means of Electrochemistry and Photochemistry. *ACS Central Science* **2020**, *6*, 1317–1340. https://doi.org/10.1021/acscentsci.0c00549.
17. Singh Gujral, S.; Khatri, S.; Riyal, P.; Gahlot, V. Suzuki Cross Coupling Reaction- A Review. *Indo Global Journal of Pharmaceutical Sciences* **2012**, *02*, 351–367. https://doi.org/10.35652/igjps.2012.41.
18. Nicolaou, K. C.; Snyder, S. A.; Montagnon, T.; Vassilikogiannakis, G. The Diels-Alder Reaction in Total Synthesis. *Angewandte Chemie International Edition* **2002**, *41*, 1668–1698. https://doi.org/10.1002/1521-3773(20020517)41:10<1668::aid-anie1668>3.0.co;2-z.
19. Perrin, C. L.; Chang, K.-L. The Complete Mechanism of an Aldol Condensation. *The Journal of Organic Chemistry* **2016**, *81*, 5631–5635. https://doi.org/10.1021/acs.joc.6b00959.
20. Schobert, R. Applications of the Wittig Reaction in the Synthesis of Heterocyclic and Carbocyclic Compounds. *Organophosphorus Reagents* **2004**, 129–150. https://doi.org/10.1093/oso/9780198502623.003.0005.
21. Peltzer, R. M.; Gauss, J.; Eisenstein, O.; Cascella, M. The Grignard Reaction – Unraveling a Chemical Puzzle. *Journal of the American Chemical Society* **2020**, *142*, 2984–2994. https://doi.org/10.1021/jacs.9b11829.
22. Lam, A. Y. S.; Li, V. O. K. Chemical Reaction Optimization: A Tutorial. *Memetic Computing* **2012**, *4*, 3–17. https://doi.org/10.1007/s12293-012-0075-1.
23. Taylor, C. J.; Pomberger, A.; Felton, K. C.; Grainger, R.; Barecka, M.; Chamberlain, T. W.; Bourne, R. A.; Johnson, C. N.; Lapkin, A. A. A Brief Introduction to Chemical Reaction Optimization. *Chemical Reviews* **2023**, *123*, 3089–3126. https://doi.org/10.1021/acs.chemrev.2c00798.

24. Vaškevičius, M.; Kapočiūtė-Dzikienė, J.; Šlepikas, L. Generative LLMs in Organic Chemistry: Transforming Esterification Reactions into Natural Language Procedures. *Applied Sciences* **2023**, *13*, 13140. https://doi.org/10.3390/app132413140.
25. He, C.; Zhang, C.; Bian, T.; Jiao, K.; Su, W.; Wu, K.-J.; Su, A. A Review on Artificial Intelligence Enabled Design, Synthesis, and Process Optimization of Chemical Products for Industry 4.0. *Processes* **2023**, *11*, 330. https://doi.org/10.3390/pr11020330.
26. Fabian, B.; Edlich, T.; Gaspar, H.; Segler, M.; Meyers, J.; Fiscato, M.; Ahmed, M. Molecular Representation Learning with Language Models and Domain-Relevant Auxiliary Tasks. *arXiv* **2020**. https://doi.org/10.48550/ARXIV.2011.13230.
27. Frey, N. C.; Soklaski, R.; Axelrod, S.; Samsi, S.; Gómez-Bombarelli, R.; Coley, C. W.; Gadepally, V. Neural Scaling of Deep Chemical Models. *Nature Machine Intelligence* **2023**, *5*, 1297–1305. https://doi.org/10.1038/s42256-023-00740-3.
28. Sterling, T.; Irwin, J. J. ZINC 15 – Ligand Discovery for Everyone. *Journal of Chemical Information and Modeling* **2015**, *55*, 2324–2337. https://doi.org/10.1021/acs.jcim.5b00559.
29. Chilingaryan, G.; Tamoyan, H.; Tevosyan, A.; Babayan, N.; Khondkaryan, L.; Hambardzumyan, K.; Navoyan, Z.; Khachatrian, H.; Aghajanyan, A. BARTSmiles: Generative Masked Language Models for Molecular Representations. *arXiv* **2022**. https://doi.org/10.48550/ARXIV.2211.16349.
30. Irwin, R.; Dimitriadis, S.; He, J.; Bjerrum, E. J. Chemformer: A Pre-Trained Transformer for Computational Chemistry. *Machine Learning: Science and Technology* **2022**, *3*, 015022. https://doi.org/10.1088/2632-2153/ac3ffb.
31. Mercado, R.; Rastemo, T.; Lindelöf, E.; Klambauer, G.; Engkvist, O.; Chen, H.; Jannik Bjerrum, E. Graph Networks for Molecular Design. *Machine Learning: Science and Technology* **2021**, *2*, 025023. https://doi.org/10.1088/2632-2153/abcf91.
32. Jin, W.; Barzilay, R.; Jaakkola, T. Hierarchical Graph-to-Graph Translation for Molecules. *arXiv* **2019**. https://doi.org/10.48550/ARXIV.1907.11223.
33. Bagal, V.; Aggarwal, R.; Vinod, P. K.; Priyakumar, U. D. MolGPT: Molecular Generation Using a Transformer-Decoder Model. *Journal of Chemical Information and Modeling* **2021**, *62*, 2064–2076. https://doi.org/10.1021/acs.jcim.1c00600.
34. Ross, J.; Belgodere, B.; Chenthamarakshan, V.; Padhi, I.; Mroueh, Y.; Das, P. Large-Scale Chemical Language Representations Capture Molecular Structure and Properties. *Nature Machine Intelligence* **2022**, *4*, 1256–1264. https://doi.org/10.1038/s42256-022-00580-7.
35. Lu, J.; Zhang, Y. Unified Deep Learning Model for Multitask Reaction Predictions with Explanation. *Journal of Chemical Information and Modeling* **2022**, *62*, 1376–1387. https://doi.org/10.1021/acs.jcim.1c01467.
36. Livne, M.; Miftahutdinov, Z.; Tutubalina, E.; Kuznetsov, M.; Polykovskiy, D.; Brundyn, A.; Jhunjhunwala, A.; Costa, A.; Aliper, A.; Aspuru-Guzik, A.; et al. Nach0: Multimodal Natural and Chemical Languages Foundation Model. *arXiv* **2023**. https://doi.org/10.48550/ARXIV.2311.12410.
37. Xie, T.; Wan, Y.; Zhou, Y.; Huang, W.; Liu, Y.; Linghu, Q.; Wang, S.; Kit, C.; Grazian, C.; Zhang, W.; et al. Creation of a Structured Solar Cell Material Dataset and Performance Prediction Using Large Language Models. *Patterns* **2024**, *5*, 100955. https://doi.org/10.1016/j.patter.2024.100955.
38. Xiong, Z.; Wang, D.; Liu, X.; Zhong, F.; Wan, X.; Li, X.; Li, Z.; Luo, X.; Chen, K.; Jiang, H.; et al. Pushing the Boundaries of Molecular Representation for Drug Discovery with the Graph Attention Mechanism. *Journal of Medicinal Chemistry* **2019**, *63*, 8749–8760. https://doi.org/10.1021/acs.jmedchem.9b00959.
39. Wang, X.; Li, Y.; Qiu, J.; Chen, G.; Liu, H.; Liao, B.; Hsieh, C.-Y.; Yao, X. RetroPrime: A Diverse, Plausible and Transformer-Based Method for Single-Step Retrosynthesis Predictions. *Chemical Engineering Journal* **2021**, *420*, 129845. https://doi.org/10.1016/j.cej.2021.129845.
40. Ahmad, W.; Simon, E.; Chithrananda, S.; Grand G.; Ramsundar B. ChemBERTa-2: Towards Chemical Foundation Models. *arXiv* **2022**. https://arxiv.org/abs/2209.01712.
41. Wang, Y.; Pang, C.; Wang, Y.; Jin, J.; Zhang, J.; Zeng, X.; Su, R.; Zou, Q.; Wei, L. Retrosynthesis Prediction with an Interpretable Deep-Learning Framework Based on Molecular Assembly Tasks. Nature Communications, 2023, 14. https://doi.org/10.1038/s41467-023-41698-5.
42. Liu, C.-H.; Korablyov, M.; Jastrzębski, S.; Włodarczyk-Pruszyński, P.; Bengio, Y.; Segler, M. RetroGNN: Fast Estimation of Synthesizability for Virtual Screening and De Novo Design by Learning from Slow Retrosynthesis Software. *Journal of Chemical Information and Modeling* **2022**, *62*, 2293–2300. https://doi.org/10.1021/acs.jcim.1c01476.
43. Sacha, M.; Błaż, M.; Byrski, P.; Dąbrowski-Tumański, P.; Chromiński, M.; Loska, R.; Włodarczyk-Pruszyński, P.; Jastrzębski, S. Molecule Edit Graph Attention Network: Modeling Chemical Reactions as Sequences of Graph Edits. *Journal of Chemical Information and Modeling* **2021**, *61*, 3273–3284. https://doi.org/10.1021/acs.jcim.1c00537.
44. Zhang, D.; Liu, W.; Tan, Q.; Chen, J.; Yan, H.; Yan, Y.; Li, J.; Huang, W.; Yue, X.; Ouyang, W.; et al. ChemLLM: A Chemical Large Language Model. *arXiv* **2024**. https://doi.org/10.48550/ARXIV.2402.06852.

45. Boiko, D. A.; MacKnight, R.; Gomes, G. Emergent Autonomous Scientific Research Capabilities of Large Language Models. *arXiv* **2023**. https://doi.org/10.48550/ARXIV.2304.05332.
46. Jablonka, K. M.; Schwaller, P.; Ortega-Guerrero, A.; Smit, B. Leveraging Large Language Models for Predictive Chemistry. *Nature Machine Intelligence* **2024**, *6*, 161–169. https://doi.org/10.1038/s42256-023-00788-1.
47. Bran, A. M.; Cox, S.; Schilter, O.; Baldassari, C.; White, A. D.; Schwaller, P. ChemCrow: Augmenting Large-Language Models with Chemistry Tools. *arXiv* **2023**. https://doi.org/10.48550/ARXIV.2304.05376.
48. Vaškevičius, M.; Kapočiūtė-Dzikienė, J.; Vaškevičius, A.; Šlepikas, L. Deep Learning-Based Automatic Action Extraction from Structured Chemical Synthesis Procedures. *PeerJ Computer Science* **2023**, *9*, e1511. https://doi.org/10.7717/peerj-cs.1511.
49. Schneider, N.; Stiefl, N.; Landrum, G. A. What's What: The (Nearly) Definitive Guide to Reaction Role Assignment. *Journal of Chemical Information and Modeling* **2016**, *56*, 2336–2346. https://doi.org/10.1021/acs.jcim.6b00564.
50. Jin, W.; Coley, C. W.; Barzilay, R.; Jaakkola, T. Predicting Organic Reaction Outcomes with Weisfeiler-Lehman Network. *arXiv* **2017**. https://doi.org/10.48550/ARXIV.1709.04555.
51. Lowe, D. M. Extraction of Chemical Structures and Reactions from the Literature. *University of Cambridge Repository* **2013**. https://doi.org/10.17863/CAM.16293.
52. Sheppard, T. D. Metal-Catalysed Halogen Exchange Reactions of Aryl Halides. *Organic & Biomolecular Chemistry* **2009**, *7*, 1043. https://doi.org/10.1039/b818155a.
53. Nasrollahzadeh, M.; Soleimani, F.; Nezafat, Z.; Orooji, Y.; Ahmadpoor, F. Facile Synthesis of Cu Nanoparticles Supported on Magnetic Lignin-Chitosan Blend as a Highly Effective Catalyst for the Preparation of 5-Aryl-1H-Tetrazoles. *Biomass Conversion and Biorefinery* **2021**, *13*, 12451–12465. https://doi.org/10.1007/s13399-021-02005-8.
54. Probst, D.; Reymond, J.-L. Visualization of Very Large High-Dimensional Data Sets as Minimum Spanning Trees. *Journal of Cheminformatics* **2020**, *12*. https://doi.org/10.1186/s13321-020-0416-x.
55. Probst, D.; Schwaller, P.; Reymond, J.-L. Reaction Classification and Yield Prediction Using the Differential Reaction Fingerprint DRFP. *Digital Discovery* **2022**, *1*, 91–97. https://doi.org/10.1039/d1dd00006c.
56. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**. https://doi.org/10.48550/ARXIV.1706.03762.
57. Klein, G.; Kim, Y.; Deng, Y.; Senellart, J.; Rush, A. M. OpenNMT: Open-Source Toolkit for Neural Machine Translation. *arXiv* **2017**. https://doi.org/10.48550/ARXIV.1701.02810.
58. Kudo, T.; Richardson, J. SentencePiece: A Simple and Language Independent Subword Tokenizer and Detokenizer for Neural Text Processing. *arXiv* **2018**. https://doi.org/10.48550/ARXIV.1808.06226.
59. Negri, M.; Turchi, M.; Bertoldi, N.; Federico, M. Online Neural Automatic Post-Editing for Neural Machine Translation. Proceedings of the Fifth Italian Conference on Computational Linguistics CLiC-it 2018, Torino, Italy, 10-12 December 2018. https://doi.org/10.4000/books.aaccademia.3534.
60. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. BART: Denoising Sequence-to-Sequence Pre-Training for Natural Language Generation, Translation, and Comprehension. *arXiv* **2019**. https://doi.org/10.48550/ARXIV.1910.13461.
61. Zouhar, V.; Meister, C.; Gastaldi, J. L.; Du, L.; Vieira, T.; Sachan, M.; Cotterell, R. A Formal Perspective on Byte-Pair Encoding. *arXiv* **2023**. https://doi.org/10.48550/ARXIV.2306.16837.
62. Rajpurkar, P.; Zhang, J.; Lopyrev, K.; Liang, P. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv* **2016**. https://doi.org/10.48550/ARXIV.1606.05250.
63. Williams, A.; Nangia, N.; Bowman, S. R. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. *arXiv* **2017**. https://doi.org/10.48550/ARXIV.1704.05426.
64. Fan, A.; Jernite, Y.; Perez, E.; Grangier, D.; Weston, J.; Auli, M. ELI5: Long Form Question Answering. arXiv 2019. https://doi.org/10.48550/ARXIV.1907.09190.
65. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P. J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv* **2019**. https://doi.org/10.48550/ARXIV.1910.10683.
66. Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S. R. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. arXiv 2018. https://doi.org/10.48550/ARXIV.1804.07461.
67. Wang, A.; Pruksachatkun, Y.; Nangia, N.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S. R. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. *arXiv* **2019**. https://doi.org/10.48550/ARXIV.1905.00537.
68. Warstadt, A.; Singh, A.; Bowman, S. R. Neural Network Acceptability Judgments. *Transactions of the Association for Computational Linguistics* **2019**, *7*, 625–641. https://doi.org/10.1162/tacl_a_00290.
69. De Marneffe, M.-C.; Simons, M.; Tonhauser, J. (2019). The CommitmentBank: Investigating projection in naturally occurring discourse. Proceedings of Sinn Und Bedeutung 23, Barcelona, Spain, 1 May 2019. https://doi.org/10.18148/SUB/2019.V23I2.601

70. Chung, H. W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, Y.; Wang, X.; Dehghani, M.; Brahma, S.; et al. Scaling Instruction-Finetuned Language Models. *arXiv* **2022**. https://doi.org/10.48550/ARXIV.2210.11416.
71. Wei, J.; Bosma, M.; Zhao, V. Y.; Guu, K.; Yu, A. W.; Lester, B.; Du, N.; Dai, A. M.; Le, Q. V. Finetuned Language Models Are Zero-Shot Learners. *arXiv* **2021**. https://doi.org/10.48550/ARXIV.2109.01652.
72. Sanh, V.; Webson, A.; Raffel, C.; Bach, S. H.; Sutawika, L.; Alyafeai, Z.; Chaffin, A.; Stiegler, A.; Scao, T. L.; Raja, A.; et al. Multitask Prompted Training Enables Zero-Shot Task Generalization. *arXiv* **2021**. https://doi.org/10.48550/ARXIV.2110.08207.
73. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q.; Zhou, D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *arXiv* **2022**. https://doi.org/10.48550/ARXIV.2201.11903.
74. Edwards, C.; Lai, T.; Ros, K.; Honke, G.; Cho, K.; Ji, H. Translation between Molecules and Natural Language. *arXiv* **2022**. https://doi.org/10.48550/ARXIV.2204.11817.
75. Dodge, J.; Sap, M.; Marasović, A.; Agnew, W.; Ilharco, G.; Groeneveld, D.; Mitchell, M.; Gardner, M. Documenting Large Webtext Corpora: A Case Study on the Colossal Clean Crawled Corpus. *arXiv* **2021**. https://doi.org/10.48550/ARXIV.2104.08758.
76. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. HuggingFace's Transformers: State-of-the-Art Natural Language Processing. arXiv 2019. https://doi.org/10.48550/ARXIV.1910.03771.
77. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.-J. BLEU. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02, Philadelphia, Pennsylvania, USA, 2001. https://doi.org/10.3115/1073083.1073135.
78. Lin, C. ROUGE: A Package for Automatic Evaluation of Summaries. Annual Meeting of the Association for Computational Linguistics, Barcelona, Spain, 2004. https://aclanthology.org/W04-1013
79. Po, D. K. Similarity Based Information Retrieval Using Levenshtein Distance Algorithm. *International Journal of Advances in Scientific Research and Engineering* **2020**, *06*, 06–10. https://doi.org/10.31695/ijasre.2020.33780.
80. De Boom, C.; Van Canneyt, S.; Bohez, S.; Demeester, T.; Dhoedt, B. Learning Semantic Similarity for Very Short Texts. IEEE International Conference on Data Mining Workshop, Atlantic City, New Jersey, USA, 15-17 November, 2015. https://doi.org/10.1109/icdmw.2015.86.