












































14 Examples of How LLMs Can Transform Materials Science and Chemistry: A Reflection on a Large Language Model Hackathon

Kevin Maik Jablonka ^{1,*} Qianxiang Ai ^{2,†} Alexander Al-Feghali ^{3,†}
 Shruti Badhwar ^{4,†} Joshua D. Bocarsly ^{5,†} Andres M Bran ^{6,7,†}
 Stefan Bringuier ^{8,†} L. Catherine Brinson ^{9,†} Kamal Choudhary ^{10,†}
 Defne Circi ^{9,†} Sam Cox ^{11,†} Wibe A. de Jong ^{12,†} Matthew L. Evans ^{13,14,†}
 Nicolas Gastellu ^{3,†} Jerome Genzling ^{3,†} María Victoria Gil ^{15,†}
 Ankur K. Gupta ^{12,†} Zhi Hong ^{16,†} Alishba Imran,^{17,†} Sabine Kruschwitz ^{18,†}
 Anne Labarre ^{3,†} Jakub Lála ^{19,†} Tao Liu ^{3,†} Steven Ma ^{3,†}
 Sauradeep Majumdar ^{1,†} Garrett W. Merz ^{20,†} Nicolas Moitessier ^{3,†}
 Elias Moubarak ^{1,†} Beatriz Mouriño ^{1,†} Brenden Pelkie ^{21,†}
 Michael Pieler ^{22,23,†} Mayk Caldas Ramos ^{11,†} Bojana Ranković ^{6,7,†}
 Samuel G. Rodrigues ^{19,†} Jacob N. Sanders ^{24,†} Philippe Schwaller ^{6,7,†}
 Marcus Schwarting,^{25,†} Jiale Shi ^{2,†} Berend Smit ^{1,†} Ben E. Smith ^{5,†}
 Joren Van Herck ^{1,†} Christoph Völker ^{18,†} Logan Ward ^{26,†}
 Sean Warren ^{3,†} Benjamin Weiser ^{3,†} Sylvester Zhang,^{3,†}
 Xiaoqi Zhang ^{1,†} Ghezal Ahmad Zia ^{18,†} Aristana Scourtas ²⁷
 KJ Schmidt,²⁷ Ian Foster ²⁸ Andrew D. White ¹¹ and Ben Blaiszik ^{27,‡}

¹*Laboratory of Molecular Simulation (LSMO),*

Institut des Sciences et Ingénierie Chimiques,

Ecole Polytechnique Fédérale de Lausanne (EPFL), Sion, Valais, Switzerland.

²*Department of Chemical Engineering,*

Massachusetts Institute of Technology,

Cambridge, Massachusetts 02139, United States.

³*Department of Chemistry, McGill University, Montreal, Quebec, Canada.*

⁴*Reincarnate Inc.*

⁵*Yusuf Hamied Department of Chemistry,*

University of Cambridge, Lensfield Road,

Cambridge, CB2 1EW, United Kingdom.

- ⁶*Laboratory of Artificial Chemical Intelligence (LIAC),
Institut des Sciences et Ingénierie Chimiques,
Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland.*
- ⁷*National Centre of Competence in Research (NCCR) Catalysis,
Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland.*
- ⁸*Independent Researcher, San Diego, CA, United States.*
- ⁹*Mechanical Engineering and Materials Science, Duke University, United States.*
- ¹⁰ *Material Measurement Laboratory,
National Institute of Standards and Technology, Maryland, 20899, United States.*
- ¹¹*Department of Chemical Engineering,
University of Rochester, United States.*
- ¹²*Applied Mathematics and Computational Research Division,
Lawrence Berkeley National Laboratory,
Berkeley, CA 94720, United States.*
- ¹³*Institut de la Matière Condensée et des Nanosciences (IMCN),
UCLouvain, Chemin des Étoiles 8, Louvain-la-Neuve, 1348, Belgium.*
- ¹⁴*Matgenix SRL, 185 Rue Armand Bury, 6534 Gozée, Belgium.*
- ¹⁵*Instituto de Ciencia y Tecnología del Carbono (INCAR),
CSIC, Francisco Pintado Fe 26, 33011 Oviedo, Spain.*
- ¹⁶*Department of Computer Science, University of Chicago,
Chicago, Illinois 60637, United States.*
- ¹⁷*Computer Science, University of California,
Berkeley, Berkeley CA 94704, United States.*
- ¹⁸*Bundesanstalt für Materialforschung und -prüfung,
Unter den Eichen 87, 12205 Berlin, Germany.*
- ¹⁹*Francis Crick Institute, 1 Midland Rd,
London NW1 1AT, United Kingdom.*
- ²⁰*American Family Insurance Data Science Institute,
University of Wisconsin-Madison, Madison WI 53706, United States.*
- ²¹*Department of Chemical Engineering,
University of Washington, Seattle, WA 98105, United States.*
- ²²*OpenBioML.org*
- ²³*Stability.AI*

²⁴*Department of Chemistry and Biochemistry,
University of California, Los Angeles, CA 90095, United States.*

²⁵*Department of Computer Science,
University of Chicago, Chicago IL 60490, United States.*

²⁶*Data Science and Learning Division,
Argonne National Lab, United States.*

²⁷*Globus, University of Chicago, Data Science and Learning Division,
Argonne National Lab, United States.*

²⁸*Department of Computer Science, University of Chicago,
Data Science and Learning Division,
Argonne National Lab, United States.*

Abstract

Large-language models (LLMs) such as GPT-4 caught the interest of many scientists. Recent studies suggested that these models could be useful in chemistry and materials science. To explore these possibilities, we organized a hackathon.

This article chronicles the projects built as part of this hackathon. Participants employed LLMs for various applications, including predicting properties of molecules and materials, designing novel interfaces for tools, extracting knowledge from unstructured data, and developing new educational applications.

The diverse topics and the fact that working prototypes could be generated in less than two days highlight that LLMs will profoundly impact the future of our fields. The rich collection of ideas and projects also indicates that the applications of LLMs are not limited to materials science and chemistry but offer potential benefits to a wide range of scientific disciplines.

*Electronic address: mail@kjablonka.com

†These authors contributed equally

‡Electronic address: blaiszik@uchicago.edu

I. INTRODUCTION

The intersection of machine learning (ML) with chemistry and materials science has witnessed remarkable advancements in recent years [1–9]. Much progress has been made in using ML to, e.g., accelerate simulations [10, 11] or to directly predict properties or compounds for a given application [12]. Thereby, developing custom, hand-crafted models for any given application is still common practice. Since science rewards doing novel things for the first time, we now face a deluge of tools and machine-learning models for various tasks. These tools commonly require input data in their own *rigid, well-defined form* (e.g., a table with specific columns or images from a specific microscope with specific dimensions). Further, they typically also report their outputs in non-standard and sometimes proprietary forms.

This rigidity sharply contrasts the standard practice in the (experimental) molecular and materials sciences, which is intrinsically *fuzzy and highly context-dependent* [13]. For instance, researchers have many ways to refer to a molecule (e.g., IUPAC name, conventional name, simplified molecular-input line-entry system (SMILES) [14]) and to report results and procedures. In particular, for the latter, it is known that small details such as the order of addition or the strength of stirring (e.g., “gently” vs. “strongly”) are crucial in determining the outcome of reactions. We do not have a natural way to deal with this fuzziness, and often a conversion into structured tabular form (the conventional input format for ML models) is impossible. Our current “solution” is to write conversion programs and chain many tools with plenty of application-specific “glue code” to enable scientific workflows. However, this fuzziness chemistry and heterogeneity of tools have profound consequences: A never-ending stream of new file formats, interfaces, and interoperability tools exists, and users cannot keep up with learning [15]. In addition, almost any transformation of highly context-dependent text (e.g., description of a reaction procedure) into structured, tabular form will lead to a loss of information.

One of the aims of this work is to demonstrate how large language models (LLMs) such as the generative pretrained transformer (GPT)-4 [16–21], can be used to address these challenges. Foundation models such as GPTs are general-purpose technologies [22] that can solve tasks they have not explicitly been trained on [23, 24], use tools [25–27], and be grounded in knowledge bases [28, 29]. As we also show in this work, they provide new pathways of exploration, new opportunities for flexible interfaces, and may be used to effectively solve certain tasks themselves; e.g., we envision LLMs enabling non-

experts to program (“malleable software”) using natural language as the “programming language” [30], extract structured information, and create digital assistants that make our tools interoperable—all based on unstructured, natural-language inputs.






























Inspired by early reports on the use of these LLMs in chemical research [31–34], we organized a virtual hackathon event focused on understanding the applicability of LLMs to materials science and chemistry. The hackathon aimed to explore the multifaceted applications of LLMs in materials science and chemistry and encourage creative solutions to some of the pressing challenges in the field. This article showcases some of the projects (Table I) developed during the hackathon.

One of the conclusions of this work is that without these LLMs, such projects would take many months. The diversity of topics these projects address illustrates the broad applicability of LLMs; the projects touch many different aspects of materials science and chemistry, from the wet lab to the computational chemistry lab, software interfaces, and even the classroom. While the examples below are not yet polished products, the simple observation that such capabilities could be created in hours underlines that we need to start thinking about how LLMs will impact the future of materials science, chemistry, and beyond [35]. The diverse applications show that LLMs are here to stay and are likely a foundational capability that will be integrated into most aspects of the research process. Even so, the pace of the developments highlights that we are only beginning to scratch the surface of what LLMs can do for chemistry and materials science.

Table I lists the different projects created in this collaborative effort across eight countries and 22 institutions (SI section V). One might expect that 1.5 days of intense collaborations would, at best, allow a cursory exploration of a topic. However, the diversity of topics and the diversity in the participants’ expertise, combined with the need to deliver a working prototype (within a short window of time) and the ease of prototyping with LLMs, generated not only many questions but also pragmatic solutions. In the remainder of this article, we focus on the insights we obtained from this collective effort. For the details of each project, we refer to the SI.

We have grouped the projects into four categories: *1. predictive modeling*, *2. automation and novel interfaces*, *3. knowledge extraction*, and *4. education*. The projects in the *predictive modeling* category use LLMs for classification and regression tasks—and also investigate ways to incorporate established concepts such as Δ -ML [36] or novel concepts such as “fuzzy” context into the modeling. The *automation and novel interfaces*

Table I: *Overview of the developed tools and links to source code repositories.* Full descriptions of the projects can be found in the Supplementary Material.

name	authors	links
<i>Predictive modeling</i>		
Accurate Molecular Energy Predictions	Ankur K. Gupta, Garrett W. Merz, Alishba Imran, Wibe A. de Jong	 ankur56/ChemLoRA  10.5281/zenodo.8104930
Text2Concrete	Sabine Kruschwitz, Christoph Völker, Ghezal Ahmad Zia	 gheزالahmad/LLMs-for-the-Design-of-Sustainable-Concretes  10.5281/zenodo.8091195
Molecule Discovery by Context	Zhi Hong, Logan Ward	 globuslabs/ScholarBERT-XL  10.5281/zenodo.8122087
Genetic algorithm without genes	Benjamin Weiser, Jerome Genzling, Nicolas Gastellu, Sylvester Zhang, Tao Liu, Alexander Al-Feghali, Nicolas Moitessier, Anne Labarre, Steven Ma	 BenjaminWeiser/LLM-Guided-GA  10.5281/zenodo.8125541
Text-template paraphrasing	Michael Pieler	 micpie/text-template-paraphrasing-chemistry  10.5281/zenodo.8093615
<i>Automation and novel interfaces</i>		
BOLLaMa	Bojana Ranković, Andres M. Bran, Philippe Schwaller	 doncamilom/BOLLaMa  10.5281/zenodo.8096827
sMolTalk	Jakub Lála, Sean Warren, Samuel G. Rodrigues	 jakublala/smoltalk-legacy  10.5281/zenodo.8081749
MAPI-LLM	Mayk Caldas Ramos, Sam Cox, Andrew White	 maykcaldas/MAPI_LLM  maykcaldas/MAPI_LLM  10.5281/zenodo.8097336
Conversational ELN interface (Whinchat)	Joshua D. Bocarsly, Matthew L. Evans and Ben E. Smith	 the-grey-group/datalab  10.5281/zenodo.8127782
<i>Knowledge Extraction</i>		
InsightGraph	Defne Circi, Shruti Badhwar	 defnecirci/InsightGraph  10.5281/zenodo.8092575
Extracting Structured Data from Free-form Organic Synthesis Text	Qianxiang Ai, Jacob N. Sanders, Jiale Shi, Stefan Bringuier, Brenden Pelkie, Marcus Schwarting	 qai222/LLM_organic_synthesis  10.5281/zenodo.8091902
TableToJson: Structured information from scientific data in tables	María Victoria Gil	 vgvinter/TableToJson  10.5281/zenodo.8093731
AbstractToTitle & TitleToAbstract: text summarization and generation	Kamal Choudhary	 usnistgov/chemnlp  10.5281/zenodo.8122419
<i>Education</i>		
I-Digest	Beatriz Mouriño, Elias Moubarak, Joren Van Herck, Sauradeep Majumdar, Xiaoqi Zhang	 XiaoqZhang/i-Digest  10.5281/zenodo.8080962

projects show that natural language might be the universal “glue” connecting our tools—perhaps in the future, we will need not to focus on new formats or standards but rather use natural language descriptions to connect across the existing diversity and different modalities [35].

LLMs can also help make knowledge more accessible, as the projects in the “knowledge extraction” category show; they can extract structured information from unstructured text. In addition, as the project in the “education” category shows, LLMs can also offer new educational opportunities.

A. Predictive modeling

Predictive modeling is a common application of ML in chemistry. Based on the language-interfaced fine-tuning (LIFT) framework [37], Jablonka et al. [32] have shown

that LLMs can be employed to predict various chemical properties, such as solubility or HOMO-LUMO gaps based on line representations of molecules such as self-referencing embedded strings (SELFIES) [38, 39] and SMILES. Taking this idea even further, Ramos et al. [34] used this framework (with in-context learning (ICL)) for Bayesian optimization—guiding experiments without even training models.

The projects in the following build on top of those initial results and extend them in novel ways as well as by leveraging established techniques from quantum machine learning.

Given that these encouraging results could be achieved with and without fine-tuning (i.e., updates to the weights of the model) for the language-interfaced training on tabular datasets, we use the term LIFT also for ICL settings in which structured data is converted into text prompts for an LLM.

a. Molecular Energy Predictions

A critical property in quantum chemistry is the atomization energy of a molecule, which gives us the basic thermochemical data used to determine a molecule’s stability or reactivity. State-of-the-art quantum chemical methods (i.e., G4(MP2) [40]) can predict this energy with an accuracy of 0.034 eV (or 0.79 kcal/mol) [41, 42]. This accuracy is similar to, and in some cases even better than, the accuracy that can be reached experimentally. This motivated Ramakrishnan et al. [41] and Narayanan et al. [42] to compute these atomization energies for the 134,000 molecules in the QM9-G4MP2 dataset.

The Berkeley-Madison team (Ankur Gupta, Garrett Merz, Alishba Imran, and Wibe de Jong) used this dataset to fine-tune different LLMs using the LIFT framework. The team investigated if they could use an LLM to predict atomization energies with chemical accuracy. Jablonka et al. [32] emphasized that these LLMs might be particularly useful in the low-data limit. Here, we have a relatively large dataset, so it is an ideal system to gather insights into the performance of these models for datasets much larger than those used by Jablonka et al. [32].

The Berkeley-Madison team showed that the LIFT framework based on simple line representations such as SMILES and SELFIES [38, 39] can yield good predictions ($R^2 > 0.95$ on a holdout test set), that are, however, still inferior to dedicated models that have access to 3D information [43, 44]. An alternative approach to achieve chemical accuracy with LLMs tuned only on string representations is to leverage a Δ -ML scheme [45] in which the LLM is tuned to predict the difference between G4(MP2) and B3LYP [46]

Table II: *LIFT for molecular atomization energies on the QM9-G4MP2 dataset.* Metrics for models tuned on 90% of the QM9-G4MP2 dataset (117,232 molecules), using 10% (13,026 molecules) as a holdout test set. GPTChem refers to the approach reported by Jablonka et al. [32], GPT-2-LoRA to PEFT of the GPT-2 model using LoRA. The results indicate that the LIFT framework can also be used to build predictive models for atomization energies, that can reach chemical accuracy using a Δ -ML scheme. Baseline performance (mean absolute error reported by Ward et al. [44]): 0.0223 eV for FCHL-based prediction of GP4(MP2) atomization energies and 0.0045 eV (SchNet) and 0.0052 eV (FCHL) for the Δ -ML scheme.

mol. repr. & framework	G4(MP2) Atomization Energy		(G4(MP2)-B3LYP) Atomization Energy	
	R ²	MAD / eV	R ²	MAD / eV
SMILES: GPTChem	0.984	0.99	0.976	0.03
SELFIES: GPTChem	0.961	1.18	0.973	0.03
SMILES: GPT2-LoRA	0.931	2.03	0.910	0.06
SELFIES: GPT2-LoRA	0.959	1.93	0.915	0.06

energies. Table II shows that good agreement could be achieved for the Δ -ML approach. This showcases how techniques established for conventional ML on molecules can also be applied with LLMs.

Importantly, this approach is not limited to the OpenAI application programming interface (API). With PEFT with LoRA [47] of the GPT-2 model [48], one can also obtain comparable results on consumer hardware. These results make the LIFT approach widely more accessible and allow research to the LIFT framework for chemistry without relying on OpenAI.

b. Text2Concrete

Concrete is the most used construction material, and the mechanical properties and climate impact of these materials are a complex function of the processing and formulation. Much research is focused on formulations of concrete that are less CO₂ intensive. [49] To expedite the design process, e.g., by prioritizing experiments using ML-predictions, data-driven methods have been investigated by Völker et al. [50] The Text2Concrete team (Sabine Kruschwitz, Christoph Völker, and Ghezal Ahmad Zia) explored, based on data reported by Rao and Rao [51], whether LLMs can be used for this task. This data set provides 240 alternative, more sustainable, concrete formulations and their respective compressive strengths. From a practical point of view, one would like to have a model that can predict the compressive strength of the concrete as a function of its formulation.

Interestingly, the largest LLMs can already give predictions without any fine-tuning. These models can “learn” from the few examples provided by the user in the prompt. Of course, such a few-shot approach (or ICL, [20]) does not allow for the same type of

optimization as fine-tuning, and one can therefore expect it to be less accurate. However, Ramos et al. [34] showed that this method could perform well—especially if only so few data points are available such that fine-tuning is not a suitable approach.

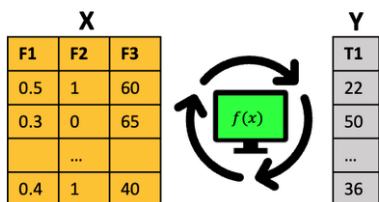
For their case study, the Text2Concrete team found a predictive accuracy comparable to a Gaussian process regression (GPR) model (but inferior to a random forest (RF) model). However, one significant advantage of LLMs is that one can *easily incorporate context*. The Text2Concrete team used this to include well-established design principles like the influence of the water-to-cement ratio on strength (Figure 1) into the modeling by simply stating the relationship between the features in natural language (e.g., “high water/cement ratio reduces strength”). This additional context reduced the outliers and outperformed the RF model (R^2 of 0.67 and 0.72, respectively).

The exciting aspect is that this is a typical example of domain knowledge that cannot be captured with a simple equation incorporable into conventional modeling workflows. Such “fuzzy” domain knowledge, which may sometimes exist only in the minds of researchers, is common in chemistry and materials science. With the incorporation of such “fuzzy” knowledge into LIFT-based predictions using LLMs, we now have a novel and very promising approach to leverage such domain expertise that we could not leverage before. Interestingly, this also may provide a way to test “fuzzy” hypotheses, e.g., a researcher could describe the hypothesis in natural language and see how it affects the model accuracy. While the Text2Concrete example has not exhaustively analyzed how “fuzzy” context alterations affect LLM performance, we recognize this as a key area for future research that could enhance the application of LLMs and our approach to leveraging “fuzzy” domain knowledge within materials science.

c. Molecule Discovery by Context

Much context is available in the full text of scientific articles. This has been exploited by Tshitoyan et al. [52] who used a Word2Vec [53] approach to embed words into a vector space. Word2Vec does so by tasking a model to predict for a word the probability for all possible next words in a vocabulary. In this way, word embeddings capture syntactic and semantic details of lexical items (i.e., words). When applied to material science abstracts, the word embeddings of compounds such as Li_2CuSb could be used for materials discovery by measuring their distance (cosine similarity) to concepts such as “thermo-electric”. [54] However, traditional Word2Vec, as used by Tshitoyan et al. [52], only produces *static* embeddings, which remain unchanged after training. Word embeddings

Training a Prediction Model between vectorized Concrete Formulations (X) and Labels (Y)



In-Context Learning

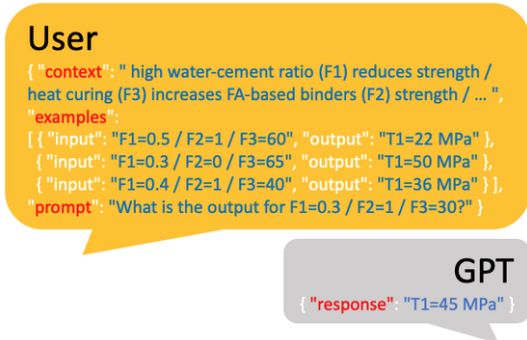


Figure 1: *Using LLMs to predict the compressive strength of concretes.* An illustration of the conventional approach for solving this task, i.e., training classical prediction models using ten training data points as tabular data (left). Using the LIFT framework LLMs can also use tabular data and leverage context information provided in natural language (right). The context can be “fuzzy” design rules often known in chemistry and materials science but hard to incorporate in conventional ML models. Augmented with this context and ten training examples, ICL with LLM leads to a performance that outperforms baselines such as RFs or GPR.

extracted from an LLM, on the other hand, are *contextualized* on the specific sequence (sentence) in which they are used and, therefore, can more effectively capture the contexts of words within a given corpus [55]. Inspired by this, the GlobusLabs team (Zhi Hong, Logan Ward) investigated if similar embeddings could be used to discover hydrogen carrier molecules, that are relevant for energy storage applications. For this, they leverage the ScholarBert model [56] trained on a large corpus of scientific articles collected by the Public.Resource.Org nonprofit organization. For different candidate molecules, they searched for sentences in the Public.Resource.Org corpus and used the average of the embeddings of these sentences as a fingerprint of the molecules. Given those fingerprints, they could rank molecules by how close their fingerprints are to the ones of known hydrogen carrier molecules. Visual inspection indicates that the selected molecules indeed bear similarities to known hydrogen carrier molecules.

d. Text template paraphrasing

In the LIFT framework used in the examples above, the data are embedded in so-called prompt templates that can have a form like `What is the <property name> of <representation>?`, where the texts in chevrons are placeholders that are replaced with actual values such as “solubility” and “2-acetyloxybenzoic acid”. In the low-data regime, data points are “wasted” by the model needing to learn the syntax of the prompt templates. In the big-data regime, in contrast, one might worry that the model loses some of its general language modeling abilities by always dealing with the same template. This

naturally raises the question if one can augment the dataset to mitigate these problems—thereby leveraging again, similar to Δ -ML, a technique that has found use in conventional ML previously. However, text-based data are challenging to augment due to their discrete nature and the fact that the augmented text still needs to be syntactically and semantically valid. Interestingly, as Michael Pieler (OpenBioML.org and Stability.AI) shows (and as has been explored by Dai et al. [57]), it turns out that LLMs can also be used to address this problem by simply prompting an LLM (e.g., GPT-4 or Anthropic’s Claude) to paraphrase a prompt template (see SI section ID).

This approach will allow us to automatically create new paraphrased high-quality prompts for LIFT-based training very efficiently—to augment the dataset and reduce the risk of overfitting to a specific template. Latter might be particularly important if one still wants to retain general language abilities of the LLMs after finetuning.

e. Genetic algorithm using an LLM

Genetic algorithms are popular methods for generating new structures; they are evolutionary algorithms in which building blocks (e.g., fragments of SMILES strings) are iteratively crossed over, mutated, and subjected to other genetic operations to evolve structures with better performance (such as catalysts with higher conversion) [58]. The efficiency of such a genetic algorithm often depends on how well the genes and genetic operations match the underlying chemistry. For example, if the algorithm replaces atom by atom, it may take several generations before a complete functional group is replaced.

One might hypothesize that LLMs can make the evolution process more efficient, e.g., by using an LLM to handle the reproduction. One might expect that inductive biases in the LLM help create recombined molecules which are more chemically viable, maintaining the motifs of the two parent molecules better than a random operation.

The team from McGill University (Benjamin Weiser, Jerome Genzling, Nicolas Gastellu, Sylvester Zhang, Tao Liu, Alexander Al-Feghali, Nicolas Moitessier) set out the first steps to test this hypothesis (Figure 2). In initial experiments, they found that GPT-3.5, without any finetuning, can fragment molecules provided as SMILES at rotatable bonds with a success rate of 70%. This indicates that GPT-3.5 understands SMILES strings and aspects of their relation to the chemical structures they represent. Subsequently, they asked the LLMs to fragment and recombine two given molecules. The LLM frequently created new combined molecules with fragments of each species which were reasonable chemical structures more often than a random SMILES string combining

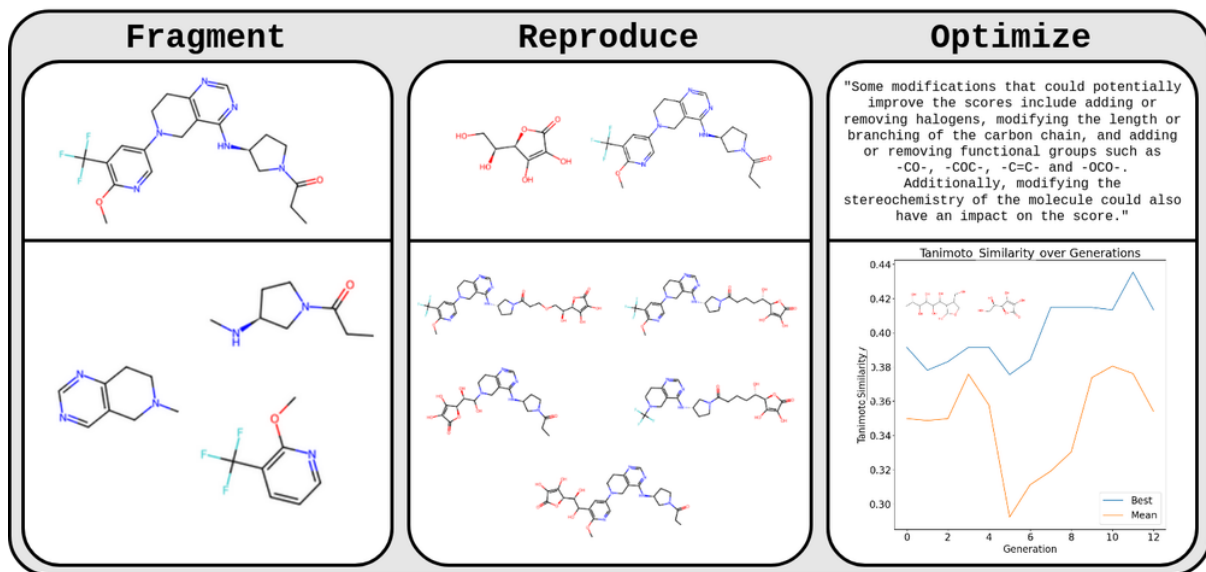


Figure 2: *GA using an LLM*. This figure illustrates how different aspects of a GA can be performed by an LLM. GPT-3.5 was used to fragment, reproduce, and optimize molecules represented by SMILES strings. The first column illustrated how an LLM can fragment a molecule represented by a SMILES string (input molecule on top, output LLM fragments below). The middle column showcases how an LLM can reproduce/mix two molecules as is done in a GA (input molecule on top, output LLM below). The right column illustrates an application in which an LLM is used to optimize molecules given their SMILES and an associated score. The LLM suggested potential modifications to optimize molecules. The plot shows best (blue) and mean (orange) Tanimoto similarity to Vitamin C per LLM produced generations.

operation (two independent organic chemists judged the LLM-GA-generated molecules to be chemically reasonable in $32/32$ cases, but only in $21/32$ cases for the random recombination operation).

Encouraged by these findings, they prompted an LLM with 30 parent molecules and their performance scores (Tanimoto similarity to vitamin C) with the task to come up with n new molecules that the LLM "believes" to improve the score. A preliminary visual inspection suggests that the LLM might produce chemically reasonable modifications. Future work will need to systematically investigate potential improvements compared to conventional GAs.

The importance of the results of the McGill team is that they indicate that these LLMs (when suitably conditioned) might not only reproduce known structures but generate new structures that make chemical sense [32, 59].

A current limitation of this approach is that most LLMs still struggle to output valid SMILES without explicit fine-tuning [33]. We anticipate that this problem might be mitigated by building foundation models for chemistry (with more suitable tokeniza-

tion [60, 61]), as, for instance, the ChemNLP project of OpenBioML.org attempts to do (<https://github.com/OpenBioML/chemnlp>). In addition, the context length limits the number of parent molecules that can be provided as examples.

Overall, we see that the flexibility of the natural language input and the in-context learning abilities allows using LLMs in very different ways—to very efficiently build predictive models or to approach molecular and material design in entirely unprecedented ways, like by providing context—such as “fuzzy” design rules—or simply prompting the LLM to come up with new structures. However, we also find that some “old” ideas, such as Δ -ML and data augmentation, can also be applied in this new paradigm.

B. Automation and novel interfaces

Yao et al. [62] and Schick et al. [25] have shown that LLMs can be used as agents that can autonomously make use of external tools such as Web-APIs—a paradigm that some call MRKL (pronounced “miracle”) Systems—modular reasoning, knowledge, and language systems [26]. By giving LLMs access to tools and forcing them to think step-by-step [63], we can thereby convert LLMs from hyperconfident models that often hallucinate to systems that can reason based on observations made by querying robust tools. As the technical report for GPT-4 highlighted [64], giving LLMs access to tools can lead to emergent behavior, i.e., enabling the system to do things that none of its parts could do before. In addition, this approach can make external tools more accessible—since users no longer have to learn tool-specific APIs. It can also make tools more interoperable—by using natural language instead of “glue code” to connect tools.

This paradigm has recently been used by Bran et al. [65] to create digital assistants that can call and combine various tools such as Google search and the IBM RXN retrosynthesis tool when prompted with natural language. Boiko et al. [66] used a similar approach and gave LLMs access to laboratories via cloud lab APIs. In their system, the LLM could use external tools to plan a synthesis, which it could execute using the cloud lab.

a. *MAPI-LLM*

Electronic structure calculations have reached such a high level of accuracy that one can answer questions like “Is the material AnByCz stable?” Indeed, the Materials Project [67] stores thermodynamic data on many components from which one can obtain a reasonable estimate of the stability of a given material. Or, if the material is not in the database, one can do a simulation instead. Similarly, to answer prompts such

as “Give me a reaction to produce CaCO_3 ”, there is a lot of helpful information in the Materials Project database and the internet that can help to come up with an answer.

To answer these questions, state-of-the-art computational tools or existing databases can be used. However, their use often requires expert knowledge. To use existing databases, one must choose which database to use, how to query the database, and what representation of the compound is used (e.g., international chemical identifier (InChI), SMILES, etc.). Otherwise, if the data is not in a database, one must run calculations, which requires a deep understanding of technical details. LLMs can simplify the use of such tools. By typing in a question, we can prompt the LLM to translate this question into a workflow that leads to the answer.

The MAPI-LLM team (Mayk Caldas Ramos, Sam Cox, Andrew White) made the first steps towards developing such a system (MAPI-LLM) and created a procedure to convert a text prompt into a query of the Materials Project API (API) to answer questions such as “Is the material AnByCz stable?” In addition, MAPI-LLM is capable of handling classification queries, such as “Is Fe_2O_3 magnetic?”, as well as regression problems, such as “What is the band gap of $\text{Mg}(\text{Fe}_2\text{O}_3)_2$?”.

Because an LLM is used to create the workflow, MAPI-LLM can process even more complex questions. For instance, the question “If Mn_2FeO_3 is not metallic, what is its band gap?” should create a two-step workflow first to check if the material is metallic and then calculate its band gap if it is not.

Moreover, MAPI-LLM applies ICL if the data for a material’s property is unavailable via the MAPI. MAPI-LLM generates an ICL prompt, building context based on the data for similar materials available in Materials Project database. This context is then leveraged by an LLM to infer properties for the unknown material. This innovative use of ICL bridges data gaps and enhances MAPI-LLM’s robustness and versatility.

b. sMolTalk

The previous application already touches on the problem that software for chemical applications requires scientists to invest a significant amount of time in learning even the most basic applications. An example of this is visualization software. Depending on the package and its associated documentation, chemists and materials scientists might spend hours to days learning the details of specific visualization software that is sometimes poorly documented. And in particular, for occasional use, if it takes a long time to learn the basics, it won’t be used.

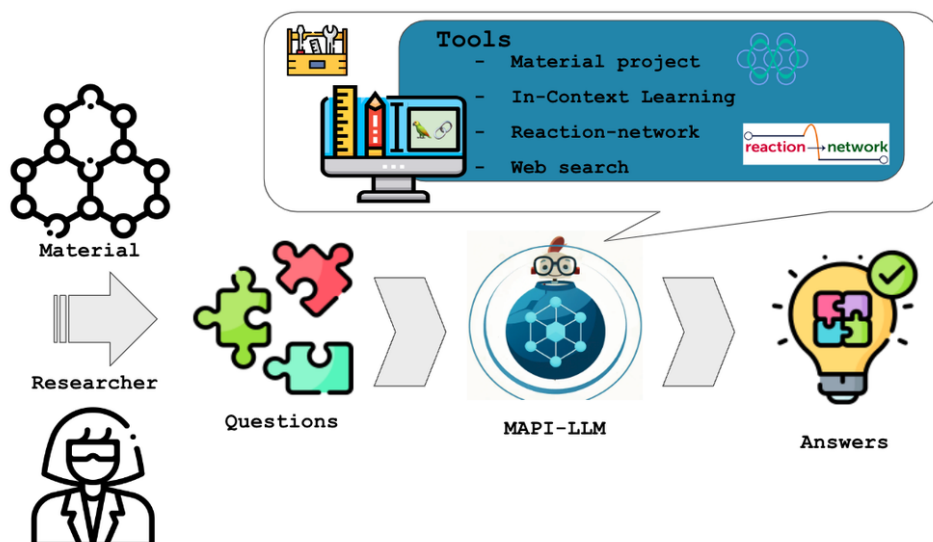


Figure 3: *Schematic overview of the MAPI-LLM workflow.* It uses LLMs to process the user’s input and decide which available tools (e.g., Materials Project API, the Reaction-Network package, and Google Search) to use following an iterative chain-of-thought procedure. In this way, it can answer questions such as “Is the material AnByCz stable?”.

As the sMolTalk-team (Jakub Lála, Sean Warren, Samuel G. Rodrigues) showed, one can use LLMs to write code for visualization tools such as `3dmol.js` to address this inefficiency [68]. Interestingly, few-shot prompting with several examples of user input with the expected JavaScript code that manipulates the `3dmol.js` viewer is all that is needed to create a prototype of an interface that can retrieve protein structures from the protein data bank (PDB) and create custom visualization solutions, e.g., to color parts of a structure in a certain way (Figure 4). The beauty of the language models is that the user can write the prompt in many different (“fuzzy”) ways: whether one writes “color” or “colour”, or terms like “light yellow” or “pale yellow” the LLM translates it into something the visualization software can interpret.

However, this application also highlights that further developments of these LLM-based tools are needed. For example, a challenge the sMolTalk tool faces is robustness. For instance, fragments from the prompt tend to leak into the output and must be handled with more involved mechanisms, such as retries in which one gives the LLMs access to the error messages or prompt engineering. Further improvement can also be expected if the application leverages a knowledge base such as the documentation of `3dmol.js`.

As the work of Glenn Hocky and Andrew White shows [69], an LLM-interface for software can also be used with other programs such as VMD [70] and extended with speech-to-text models (such as Whisper [71]) to enable voice control of such programs. In

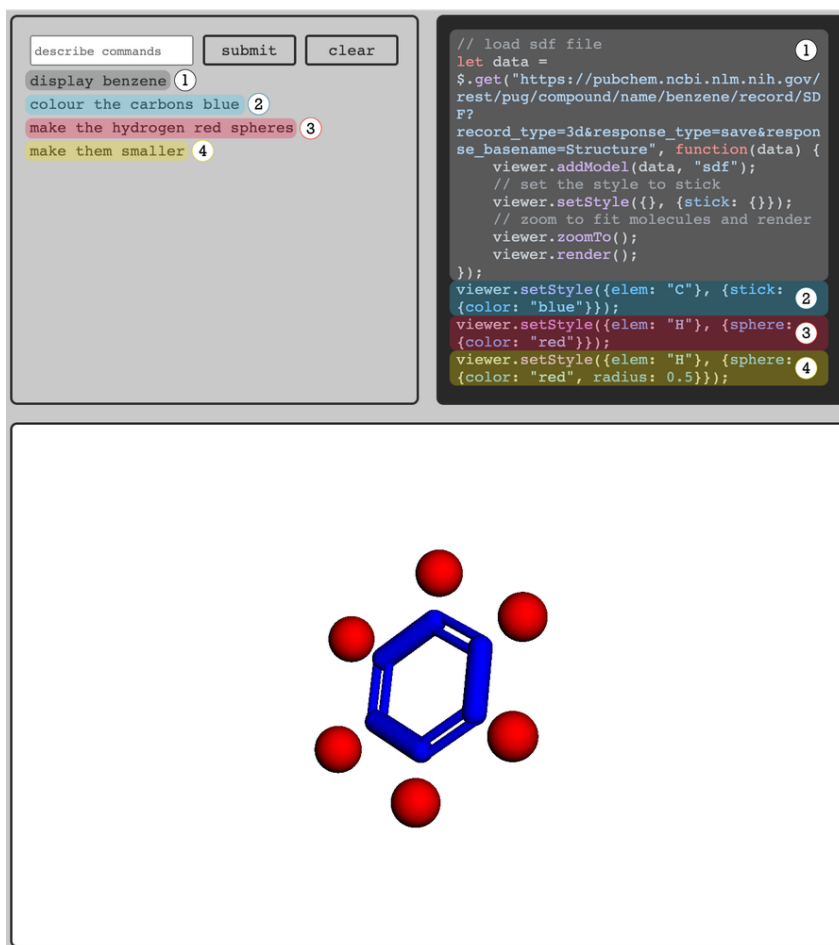


Figure 4: *The sMolTalk interface.* Based on few-shot prompting LLMs can create code for visualization tools such as `3dmol.js` that can create custom visualization based on a natural-language description of the desired output. The top left box is the input field where users can enter commands in natural language. The top right box prints the code the LLM generates. This code generates the visualization shown in the lower box. In this example, the user entered a sequence of four commands: the LLM (1) generates code for retrieving the structure, (2) colors the carbons blue, (3) displays the hydrogens as red spheres, and (4) reduces the size of the spheres.

particular, such an LLM-based agent approach might be implemented for the PyMOL program, where various tools for protein engineering could be interfaced through a chat interface, lowering the barrier to entry for biologists to use recent advancements within *in silico* protein engineering (such as RosettaFold [72] or RFDiffusion [73]).

c. *ELN interface: whinchat*




In addition to large, highly curated databases with well-defined data models [74] (such as those addressed by the MAPI-LLM project), experimental materials and chemistry data is increasingly being captured using digital tools such as ELNs and or laboratory information systems (LIMS). Importantly, these tools can be used to record both struc-

tured and unstructured lab data in a manner that is actionable by both humans and computers. However, one challenge in developing these systems is that it is difficult for a traditional user interface to have enough flexibility to capture the richness and diversity of real, interconnected, experimental data. Interestingly, LLMs can interpret and contextualize both structured and unstructured data and can therefore be used to create a novel type of flexible, conversational interface to such experimental data. The **whinchat** team (Joshua D. Bocarsly, Matthew L. Evans, and Ben E. Smith) embedded an LLM chat interface within **data_{lab}**, an open source materials chemistry data management system, where the virtual LLM-powered assistant can be “attached” to a given sample. The virtual assistant has access to responses from the JavaScript object notation (JSON) API of **data_{lab}** (containing both structured and unstructured/free text data) and can use them to perform several powerful tasks: First, it can contextualize existing data by explaining related experiments from linked responses, resolving acronyms/short-hand notations used by experimentalists, or creating concise textual summaries of complex and nested entries. Second, it can reformat or render the data, for instance, by creating (**mermaid.js**) flowcharts or (Markdown) tables (Figure 5). Third, it can use its generic reasoning abilities to suggest future experiments, for instance, related materials to study, synthesis protocols to try, or additional characterization techniques. This is shown in the examples given in SI section 2C, where **whinchat** was able to provide hints about which NMR-active nuclei can be probed in the given sample.

It is easy to envision that this tool could be even more helpful by fine-tuning or conditioning it on a research group’s knowledge base (e.g., group Wiki or standard operating procedures) and communication history (e.g., a group’s Slack history). An important limitation of the current implementation is that the small context window of available LLMs limits the amount of JSON data one can directly provide within the prompt, limiting each conversation to analyzing a relatively small number of samples. Therefore, one needs to either investigate the use of embeddings to determine which samples to include in the context or adopt an “agent” approach where the assistant is allowed to query the API of the ELN (interleaved with extraction and summarization calls).

d. BOLLaMa: facilitating Bayesian optimization with large language models

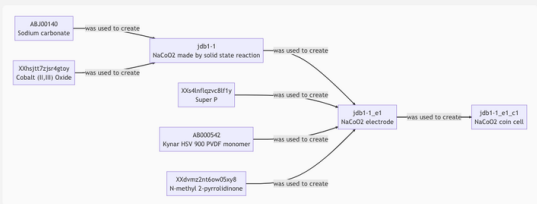
Bayesian optimization (BO) is a powerful tool for optimizing expensive functions, such as mapping of reaction conditions to the reaction yield. Chemists would greatly benefit from using this method to reduce the number of costly experiments they need

LLM Chat Block with contextual data (powered by GPT-3.5-turbo)   

Hello! I am your virtual assistant and I am here to help you manage your experimental data. What can I help you with today? raw | formatted

Please make me a mermaid flowchart showing how this cell was made from other items in the database raw | formatted

Certainly! Here is a Mermaid flowchart that shows how the cell with item_id jdb1-1_e1_c1 was made from other items in the database:



```

graph LR
    A[ABJ00140  
Sodium carbonate] -- was used to create --> B[JDB1-1  
NaCoO2 made by solid state reaction]
    C[XK0017230  
Aglyco Cobalt (II) Oxide] -- was used to create --> B
    B -- was used to create --> D[JDB1-1_e1  
NaCoO2 electrode]
    E[XK0461963  
Super P] -- was used to create --> D
    F[AB000542  
Kynar HVX 900 PFCF monomer] -- was used to create --> D
    G[XK0002060  
N-methyl-2-pyrrolidone] -- was used to create --> D
    D -- was used to create --> H[JDB1-1_e1_c1  
NaCoO2 coin cell]
  
```

In this flowchart, each rectangular box represents an item in the database, and the arrows between the boxes represent the relationships between the items. For example, SM1 and SM2 were used to create sample JDB1-1, which was then used to create sample E1, which was finally used to create the cell C1. raw | formatted

Figure 5: *Using an LLM as an interface to an ELN/data management system.* LLM-based assistants can provide powerful interfaces to digital experimental data. The figure shows a screenshot of a conversation with `whinchat` in the `datalab` data management system (<https://github.com/the-grey-group/datalab>). Here, `whinchat` is provided with data from the JSON API of `datalab` of an experimental battery cell. The user then prompts (green box) the system to build a flowchart of the provenance of the sample. The assistant responds with `mermaid.js` markdown code, which the `datalab` interface automatically recognizes and translates into a visualization.

to run [75, 76]. However, BO faces an interface and accessibility problem, too. The existing frameworks require significant background knowledge and coding experience not conventionally taught in chemistry curricula. Therefore, many chemists cannot benefit from tools such as BO. The BOLLaMa-team (Bojana Ranković, Andres M. Bran, Philippe Schwaller) showed that LLMs can lower the barrier for the use of BO by providing a natural language chat-like interface to BO algorithms. Figure 6 shows a prototype of a chat interface in which the LLM interprets the user request, initializes a BO run by suggesting initial experimental conditions, and then uses the feedback of the user to drive the BO algorithm and suggest new experiments. The example used data on various additives for a cooperative nickel-photoredox catalyzed reaction [77] and the BO code from Ranković et al. [78]. This ideally synergizes with an LLM interface to a data management solution (as discussed in the previous project) as one could directly persist the experimental results and leverage prior records to “bootstrap” BO runs.

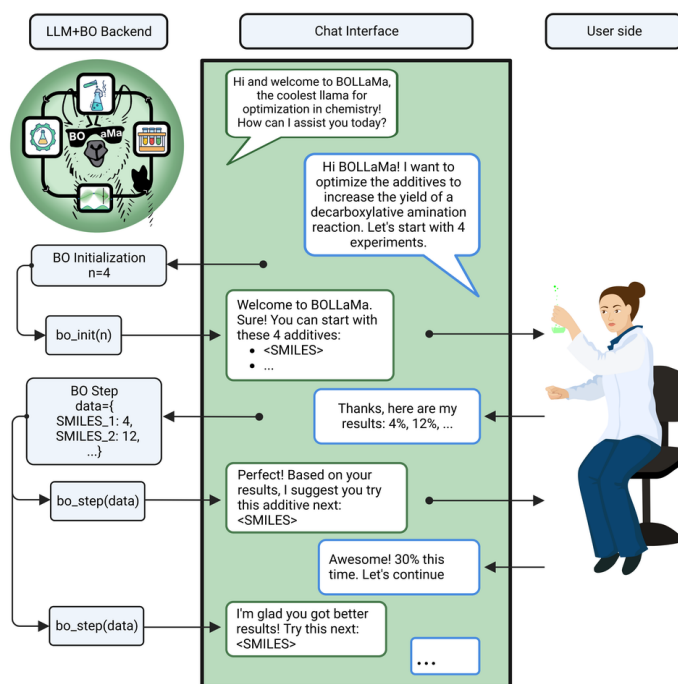


Figure 6: *Schematic overview of BoLLama.* An LLM can act as an interface to a BO algorithm. An experimental chemist can bootstrap an optimization and then, via a chat interface, update the state of the simulation to which the bot responds with the recommended next steps.

As the examples in this section show, we find that LLMs have the potential to greatly enhance the efficiency of a diverse array of processes in chemistry and materials science by providing novel interfaces to tools or by completely automating their use. This can help streamline workflows, reduce human error, and increase productivity—often by replacing “glue code” with natural language or studying a software library by chatting with an LLM.

C. Knowledge Extraction

Beyond proving novel interfaces for tools, LLMs can also serve as powerful tools for extracting knowledge from the vast amount of chemical literature available. With LLMs, researchers can rapidly mine and analyze large volumes of data, enabling them to uncover novel insights and advance the frontiers of chemical knowledge. Tools such as paperqa [28] can help to dramatically cut down the time required for literature search by automatically retrieving, summarizing, and contextualizing relevant fragments from the entire corpus of the scientific literature—for example, answering questions (with suitable citations) based on a library of hundreds of documents [35]. As the examples in the

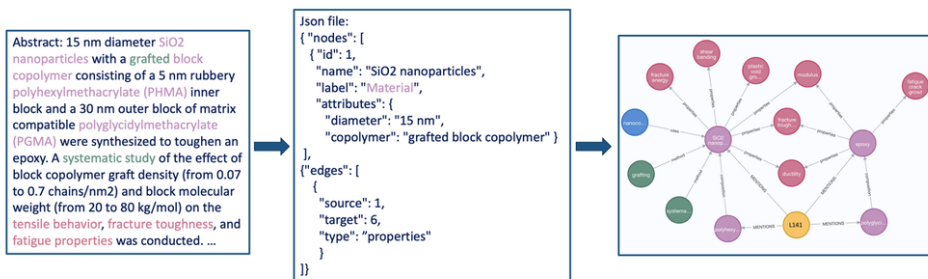


Figure 7: *The InsightGraph interface.* A suitably prompted LLM can create knowledge graph representations of scientific text that can be visualized using tools such as neo4j’s visualization tools. [81]

previous section indicated, this is particularly useful if the model is given access to search engines on the internet.

a. *InsightGraph*

To facilitate downstream use of the information, LLMs can also convert unstructured data—the typical form of these literature reports—into structured data. The use of GPT for this application has been reported by Dunn et al. [79] and Walker et al. [80], who used an iterative fine-tuning approach to extract data structured in JSON from papers. In their approach, initial (zero-shot) completions of the LLM are corrected by domain experts. Those corrected completions are then used to finetune LLMs, showing improved performance on this task.

However, for certain applications, one can construct powerful prototypes using only careful prompting. For instance, the InsightGraph team (Defne Circi, Shruti Badhwar) showed that GPT-3.5-turbo, when prompted with an example JSON containing a high-level schema and information on possible entities (e.g., materials) and pairwise relationships (e.g., properties), can, as Figure 7 illustrates, provide a knowledge graph representation of the entities and their relationships in a text describing the properties and composition of polymer nanocomposites. A further optimized version of this tool might offer a concise and visual means to quickly understand and compare material types and uses across sets of articles and could be used to launch a literature review. An advanced potential application is the creation of structured, materials-specific datasets for fact-based question-answering and downstream machine-learning tasks.

b. *Extracting Structured Data from Free-form Organic Synthesis Text*

Unstructured text is commonly used for describing organic synthesis procedures. Due to the large corpus of literature, manual conversion from unstructured text to struc-

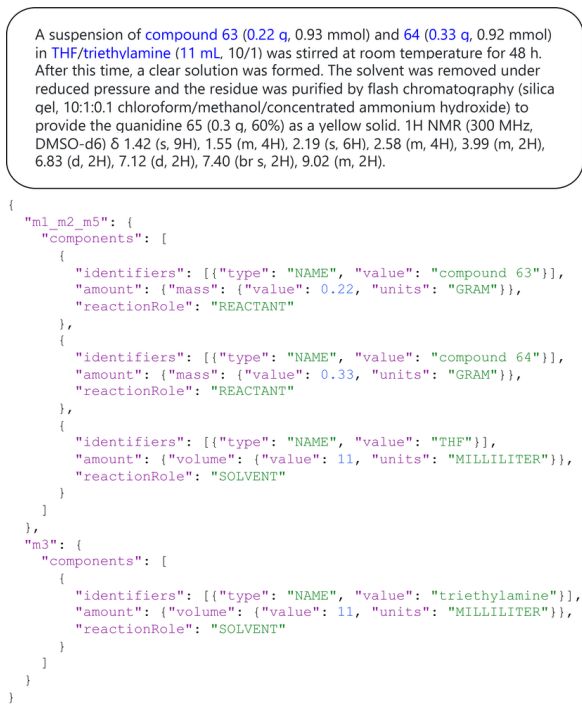


Figure 8: *The Organic Synthesis Parser interface.* The top box shows text describing an organic reaction (<https://open-reaction-database.org/client/id/ord-1f99b308e17340cb8e0e3080c270fd08>), which the finetuned LLM converts into structured JSON (bottom). A demo application can be found at https://qai222.github.io/LLM_organic_synthesis/.

tured data is unrealistic. However, structured data are needed for building conventional ML models for reaction prediction and condition recommendation. The Open Reaction Database (ORD) [82] is a database of curated organic reactions. In the ORD, while reaction data are structured by the ORD schema, many of their procedures are also available as plain text. Interestingly, an LLM (e.g., OpenAI’s `text-davinci-003`) can, after fine-tuning on only 300 prompt-completion pairs, extract 93% of the components from the free-text reaction description into valid JSONs (Figure 8). Such models might significantly increase the data available for training models on tasks such as predicting reaction conditions and yields. It is worth noting that all reaction data submitted to ORD are made available under the CC-BY-SA license, which makes ORD a suitable data source for fine-tuning or training an LLM to extract structured data from organic procedures. A recent study on gold nanorod growth procedures also demonstrated the ability of LLM in a similar task. [80] In contrast to the LIFT-based prediction of atomization energies reported in the first section by the Berkeley-Madison team, parameter-efficient fine-tuning of the open-source Alpaca model [83–85] using LoRA [47] did not yield a model that can construct valid JSONs.

c. TableToJson: Structured information from tables in scientific papers

The previous example shows how structured data can be extracted from plain text using LLMs. However, relevant information in the scientific literature is not only found in text form. Research papers often contain tables that collect data on material properties, synthesis conditions, and results of characterization and experiments. Converting table information into structured formats is essential to enable automated data analysis, extraction, and integration into computational workflows. Although some techniques could help in the process of extracting this information (performing OCR or parsing XML), converting this information in structured data following, for example, a specific JSON schema with models remains a challenge. The INCAR-CSIC team showed that the OpenAI `text-davinci-003` model, when prompted with a desired JSON schema and the HyperText Markup Language (HTML) of a table contained in a scientific paper, can generate structured JSON with the data in the table.

First, the OpenAI `text-davinci-003` model was directly used to generate JSON objects from the table information. This approach was applied to several examples using tables collected from papers on different research topics within the field of chemistry [86–92]. The accuracy for those different examples, calculated as the percentage of schema values generated correctly, is shown in Figure 9. When the OpenAI model was prompted with the table and desired schema to generate a JSON object, it worked remarkably well in extracting the information from each table cell and inserting it at the expected place in the schema. As output, it provided a valid JSON object with a 100 % success rate of error-free generated values in all the studied examples. However, in some examples, the model did not follow the schema.

To potentially address this problem the team utilized the `jsonformer` approach. This tool reads the keys from the JSON schema and only generates the value tokens, guaranteeing the generation of a syntactically valid JSON (corresponding to the desired schema) by the LLM [93, 94]. Using an LLM without such a decoding strategy cannot guarantee that valid JSON outputs are produced. With the `jsonformer` approach, in most cases, by using a simple descriptive prompt about the type of input text, structured data can be obtained with 100 % correctness of the generated values. In one example, an accuracy of 80 % was obtained due to errors in the generation of numbers in scientific notation. For a table with more complex content (long molecule names, hyphens, power numbers, subscripts, and superscripts, . . .) the team achieved an accuracy of only 46 %. Most of

these issues could be solved by adding a specific explanation in the prompt, increasing the accuracy to 100% in most cases.

Overall, both approaches performed well in generating the JSON format. The OpenAI `text-davinci-003` model could correctly extract structured information from tables and give a valid JSON output, but it cannot guarantee that the outputs will always follow the provided schema. `Jsonformer` may present problems when special characters need to be generated, but most of these issues could be solved with careful prompting. These results show that LLMs can be a useful tool to help to extract scientific information in tables and convert it into a structured form with a fixed schema that can be stored in a database, which could encourage the creation of more topic-specific databases of research results.

	text-davinci-003 (schema in prompt)		text-davinci-003 + jsonformer	
	<pre>prompt = "Generate a JSON object extracting the information from this table in html code: " + HTML_table + "Generate the JSON result with the following JSON schema and give only the JSON as output: " + JSON_schema</pre>			
	JSON follows schema	% values extracted ok	JSON follows schema	% values extracted ok
carbon materials for CO ₂ adsorption	✓	100%	✓	100%
MOFs properties	✓	100%	✓	100%
supercapacitor performance	✓	100%	✓	100%
catalysts for CO ₂ conversion	✗ → ✓	100%	✓	94%
biomass properties	✓	100%	✓	100%
anode materials for SOFCs	✗ → ✓	100%	✓	80% → 100%
perovskite cathodes for SOFCs	✓	100%	✓	46% → 60% → 86-100%

Figure 9: *TableToJson*. Results of the structured JSON generation of tables contained in scientific articles. Two approaches are compared: (i) the use of an OpenAI model prompted with the desired JSON schema, and (ii) the use of an OpenAI model together with `jsonformer`. In both cases, JSON objects were always obtained. The output of the OpenAI model did not always follow the provided schema, although this might be solved by modifying the schema. The accuracy of the results from the `jsonformer` approach used with OpenAI models could be increased (as shown by the blue arrows) by solving errors in the generation of power numbers and special characters with a more detailed prompt. The results can be visualized in this demo app: <https://vgvinter-tabletojson-app-kt5aiv.streamlit.app/>

d. *AbstractToTitle & TitleToAbstract: text summarization and text generation*

Technical writing is a challenging task that often requires presenting complex abstract ideas in limited space. For this, frequent rewrites of sections are needed, in which LLMs could assist domain experts. Still, evaluating their ability to generate text such as a scientific paper is essential, especially for chemistry and materials science applications.

Large datasets of chemistry-related text are available from open-access platforms such as arXiv and PubChem. These articles contain titles, abstracts, and often complete manuscripts, which can be a testbed for evaluating LLMs as these titles and abstracts are usually written by expert researchers. Ideally, an LLM should be able to generate a title of an abstract close to the one developed by the expert, which can be considered a specialized text-summarization task. Similarly, given a title, an LLM should generate text close to the original abstract of the article, which can be considered a specialized text-generation task.

These tasks have been introduced by the AbstractToTitle & TitleToAbstract team (Kamal Choudhary) in the JARVIS-ChemNLP package [95]. For text summarization, it uses a pre-trained Text-to-Text Transfer Transformer (T5) model developed by Google [96] that is further fine-tuned to produce summaries of abstracts. On the arXiv condensed-matter physics (cond-mat) data, the team found that fine-tuning the model can help improve the performance (Recall-Oriented Understudy for Gisting Evaluation (ROUGE)-1 score of 39.0% which is better than an untrained model score of 30.8% for an 80/20 split).

For text generation, JARVIS-ChemNLP finetunes the pretrained GPT-2-medium [48] model available in the HuggingFace library. [97] After finetuning, the team found a ROUGE score of 31.7%, which is a good starting point for pre-suggestion text applications. Both tasks with well-defined train and test splits are now available in the JARVIS-Leaderboard platform for the AI community to compare other LLMs and systematically improve the performance.

In the future, such title to abstract capabilities can be extended to generating full-length drafts with appropriate tables, multi-modal figures, and results as an initial start for the human researcher to help in the technical writing processes. Note that there have been recent developments in providing guidelines for using LLM-generated text in technical manuscripts [98], so such an LLM model should be considered as an assistant of writing and not the master/author of the manuscripts.

D. Education

Given all the opportunities LLM open for materials science and chemistry, there is an urgent need for education to adapt. Interestingly, LLMs also provide us with entirely

A

Video link

<https://www.youtube.com/watch?v=aKLOIF7wxsQ>

Path to audio

Go

Keywords:

1. Monte Carlo simulation
2. Metropolis algorithm
3. Importance sampling

Summary:

The speaker, Beard Smith, is continuing a lecture on Monte Carlo simulation, focusing on the details of the algorithm. The audience is introduced to a basic form of the metropolis algorithm, in which particles in a box are randomly displaced and their energy is calculated before acceptance. The speaker raises several questions about the algorithm's effectiveness and implementation, including the selection of particles at random, the order of displacement, and the size of the displacement. The talk will delve into these questions and provide answers to them.

Questions:

1. Can you explain the concept of importance sampling in Monte Carlo simulation?
2. How do you calculate the energy of the system in the new position of the particle?
3. Why is it important to space out the sampling of thermodynamic properties during Monte Carlo simulation?

B

Questions:

1. What inspired you to pursue research in the field of carbon capture?
2. Can you discuss the main challenges faced by scientists in the field of carbon capture?
3. How do you think the general public can become more aware of the importance of carbon capture and reducing CO₂ emissions?
4. Are there any potential negative impacts or drawbacks of using carbon capture technology?
5. Can you discuss any current or ongoing projects related to carbon capture that you are involved in?
6. How do you see the field of carbon capture evolving in the next few years, and what role do you see computational modeling playing in that evolution?

Chemical compounds:

[copper](#), [co₂](#), [carbon](#), [carbon dioxide](#), [zinc](#)

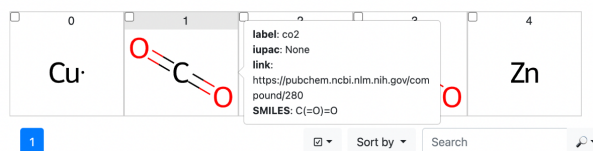


Figure 10: *The I-digest interface*. A video (e.g., of a lecture recording) can be described using the Whisper model. Based on the transcript, an LLM can generate questions (and answers). Those can assist students in their learning. The LLM can also detect mentions of chemicals and link to further information about them (e.g., on PubChem [100–102]).

novel educational opportunities [99], for example, by personalizing content or providing almost limitless varied examples.

The I-Digest (Information-Digestor) hackathon team (Beatriz Mouriño, Elias Moubarak, Joren Van Herck, Sauradeep Majumdar, Xiaoqi Zhang) created a path toward such a new educational opportunity by providing students with a digital tutor based on course material such as lecture recordings. Using the Whisper model [71], videos of lecture recordings can be transcribed to text transcripts. The transcripts can then be fed into an LLM with the prompt to come up with questions about the content presented in the video (Figure 10). In the future, these questions might be shown to students before a video starts, allowing them to skip parts they already know or after the video, guiding students to the relevant timestamps or additional material in case of an incorrect answer.

Importantly, and in contrast to conventional educational materials, this approach can generate a practically infinite number of questions and could, in the future, be continuously be improved by student feedback. In addition, it is easy to envision extending this approach to consider lecture notes or books to guide the students further or even recommend specific exercises.

II. CONCLUSION

The fact that the groups were able to present prototypes that could do quite complex tasks in such a short time illustrates the power of LLMs. Some of these prototypes would have taken many months of programming just a few months ago, but the fact that LLMs could reduce this time to a few hours is one of the primary reasons for the success of our hackathon. Combined with the time-constrained environment in teams (with practically zero cost of “failure”), we found more energy and motivation. The teams delivered more results than in most other hackathons we participated in.

Through the LIFT framework, one can use LLMs to address problems that could already be addressed with conventional approaches—but in a much more accessible way (using the same approach for different problems), while also reusing established concepts such as Δ -ML. At the same time, however, we can use LLMs to model chemistry and materials science in novel ways; for example, by incorporating context information such as “fuzzy” design rules or directly operating on unstructured data. Overall, a common use case has been to use LLMs to deal with “fuzziness” in programming and tool development. We can already see tools like Copilot and ChatGPT being used to convert “fuzzy abstractions” or hard-to-define tasks into code. These advancements may soon allow everyone to write small apps or customize them to their needs (end-user programming). Additionally, we can observe an interesting trend in tool development: most of the logic in the showcased tools is written in English, not in Python or another programming language. The resulting code is shorter, easier to understand, and has fewer dependencies because LLMs are adept at handling fuzziness that is difficult to address with conventional code. This suggests that we may not need more formats or standards for interoperability; instead, we can simply describe existing solutions in natural language to make them interoperable. Exploring this avenue further is exciting, but it is equally important to recognize the limitations of LLMs, as they currently have limited interpretability and lack robustness.

It is interesting to note that none of the projects relied on the knowledge or understanding of chemistry by LLMs. Instead, they relied on general reasoning abilities and provided chemistry information through the context or fine-tuning. However, this also brings new and unique challenges. All projects used the models provided by OpenAI’s API. While these models are powerful, we cannot examine how they were built or have any guarantee of continued reliable access to them.

Although there are open-source language models and techniques available, they are generally more difficult to use compared to simply using OpenAI’s API. Furthermore, the performance of language models can be fragile, especially for zero- or few-shot applications. To further investigate this, new benchmarks are needed that go beyond the tabular datasets we have been using for ML for molecular and materials science—we simply have no frameworks to compare and evaluate predictive models that use context, unstructured data, or tools. Without automated tests, however, it is difficult to improve these systems systematically. On top of that, consistent benchmarking is hard because de-duplication is ill-defined even if the training data are known. To enable a scientific approach to the development and analysis of these systems, we will also need to revisit versioning frameworks to ensure reproducibility as systems that use external tools depend on the exact versions of training data, LLM, as well as of the external tools and prompting setup.

The diversity of the prototypes presented in this work shows that the potential applications are almost unlimited, and we can probably only see the tip of the iceberg—for instance, we didn’t even touch modalities other than text thus far.

Given these new ways of working and thinking, combined with the rapid pace of developments in the field, we believe that we urgently need to rethink how we work and teach. We must discuss how we ensure safe use [103], standards for evaluating and sharing those models, and robust and reliable deployments. But we also need to discuss how we ensure that the next generation of chemists and materials scientists are proficient and critical users of these tools—that can use them to work more efficiently while critically reflecting on the outputs of the systems. We believe that to truly leverage the power of LLMs in the molecular and material sciences, we need a community effort—including not only chemists and computer scientists but also lawyers, philosophers, and ethicists: the possibilities and challenges are too broad and profound to tackle alone.

Acknowledgements

We would like to specifically thank Jim Warren (NIST) for his contributions to discussions leading up to the hackathon and his participation as a judge during the event. We would also like to thank Anthony Costa and Christian Dallago (NVIDIA) for supporting the hackathon.

B.B., I.T.F, and ZH acknowledge support from the the National Science Foundation awards #2226419 and #2209892. This work was performed under the following financial

assistance award 70NANB19H005 from the U.S. Department of Commerce, National Institute of Standards and Technology as part of the Center for Hierarchical Materials Design (CHiMaD).

K.J.S, A.S. acknowledge support from the the National Science Foundation award #1931306.

K.M.J., S.M., J.v.H., X.Z., B.M., E.M., and B.S. were supported by the MARVEL National Centre for Competence in Research funded by the Swiss National Science Foundation (grant agreement ID 51NF40-182892) and the USorb-DAC Project, which is funded by a grant from The Grantham Foundation for the Protection of the Environment to RMI’s climate tech accelerator program, Third Derivative. **B.M.** was further supported by the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 945363.

M.C.R., S.C., and A.D.W. were supported by the National Science Foundation and the National Institute of General Medical Sciences under Grant No. 1764415 and award number R35GM137966, respectively.

Q.A.’s contribution to this work was supported by the National Center for Advancing Translational Sciences of the National Institutes of Health under award number U18TR004149. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

M.V.G. acknowledges support from the Spanish National Research Council (CSIC) through the Programme for internationalization i-LINK 2021 (Project LINKA20412), and from the Spanish Agencia Estatal de Investigación (AEI) through the Grant TED2021-131693B-I00 funded by MCIN/AEI/ 10.13039/501100011033 and by the “European Union NextGenerationEU/PRTR” and through the Ramón y Cajal Grant RYC-2017-21937 funded by MCIN/AEI/ 10.13039/501100011033 and by “ESF Investing in your future”.

The `dataLab` project (**M.L.E., B.E.S.** and **J.D.B.**) has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement 957189 (DOI: 10.3030/957189), the Battery Interface Genome - Materials Acceleration Platform (BIG-MAP), as an external stakeholder project. **M.L.E.** additionally thanks the BEWARE scheme of the Wallonia-Brussels Federation for funding under the European Commission’s Marie Curie-Skłodowska Action (COFUND 847587). **B.E.S.** acknowledges support from the UK’s Engineering and Physical Sciences Research Council (ESPRC).

B.P. acknowledges support from the National Science Foundation through NSF-CBET Grant No. 1917340. The authors thank Phung Cheng Fei, Hassan Harb, and Vinayak Bhat for their helpful comments on this project.

D.C. and L.C.B. thank NSF DGE-2022040 for the aiM NRT funding support.

K.C. thank the National Institute of Standards and Technology for funding, computational, and data-management resources. Please note certain equipment, instruments, software, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement of any product or service by NIST, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

A.K.G., G.W.M., A.I., and W.A.d.J. were supported by the U.S. Department of Energy, Office of Science, Basic Energy Sciences, Materials Sciences and Engineering Division under Contract No. DE-AC02-05CH11231, FWP No. DAC-LBL-Long, and by the U.S. Department of Energy, Office of Science, Office of High Energy Physics under Award Number DE-FOA-0002705.

M.B, B.R., and P.S. were supported by the NCCR Catalysis (grant number 180544), a National Centre of Competence in Research funded by the Swiss National Science Foundation.

S.G.R. and J.L. acknowledge the generous support of Eric and Wendy Schmidt, and the core funding of the Francis Crick Institute, which receives its funding from Cancer Research UK, the UK Medical Research Council, and the Wellcome Trust.

-
- [1] Butler, K. T.; Davies, D. W.; Cartwright, H.; Isayev, O.; Walsh, A. Machine learning for molecular and materials science. *Nature* **2018**, *559*, 547–555.
 - [2] Moosavi, S. M.; Jablonka, K. M.; Smit, B. The Role of Machine Learning in the Understanding and Design of Materials. *J. Am. Chem. Soc.* **2020**, *142*, 20273–20287.
 - [3] Morgan, D.; Jacobs, R. Opportunities and Challenges for Machine Learning in Materials Science. *Annu. Rev. Mater. Res.* **2020**, *50*, 71–103.
 - [4] Ramprasad, R.; Batra, R.; Pilania, G.; Mannodi-Kanakkithodi, A.; Kim, C. Machine learning in materials informatics: recent applications and prospects. *npj Comput. Mater.* **2017**, *3*, 54.
 - [5] Schmidt, J.; Marques, M. R. G.; Botti, S.; Marques, M. A. L. Recent advances and

- applications of machine learning in solid-state materials science. *npj Comput. Mater.* **2019**, *5*.
- [6] Choudhary, K.; DeCost, B.; Chen, C.; Jain, A.; Tavazza, F.; Cohn, R.; Park, C. W.; Choudhary, A.; Agrawal, A.; Billinge, S. J., et al. Recent advances and applications of deep learning methods in materials science. *npj Comput. Mater.* **2022**, *8*, 59.
- [7] Jablonka, K. M.; Ongari, D.; Moosavi, S. M.; Smit, B. Big-Data Science in Porous Materials: Materials Genomics and Machine Learning. *Chem. Rev.* **2020**, *120*, 8066–8129.
- [8] Shi, J.; Quevillon, M. J.; Amorim Valença, P. H.; Whitmer, J. K. Predicting Adhesive Free Energies of Polymer–Surface Interactions with Machine Learning. *ACS Appl. Mater. Interfaces* **2022**, *14*, 37161–37169.
- [9] Shi, J.; Albreiki, F.; Colón, Y. J.; Srivastava, S.; Whitmer, J. K. Transfer Learning Facilitates the Prediction of Polymer–Surface Adhesion Strength. *J. Chem. Theory Comput.* **2023**,
- [10] Noé, F.; Tkatchenko, A.; Müller, K.-R.; Clementi, C. Machine Learning for Molecular Simulation. *Annu. Rev. Phys. Chem.* **2020**, *71*, 361–390.
- [11] Batzner, S.; Musaelian, A.; Sun, L.; Geiger, M.; Mailoa, J. P.; Kornbluth, M.; Molinari, N.; Smidt, T. E.; Kozinsky, B. E(3)equivariant graph neural networks for data efficient and accurate interatomic potentials. *Nat. Commun.* **2022**, *13*, 2453.
- [12] Sanchez-Lengeling, B.; Aspuru-Guzik, A. Inverse molecular design using machine learning: Generative models for matter engineering. *Science* **2018**, *361*, 360–365.
- [13] Gonthier, J. F.; Steinmann, S. N.; Wodrich, M. D.; Corminboeuf, C. Quantification of “fuzzy” chemical concepts: a computational perspective. *Chem. Soc. Rev.* **2012**, *41*, 4671.
- [14] Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31–36.
- [15] Jablonka, K. M.; Patiny, L.; Smit, B. Making the collective knowledge of chemistry open and machine actionable. *Nat. Chem.* **2022**, *14*, 365–376.
- [16] Bommasani, R. et al. On the Opportunities and Risks of Foundation Models. *CoRR* **2021**, *abs/2108.07258*.
- [17] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Advances in neural information processing systems* **2017**, *30*.
- [18] Chowdhery, A. et al. PaLM: Scaling Language Modeling with Pathways. arXiv preprint:

- Arxiv-2204.02311. 2022.
- [19] Hoffmann, J. et al. Training Compute-Optimal Large Language Models. arXiv preprint: Arxiv-2203.15556. 2022.
- [20] Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems* **2020**, *33*, 1877–1901.
- [21] Edwards, C. N.; Lai, T.; Ros, K.; Honke, G.; Ji, H. Translation between Molecules and Natural Language. *Conference On Empirical Methods In Natural Language Processing* **2022**,
- [22] Eloundou, T.; Manning, S.; Mishkin, P.; Rock, D. GPTs are GPTs: An Early Look at the Labor Market Impact Potential of Large Language Models. *arXiv preprint: Arxiv-2303.10130* **2023**,
- [23] Srivastava, A. et al. Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models. arXiv preprint: Arxiv-2206.04615. 2022.
- [24] Bubeck, S.; Chandrasekaran, V.; Eldan, R.; Gehrke, J.; Horvitz, E.; Kamar, E.; Lee, P.; Lee, Y. T.; Li, Y.; Lundberg, S.; Nori, H.; Palangi, H.; Ribeiro, M. T.; Zhang, Y. Sparks of Artificial General Intelligence: Early experiments with GPT-4. *arXiv preprint: Arxiv-2303.12712* **2023**,
- [25] Schick, T.; Dwivedi-Yu, J.; Dessì, R.; Raileanu, R.; Lomeli, M.; Zettlemoyer, L.; Cancedda, N.; Scialom, T. Toolformer: Language Models Can Teach Themselves to Use Tools. *arXiv preprint: Arxiv-2302.04761* **2023**,
- [26] Karpas, E. et al. MRKL Systems: A modular, neuro-symbolic architecture that combines large language models, external knowledge sources and discrete reasoning. *arXiv preprint: Arxiv-2205.00445* **2022**,
- [27] Shen, Y.; Song, K.; Tan, X.; Li, D.; Lu, W.; Zhuang, Y. HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in HuggingFace. arXiv preprint: Arxiv-2303.17580. 2023.
- [28] White, A. paper-qa. <https://github.com/whitead/paper-qa>, 2022.
- [29] Liu, J. LlamaIndex. 2022; https://github.com/jerryjliu/llama_index, last accessed 2023-05-30.
- [30] Andrej Karpathy [@karpathy], The Hottest New Programming Language Is English. 2023; <https://twitter.com/karpathy/status/1617979122625712128>, last accessed 2023-05-11.

- [31] Hocky, G. M.; White, A. D. Natural language processing models that automate programming will transform chemistry research and teaching. *Digital Discovery* **2022**, *1*, 79–83.
- [32] Jablonka, K. M.; Schwaller, P.; Ortega-Guerrero, A.; Smit, B. Is GPT-3 all you need for low-data discovery in chemistry? *ChemRxiv preprint 10.26434/chemrxiv-2023-fw8n4* **2023**,
- [33] White, A. D.; Hocky, G. M.; Gandhi, H. A.; Ansari, M.; Cox, S.; Wellawatte, G. P.; Sasmal, S.; Yang, Z.; Liu, K.; Singh, Y., et al. Assessment of chemistry knowledge in large language models that generate code. *Digital Discovery* **2023**,
- [34] Ramos, M. C.; Michtavy, S. S.; Porosoff, M. D.; White, A. D. Bayesian Optimization of Catalysts With In-context Learning. *arXiv preprint: Arxiv-2304.05341* **2023**,
- [35] White, A. D. The future of chemistry is language. *Nat. Rev. Chem.* **2023**,
- [36] Ramakrishnan, R.; Dral, P. O.; Rupp, M.; Von Lilienfeld, O. A. Big data meets quantum chemistry approximations: the Δ -machine learning approach. *J. Chem. Theory Comput.* **2015**, *11*, 2087–2096.
- [37] Dinh, T.; Zeng, Y.; Zhang, R.; Lin, Z.; Gira, M.; Rajput, S.; Sohn, J.-Y.; Papailiopoulos, D.; Lee, K. LIFT: Language-Interfaced Fine-Tuning for Non-Language Machine Learning Tasks. *arXiv preprint: Arxiv-2206.06565*. 2022.
- [38] Krenn, M.; Häse, F.; Nigam, A.; Friederich, P.; Aspuru-Guzik, A. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Mach. Learn.: Sci. Technol.* **2020**, *1*, 045024.
- [39] Krenn, M.; Ai, Q.; Barthel, S.; Carson, N.; Frei, A.; Frey, N. C.; Friederich, P.; Gaudin, T.; Gayle, A. A.; Jablonka, K. M., et al. SELFIES and the future of molecular string representations. *Patterns* **2022**, *3*, 100588.
- [40] Curtiss, L. A.; Redfern, P. C.; Raghavachari, K. Gaussian-4 theory using reduced order perturbation theory. *J. Chem. Phys.* **2007**, *127*, 124105.
- [41] Ramakrishnan, R.; Dral, P. O.; Rupp, M.; Von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Sci. Data* **2014**, *1*, 1–7.
- [42] Narayanan, B.; Redfern, P. C.; Assary, R. S.; Curtiss, L. A. Accurate quantum chemical energies for 133000 organic molecules. *Chem. Sci.* **2019**, *10*, 7449–7455.
- [43] Gupta, A. K.; Raghavachari, K. Three-Dimensional Convolutional Neural Networks Utilizing Molecular Topological Features for Accurate Atomization Energy Predictions. *J. Chem. Theory Comput.* **2022**, *18*, 2132–2143.

- [44] Ward, L.; Blaiszik, B.; Foster, I.; Assary, R. S.; Narayanan, B.; Curtiss, L. Machine learning prediction of accurate atomization energies of organic molecules from low-fidelity quantum chemical calculations. *MRS Commun.* **2019**, *9*, 891–899.
- [45] Ramakrishnan, R.; Dral, P. O.; Rupp, M.; von Lilienfeld, O. A. Big Data Meets Quantum Chemistry Approximations: The Δ -Machine Learning Approach. *J. Chem. Theory Comput.* **2015**, *11*, 2087–2096.
- [46] Becke, A. D. Density-functional thermochemistry. III. The role of exact exchange. *J. Chem. Phys.* **1993**, *98*, 5648–5652.
- [47] Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv preprint: Arxiv-2106.09685* **2021**,
- [48] Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language Models are Unsupervised Multitask Learners. **2019**, https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.
- [49] Scrivener, K. L.; John, V. M.; Gartner, E. M. Eco-efficient cements: Potential economically viable solutions for a low-CO₂ cement-based materials industry. *Cem. Concr. Res.* **2018**, *114*, 2–26.
- [50] Völker, C.; Benjami Moreno Torres,; Tehseen Rug,; Firdous, R.; Ghezal Ahmad,; Zia, J.; Lüders, S.; Scaffino, H. L.; Höpler, M.; Böhmer, F.; Pfaff, M.; Stephan, D.; Kruschwitz, S. Green building materials: a new frontier in data-driven sustainable concrete design. Preprint 10.13140/RG.2.2.29079.85925. 2023.
- [51] Rao, G. M.; Rao, T. D. G. A quantitative method of approach in designing the mix proportions of fly ash and GGBS-based geopolymer concrete. *Aust. J. Civ. Eng.* **2018**, *16*, 53–63.
- [52] Tshitoyan, V.; Dagdelen, J.; Weston, L.; Dunn, A.; Rong, Z.; Kononova, O.; Persson, K. A.; Ceder, G.; Jain, A. Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature* **2019**, *571*, 95–98.
- [53] Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. International Conference On Learning Representations. 2013.
- [54] Olivetti, E. A.; Cole, J. M.; Kim, E.; Kononova, O.; Ceder, G.; Han, T. Y.-J.; Hiszpanski, A. M. Data-driven materials research enabled by natural language processing and


















































- information extraction. *Appl. Phys. Rev.* **2020**, *7*, 041317.
- [55] Selva Birunda, S.; Kanniga Devi, R. A review on word embedding techniques for text classification. *Innovative Data Communication Technologies and Application: Proceedings of ICIDCA 2020* **2021**, 267–281.
- [56] Hong, Z.; Ajith, A.; Pauloski, G.; Duede, E.; Malamud, C.; Magoulas, R.; Chard, K.; Foster, I. ScholarBERT: Bigger is Not Always Better. arXiv preprint: Arxiv-2205.11342. 2022.
- [57] Dai, H. et al. AugGPT: Leveraging ChatGPT for Text Data Augmentation. arXiv preprint: Arxiv-2302.13007. 2023.
- [58] Venkatasubramanian, V.; Chan, K.; Caruthers, J. M. Computer-aided molecular design using genetic algorithms. *Comput. Chem. Eng.* **1994**, *18*, 833–844.
- [59] Flam-Shepherd, D.; Aspuru-Guzik, A. Language models can generate molecules, materials, and protein binding sites directly in three dimensions as XYZ, CIF, and PDB files. arXiv preprint: Arxiv-2305.05708. 2023.
- [60] Taylor, R.; Kardas, M.; Cucurull, G.; Scialom, T.; Hartshorn, A.; Saravia, E.; Poulton, A.; Kerkez, V.; Stojnic, R. Galactica: A Large Language Model for Science. *arXiv preprint: Arxiv-2211.09085* **2022**,
- [61] 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. *Chem. Sci.* **2018**, *9*, 6091–6098.
- [62] Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; Cao, Y. ReAct: Synergizing Reasoning and Acting in Language Models. *arXiv preprint: Arxiv-2210.03629* **2023**,
- [63] Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Chi, E.; Xia, F.; Le, Q.; Zhou, D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Neural Information Processing Systems* **2022**,
- [64] OpenAI, GPT-4 Technical Report. arXiv preprint: Arxiv-2303.08774v3. 2023.
- [65] Bran, A. M.; Cox, S.; White, A. D.; Schwaller, P. ChemCrow: Augmenting large-language models with chemistry tools. *arXiv preprint: Arxiv-2304.05376* **2023**,
- [66] Boiko, D. A.; MacKnight, R.; Gomes, G. Emergent autonomous scientific research capabilities of large language models. *arXiv preprint: Arxiv-2304.05332* **2023**,
- [67] Jain, A.; Ong, S. P.; Hautier, G.; Chen, W.; Richards, W. D.; Dacek, S.; Cholia, S.; Gunter, D.; Skinner, D.; Ceder, G.; Persson, K. A. Commentary: The Materials Project:

- A materials genome approach to accelerating materials innovation. *APL Materials* **2013**, *1*, 011002.
- [68] Rego, N.; Koes, D. 3Dmol.js: molecular visualization with WebGL. *Bioinformatics* **2014**, *31*, 1322–1324.
- [69] White, A.; Hocky, G. marvis - VMD Audio/Text control with natural language. <https://github.com/whitead/marvis>, 2022.
- [70] Humphrey, W.; Dalke, A.; Schulten, K. VMD: Visual molecular dynamics. *J. Mol. Graph.* **1996**, *14*, 33–38.
- [71] Radford, A.; Kim, J. W.; Xu, T.; Brockman, G.; McLeavey, C.; Sutskever, I. Robust speech recognition via large-scale weak supervision. arXiv preprint: ArXiv-2212.04356. 2022.
- [72] Baek, M. et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science* **2021**, *373*, 871–876.
- [73] Watson, J. L. et al. Broadly applicable and accurate protein design by integrating structure prediction networks and diffusion generative models. bioRxiv preprint: 10.1101/2022.12.09.519842. 2022.
- [74] Andersen, C. W. et al. OPTIMADE, an API for exchanging materials data. *Sci. Data* **2021**, *8*, 217.
- [75] Volk, A. A.; Epps, R. W.; Yonemoto, D. T.; Masters, B. S.; Castellano, F. N.; Reyes, K. G.; Abolhasani, M. AlphaFlow: autonomous discovery and optimization of multi-step chemistry using a self-driven fluidic lab guided by reinforcement learning. *Nat. Commun.* **2023**, *14*, 1403.
- [76] Shields, B. J.; Stevens, J.; Li, J.; Parasram, M.; Damani, F.; Alvarado, J. I. M.; Janey, J. M.; Adams, R. P.; Doyle, A. G. Bayesian reaction optimization as a tool for chemical synthesis. *Nature* **2021**, *590*, 89–96.
- [77] Prieto Kullmer, C. N.; Kautzky, J. A.; Krska, S. W.; Nowak, T.; Dreher, S. D.; MacMillan, D. W. Accelerating reaction generality and mechanistic insight through additive mapping. *Science* **2022**, *376*, 532–539.
- [78] Ranković, B.; Griffiths, R.-R.; Moss, H. B.; Schwaller, P. Bayesian optimisation for additive screening and yield improvements in chemical reactions – beyond one-hot encodings. ChemRxiv preprint 10.26434/chemrxiv-2022-nll2j. 2022.
- [79] Dunn, A.; Dagdelen, J.; Walker, N.; Lee, S.; Rosen, A. S.; Ceder, G.; Persson, K. A.;

- Jain, A. Structured information extraction from complex scientific text with fine-tuned large language models. *arXiv preprint: Arxiv-2212.05238* **2022**,
- [80] Walker, N.; Dagdelen, J.; Cruse, K.; Lee, S.; Gleason, S.; Dunn, A.; Ceder, G.; Alivisatos, A. P.; Persson, K. A.; Jain, A. Extracting Structured Seed-Mediated Gold Nanorod Growth Procedures from Literature with GPT-3. *arXiv preprint: Arxiv-2304.13846* **2023**,
- [81] Neo4j, Neo4j - The World's Leading Graph Database. 2012; <http://neo4j.org/>.
- [82] Kearnes, S. M.; Maser, M. R.; Wlekliniski, M.; Kast, A.; Doyle, A. G.; Dreher, S. D.; Hawkins, J. M.; Jensen, K. F.; Coley, C. W. The Open Reaction Database. *J. Am. Chem. Soc.* **143**, 18820–18826.
- [83] Taori, R.; Gulrajani, I.; Zhang, T.; Dubois, Y.; Li, X.; Guestrin, C.; Liang, P.; Hashimoto, T. B. Stanford Alpaca: An Instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca, 2023.
- [84] Alpaca-LoRA. <https://github.com/tloen/alpaca-lora>.
- [85] Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint:2302.13971* **2023**,
- [86] Mamaghani, Z. G.; Hawboldt, K. A.; MacQuarrie, S. Adsorption of CO₂ using biochar - Review of the impact of gas mixtures and water on adsorption. *J. Environ. Chem. Eng.* **2023**, *11*, 109643.
- [87] Peng, Y.; Krungleviciute, V.; Eryazici, I.; Hupp, J. T.; Farha, O. K.; Yildirim, T. Methane Storage in Metal–Organic Frameworks: Current Records, Surprise Findings, and Challenges. *J. Am. Chem. Soc.* **2013**, *135*, 11887–11894.
- [88] Sahoo, B.; Pandey, V.; Dogonchi, A.; Mohapatra, P.; Thatoi, D.; Nayak, N.; Nayak, M. A state-of-art review on 2D material-boosted metal oxide nanoparticle electrodes: Supercapacitor applications. *J. Energy Storage* **2023**, *65*, 107335.
- [89] Suppiah, D. D.; Daud, W. M. A. W.; Johan, M. R. Supported Metal Oxide Catalysts for CO₂ Fischer–Tropsch Conversion to Liquid Fuels-A Review. *Energy Fuels.* **2021**, *35*, 17261–17278.
- [90] González-Vázquez, M.; García, R.; Gil, M.; Pevida, C.; Rubiera, F. Comparison of the gasification performance of multiple biomass types in a bubbling fluidized bed. *Energy Convers. Manag.* **2018**, *176*, 309–323.

- [91] Mohsin, M.; Farhan, S.; Ahmad, N.; Raza, A. H.; Kayani, Z. N.; Jafri, S. H. M.; Raza, R. The electrochemical study of $\text{Ni}_x\text{Ce}_{1-x}\text{O}_{2-\delta}$ electrodes using natural gas as a fuel. *New J. Chem.* **2023**, *47*, 8679–8692.
- [92] Kaur, P.; Singh, K. Review of perovskite-structure related cathode materials for solid oxide fuel cells. *Ceram. Int.* **2020**, *46*, 5521–5535.
- [93] Sengottuvelu, R. jsonformer. <https://github.com/1rgs/jsonformer>, 2018.
- [94] Sengottuvelu, R. jsonformer. <https://github.com/martinezpl/jsonformer/tree/add-openai>, 2018.
- [95] Choudhary, K.; Kelley, M. L. ChemNLP: A Natural Language Processing based Library for Materials Chemistry Text Data. *arXiv preprint arXiv:2209.08203* **2022**,
- [96] 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. *J. Mach. Learn. Res.* **2020**, *21*, 5485–5551.
- [97] Wolf, T. et al. Transformers: State-of-the-Art Natural Language Processing. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. 2020.
- [98] Editorials, N. Tools such as ChatGPT threaten transparent science; here are our ground rules for their use. *Nature* **2023**, *613*, 10–1038.
- [99] Mollick, E. R.; Mollick, L. Using AI to Implement Effective Teaching Strategies in Classrooms: Five Strategies, Including Prompts. *SSRN Electronic Journal* **2023**,
- [100] Kim, S.; Chen, J.; Cheng, T.; Gindulyte, A.; He, J.; He, S.; Li, Q.; Shoemaker, B. A.; Thiessen, P. A.; Yu, B.; Zaslavsky, L.; Zhang, J.; Bolton, E. E. PubChem 2023 update. *Nucleic Acids Res.* **2022**, *51*, D1373–D1380.
- [101] Kim, S.; Thiessen, P. A.; Cheng, T.; Yu, B.; Bolton, E. E. An update on PUG-REST: RESTful interface for programmatic access to PubChem. *Nucleic Acids Res.* **2018**, *46*, W563–W570.
- [102] Kim, S.; Chen, J.; Cheng, T.; Gindulyte, A.; He, J.; He, S.; Li, Q.; Shoemaker, B. A.; Thiessen, P. A.; Yu, B.; Zaslavsky, L.; Zhang, J.; Bolton, E. E. PubChem 2019 update: improved access to chemical data. *Nucleic Acids Res.* **2018**, *47*, D1102–D1109.
- [103] Campbell, Q.; Herington, J.; White, A. D. Censoring chemical data to mitigate dual use risk. *arXiv preprint: Arxiv-2304.10510* **2023**,

Supporting Information for: 14 Examples of How LLMs Can Transform Materials Science and Chemistry: A Reflection on a Large Language Model Hackathon

Kevin Maik Jablonka ^{1,*} Qianxiang Ai ^{2,†} Alexander Al-Feghali ^{3,†} Shruti Badhwar ^{4,†}
Joshua D. Bocarsly ^{5,†} Andres M Bran ^{6,7,†} Stefan Bringuier ^{8,†} L. Catherine Brinson ^{9,†}
Kamal Choudhary ^{10,†} Defne Circi ^{9,†} Sam Cox ^{11,†} Wibe A. de Jong ^{12,†}
Matthew L. Evans ^{13,14,†} Nicolas Gastellu ^{3,†} Jerome Genzling ^{3,†} María Victoria Gil ^{15,†}
Ankur K. Gupta ^{12,†} Zhi Hong ^{16,†} Alishba Imran,^{17,†} Sabine Kruschwitz ^{18,†} Anne Labarre ^{3,†}
Jakub Lála ^{19,†} Tao Liu ^{3,†} Steven Ma ^{3,†} Sauradeep Majumdar ^{1,†} Garrett W. Merz ^{20,†}
Nicolas Moitessier ^{3,†} Elias Moubarak ^{1,†} Beatriz Mouriño ^{1,†} Brenden Pelkie ^{21,†}
Michael Pieler ^{22,23,†} Mayk Caldas Ramos ^{11,†} Bojana Ranković ^{6,7,†} Samuel G. Rodrigues ^{19,†}
Jacob N. Sanders ^{24,†} Philippe Schwaller ^{6,7,†} Marcus Schwarting,^{25,†} Jiale Shi ^{2,†}
Berend Smit ^{1,†} Ben E. Smith ^{5,†} Joren Van Herck ^{1,†} Christoph Völker ^{18,†} Logan Ward ^{26,†}
Sean Warren ^{3,†} Benjamin Weiser ^{3,†} Sylvester Zhang,^{3,†} Xiaoqi Zhang ^{1,†} Ghezal Ahmad Zia ^{18,†}
Aristana Scourtas ²⁷ KJ Schmidt,²⁷ Ian Foster ²⁸ Andrew D. White ¹¹ and Ben Blaiszik ^{27,‡}

¹Laboratory of Molecular Simulation (LSMO),
Institut des Sciences et Ingénierie Chimiques,
Ecole Polytechnique Fédérale de Lausanne (EPFL), Sion, Valais, Switzerland.

²Department of Chemical Engineering, Massachusetts Institute of Technology,
Cambridge, Massachusetts 02139, United States.

³Department of Chemistry, McGill University, Montreal, Quebec, Canada.

⁴Reincarnate Inc.

⁵Yusuf Hamied Department of Chemistry, University of Cambridge,
Lensfield Road, Cambridge, CB2 1EW, United Kingdom.

⁶Laboratory of Artificial Chemical Intelligence (LIAC),
Institut des Sciences et Ingénierie Chimiques,
Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland.

⁷National Centre of Competence in Research (NCCR) Catalysis,
Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland.

⁸Independent Researcher, San Diego, CA, United States.

⁹Mechanical Engineering and Materials Science, Duke University, United States.

¹⁰Material Measurement Laboratory, National Institute of Standards and Technology, Maryland, 20899, United States.

¹¹Department of Chemical Engineering, University of Rochester, United States.

¹²Applied Mathematics and Computational Research Division,
Lawrence Berkeley National Laboratory, Berkeley, CA 94720, United States.

¹³Institut de la Matière Condensée et des Nanosciences (IMCN),
UCLouvain, Chemin des Étoiles 8, Louvain-la-Neuve, 1348, Belgium.

¹⁴Matgenix SRL, 185 Rue Armand Bury, 6534 Gozée, Belgium.

¹⁵Instituto de Ciencia y Tecnología del Carbono (INCAR),
CSIC, Francisco Pintado Fe 26, 33011 Oviedo, Spain.

¹⁶Department of Computer Science, University of Chicago, Chicago, Illinois 60637, United States.

¹⁷Computer Science, University of California,
Berkeley, Berkeley CA 94704, United States.

¹⁸Bundesanstalt für Materialforschung und -prüfung,
Unter den Eichen 87, 12205 Berlin, Germany.

¹⁹Francis Crick Institute, 1 Midland Rd, London NW1 1AT, United Kingdom.

²⁰American Family Insurance Data Science Institute,
University of Wisconsin-Madison, Madison WI 53706, United States.

²¹Department of Chemical Engineering, University of Washington, Seattle, WA 98105, United States.

²²OpenBioML.org

²³Stability.AI

²⁴Department of Chemistry and Biochemistry,
University of California, Los Angeles, CA 90095, United States.

²⁵Department of Computer Science, University of Chicago, Chicago IL 60490, United States.

²⁶Data Science and Learning Division, Argonne National Lab, United States.

²⁷Globus, University of Chicago, Data Science and Learning Division, Argonne National Lab, United States.

²⁸*Department of Computer Science, University of Chicago,
Data Science and Learning Division, Argonne National Lab, United States.*

Contents

I. Predictive Modeling	3
A. Leveraging LLMs for Accurate Molecular Energy Predictions	3
B. From Text to Cement: Developing Sustainable Concretes Using In-Context Learning	6
C. Molecule Discovery by Context	8
D. Text template paraphrasing with LLMs	10
1. Problem	10
2. Solution	10
3. Impact	12
4. Lessons learned	12
E. GA without genes	13
II. Automation and novel interfaces	18
A. Using chain-of-thought and chemical tools to answer materials questions	18
B. sMolTalk	20
C. whinchat : A Conversational electronic lab notebook (ELN) Interface	22
D. BOLLaMa	25
III. Knowledge Extraction	27
A. InsightGraph	27
B. Extracting Structured Data from Free-form Organic Synthesis Text	29
C. TableToJson: Extracting structured information from tables in scientific papers	31
D. AbstractToTitle & TitleToAbstract: text summarization and text generation	36
1. Problem	36
2. Solution	36
3. Example	36
IV. Education	38
A. i-Digest	38
V. Meta analysis of the workshop contributions	40
References	42

* mail@kjablonka.com

† These authors contributed equally

‡ blaiszik@uchicago.edu

I. Predictive Modeling

A. Leveraging LLMs for Accurate Molecular Energy Predictions

Table I. *LIFT for molecular atomization energies on the QM9-G4MP2 dataset.* Metrics for models tuned on 90% of the QM9-G4MP2 dataset (117,232 molecules), using 10% (13,026 molecules) as a holdout test set. Note that the metric used for the baseline results [1] is MAE, whereas this work used the MAD. The results indicate that the LIFT framework can also be used to build predictive models for atomization energies, that can reach chemical accuracy using a Δ -ML scheme.

mol. repr. & framework	G4(MP2) Atomization Energy		(G4(MP2)-B3LYP) Atomization Energy	
	R ²	MAD / eV	R ²	MAD / eV
SMILES: GPTChem	0.984	0.99	0.976	0.03
SELFIES: GPTChem	0.961	1.18	0.973	0.03
SMILES: GPT2-LoRA	0.931	2.03	0.910	0.06
SELFIES: GPT2-LoRA	0.959	1.93	0.915	0.06
SchNet baseline	-	-	-	0.0045
FCHL baseline	-	0.0223	-	0.0052

Accurate prediction of chemical properties has long been the ultimate objective in computational chemistry and materials science. However, the significant computational demands of precise methods often hinder their routine application in modeling chemical processes. The recent surge in machine learning development, along with the subsequent popularity of large language models (LLMs), offers innovative and effective approaches to overcome these computational limitations. Our project takes steps toward establishing a comprehensive, open-source framework that harnesses the full potential of LLMs to accurately model chemical problems and uncover novel solutions to chemical challenges. In this study, we assessed the capability of LLMs to predict the atomization energies of molecules at the G4(MP2) [2] level of theory from the QM9-G4MP2 dataset [3, 4] using solely string representations for molecules, specifically, SMILES [5] and SELFIES [6, 7]. G4(MP2) is a highly accurate composite quantum chemistry method, known for its accuracy within 1.0 kcal/mol for molecular energies compared to experimental values, making atomization energy an ideal property to predict to demonstrate the usefulness and impact of LLMs on the field of computational chemistry.

Jablonka et al. [8] recently demonstrated the potential of fine-tuning pre-trained LLMs on chemistry datasets for a broad array of predictive chemistry tasks. As an initial validation for our project, we fine-tuned generative pretrained transformer (GPT)-3 [9] to learn how to reproduce a molecule’s atomization energy at the G4(MP2) level of theory, using its SMILES or SELFIES string through the prompt, “What is the G4MP2 atomization energy in kcal/mol of ‘SMILES/SELFIES string of a molecule’?” Additionally, we fine-tuned LLMs to predict the atomization energy difference between B3LYP/6-31G(2df,p) and G4(MP2) levels of theory with the prompt, “What is the G4MP2 and B3LYP atomization energy difference in kcal/mol of ‘SMILES/SELFIES string of a molecule’?”, which mirrors the Δ -machine learning (Δ -ML) schemes [10] found in the existing literature.

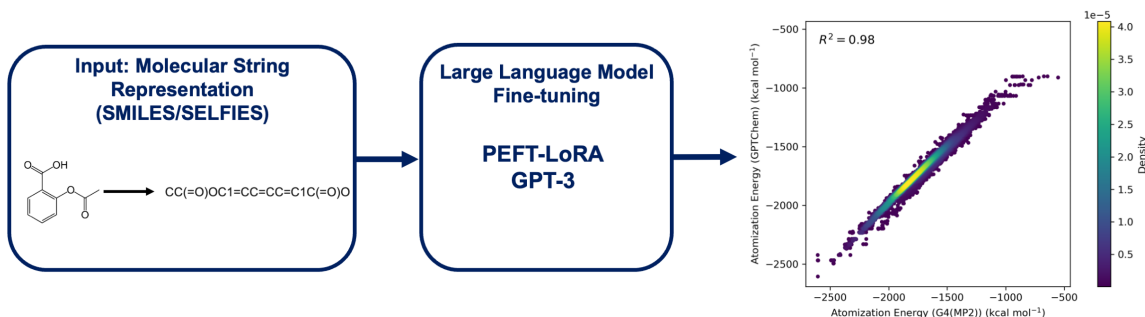


Figure 1. *Illustration of the molecular property prediction workflow, from molecule representation to model fine-tuning and performance evaluation.*

We fine-tuned the GPT-3 (Ada) model using 90% of the QM9-G4MP2 dataset (117,232 molecules) for eight epochs with the GPTChem [8] framework’s default settings. The remaining 10% (13,026 molecules) was kept as the hold-out set, following the same data split as Ward et al. [1], to evaluate the model’s performance. Table I summarizes the regression metrics for the hold-out set. The strong correlation between the predicted and ground truth values suggests that the model effectively learned the structural information from the molecular string representation. Although the MAD remains relatively high compared to state-of-the-art models in the literature [1, 11] that utilize a molecule’s full 3D structural information for descriptor construction, we achieved chemical accuracy ($< 1.0 \text{ kcal/mol} \approx 0.04 \text{ eV}$) for the Δ -ML task. Consequently, this approach can predict G4(MP2) energies with high accuracy when B3LYP energies are available. We also compared the model’s performance using SMILES and SELFIES molecular representations, with the former proving marginally superior for predicting atomization energies, possibly due to its more compact representation for molecules. We additionally calculated regression metrics for the G4MP2-Heavy dataset [1], the results of which are provided in Table II.

Table II. Regression metrics, (Coefficient of Determination), and MAD (Mean Absolute Deviation) for predicting G4(MP2) and (G4(MP2)B3LYP) atomization energies for the G4MP2-Heavy dataset using a fine-tuned GPT-3 model with GPTChem

mol. repr. & framework	G4(MP2) Atomization Energy R ²	MAD / eV	(G4(MP2)-B3LYP) Atomization Energy R ²	MAD / eV
SMILES: GPTChem	0.583	6.02	0.856	0.13
SELFIES: GPTChem	0.146	9.44	0.659	0.15

While GPT-3 fine-tuning models are accessible through the OpenAI application programming interface (API), their usage costs can become prohibitive for larger datasets, rendering hyperparameter searches and other exploratory research economically unfeasible. Consequently, we aim to develop a free and open-source framework for fine-tuning LLMs to perform a wide range of predictive modeling tasks, encompassing chemical property prediction and inverse design.

To fine-tune a pre-trained LLM locally on a GPU instead of querying OpenAI’s API, we employed the Hugging Face parameter efficient fine-tuning (PEFT) library [12] to implement the low-rank adapters (LoRA) tuning paradigm [13]. Conventional fine-tuning updates all model parameters, utilizing pretrained weights from a large training dataset as a starting point for gradient descent. However, fine-tuning memory-intensive LLMs on consumer hardware is often impractical. The LoRA approach addresses this by freezing the model’s weights and tuning a low-rank adapter layer rather than the entire model, parameterizing changes concerning the initial weights rather than the updated weights.

Using this approach, we fine-tuned the smallest version of GPT-2 [14] (124 million parameters) for 20 epochs on the same 90% training set as used in GPTChem, allocating 10% of that training set for validation, and computed metrics on the same 10% hold-out set as in the GPTChem run, employing the same prompt structure. Although the model performs well, it demonstrates slightly inferior performance to GPT-3 on the G4MP2 task and moderately worse on the (G4(MP2)-B3LYP) task. This is not unexpected, given that GPT-3 is a more recent model with substantially more parameters than GPT-2 (175 billion vs. 124 million) and has exhibited superior few-shot performance on various tasks [15].

Moving forward, we plan to employ the LoRA tuning framework to fine-tune other models, such as LLaMA [16] and GPT-J, to investigate the impact of LLM selection on performance in chemistry-related tasks. Moreover, we intend to experiment with molecular-input representations beyond string formats to more accurately represent a molecule’s 3D environment [17].

One sentence summaries

- a. Problem/Task* Predicting the atomization energies of molecules using large language models.
- b. Approach* Fine-tuning of GPT-3 **ada** model as well as PEFT of a small open-source model (GPT-2) on SMILES to either directly predict the atomization energies or the difference between a lower and a higher level of theory.

c. Results and Impact Even though simpler, direct fine-tuning for a complicated property on SMILES leads to errors one order of magnitude higher than baselines, and the error can only be brought close to the baselines with an $\Delta - ML$ approach—first demonstration of Δ -ML in the LIFT framework for chemistry.

d. Challenges and Future Work Since the predictions without 3D coordinates is not satisfactory, a question for future work is how the approach would perform when provided with 3D coordinates.

B. From Text to Cement: Developing Sustainable Concretes Using In-Context Learning

The inherently intricate chemistry and variability of feedstocks in the construction industry have limited the development of novel sustainable concretes to labor-intensive laboratory testing. This major bottleneck in material innovation has significant consequences due to the substantial contribution of CO₂ emissions of materials in use today. The production of Portland cement alone amounts to approximately 8% of anthropogenic CO₂ emissions [18]. The increasing complexity of alternative raw materials and the uncertain future availability of established substitutes like fly ash and granulated blast furnace slag make the experimental development of more sustainable formulations time-consuming and challenging. Traditional trial-and-error approaches are ill-suited to efficiently explore the vast design space of potential formulations.

In previous studies, inverse design (ID) has been shown to accelerate the discovery of novel, sustainable, and high-performance materials by reducing labor-intensive laboratory testing [19–21]. Despite their potential, the adoption of these techniques has been impeded by several difficulties that are connected to the predictive model at the core of ID: Incorporating domain knowledge typically requires extensive data collection to accurately capture underlying relationships, which makes representing complex tasks in practice challenging due to the high costs of data acquisition. Furthermore, ID necessitates formulating research problems as search space vectors. This process can be unintuitive and challenging for lab personnel, limiting the comprehension and adoption of these techniques. Lastly, sparse training samples in high dimensions can lead to co-linearities and overfitting, negatively impacting prediction performance. With in-context learning

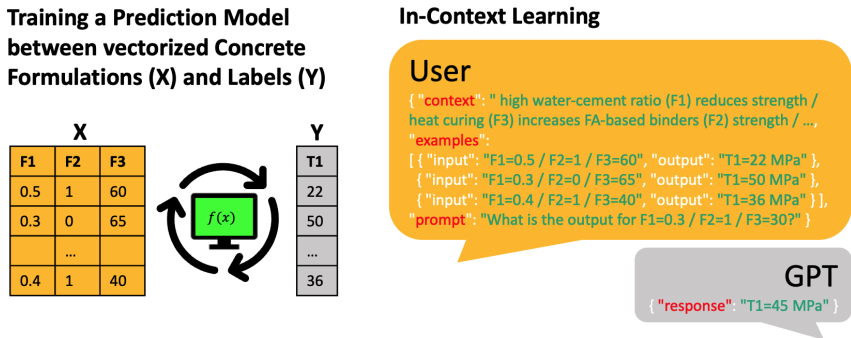


Figure 2. Using LLMs to predict the compressive strength of concretes. The left part illustrates the conventional approach for solving this task, i.e., training classical prediction models using tabular data. Using the LIFT framework LLM can also use tabular data but also leverage context information provided in natural language. Augmented with this context, in-context-learning with LLM leads to a performance that outperforms baselines such as RFs or GPRs.

(ICL), Jablonka et al. [8] and Ramos et al. [22] demonstrated that LLMs offer a solution by incorporating context and general knowledge, providing flexibility in handling non-numeric inputs and overcoming the limitations of traditional vector space formulations (Figure 2).

In this study, we have adopted an ICL approach based on a dataset from a study by Rao and Rao [23]. The dataset comprises 240 alternative and more sustainable concrete formulations based on fly ash and ground granulated slag binders, along with their respective compressive strengths. The goal is to compare the prediction performance of the compressive strength with ICL using the `text-davinci-003` model [24] against established methods, RF [25].

Randomly sampled training subsets containing ten formulations are drawn. The prediction performance is assessed on a separate, randomly sampled test set of 25 samples and evaluated using the coefficient of determination (R-squared) [26]. This process is repeated ten times to ensure more reliable results.

The experimental results reveal that ICL attains comparable performance to GPR but underperforms RF when provided with small training data sets (R-squared of 0.5, 0.54, and 0.67, respectively). However, when using general, qualitative concrete design knowledge, such as the influence of the water-to-cement ratio on strength, the models significantly reduce prediction outliers and ultimately surpass RF (R-squared = 0.71). When we incorrectly changed the context of the ratio of fly ash to GGBFS, it negatively affected the R-squared value for ICL, causing it to drop to 0.6. This misrepresentation of the rule led to a decrease in the model’s predictive accuracy, demonstrating that the quality of the information included in the “fuzzy” context is critical to the overall performance of LLMs. It should be noted, however, that the impact on the R-squared value may vary depending on the importance of the rule in the overall context. That is, not all

changes in context have a similar impact, and the drop to 0.6 might occur only in the case of the ratio of fly ash to GGBFS. Other studies, such as those conducted in the LIFT work, [27] have shown LLM performance for minor changes in wording or the presence of noise in the features. In these experiments, the robustness of LIFT-based predictions was comparable to classical ML algorithms, making it a promising alternative for using fuzzy domain knowledge in predictive modeling.

LLMs have been shown to provide significant advantages in sustainable concrete development, including context incorporation, adaptable handling of non-numeric inputs, and efficient domain knowledge integration, surpassing traditional methods' limitations. ICLs simplifies formulating data-driven research questions, increasing accessibility and democratizing a data-driven approach within the building materials sector. This highlights LLMs potential to contribute to the construction industry's sustainability objectives and foster efficient solutions.

One sentence summaries

- a. Problem/Task* Predicting the compressive strength of concrete formulations.
- b. Approach* ICL on language-interfaced tabular data, with and without “fuzzy” domain expertise (such as relationship between columns) provided in natural language.
- c. Results and Impact* Predictive models can be built without any training; if provided with domain expertise, those models outperform the baselines—first demonstration in chemistry of such fuzzy knowledge can be incorporated into models.
- d. Challenges and Future Work* ICL can be very sensitive to the prompt, hence future work should investigate the robustness of this approach.

C. Molecule Discovery by Context

The escalating climate crisis necessitates the deployment of clean, sustainable fuels to reduce carbon emissions. Hydrogen, with its potential to prevent approximately 60 gigatons of CO₂ emissions by 2050, according to the World Economic Forum, stands as a promising solution [28]. However, its storage and shipping remain formidable challenges due to the necessity for high-pressure tanks. To address this, we sought new molecules to which hydrogen could be conveniently added for storage. Traditional screening methods, like brainstorming, are insufficient due to their limited throughput. This research proposes a novel method of leveraging ScholarBERT, [29] a pre-trained science-focused LLM, to screen potential hydrogen carrier molecules efficiently. This approach utilizes ScholarBERT’s ability to understand and relate the context of scientific literature. The data used for this study consisted of three datasets. The “Known” dataset comprised 78 known hydrogen carrier molecules. The “Relevant” dataset included 577 molecules, all of which are structurally similar to the “Known” molecules. The “Random” dataset contained 111 randomly selected molecules from the PubChem database [30]. The first step involved searching for contexts for molecules in the Public Resource Dataset (PRD), which includes 75M English language research articles. These contexts (i.e. sentences that mentioned the molecule name) were then fed into ScholarBERT. For each context, three calculations were made:

1. the average of the last four encoder layers in ScholarBERT
2. the average embedding of all tokens constituting the molecule name as one contextualized embedding for this molecule, and
3. the average of all contextualized embeddings for a molecule as ScholarBERT’s representation of this molecule.

Subsequently, we calculated the similarity between the known and candidate molecules. The definition of “similarity” used in this study was the cosine similarity between the ScholarBERT representations of two molecules. We then sorted the candidates based on the similarity score in descending order, with a higher score indicating greater potential as a hydrogen carrier. Figure 3 and 4 show the candidate molecules with the highest similarity to the known molecules. We can see that ScholarBERT does a passable job finding similar molecules from the random set. We do see that it favors finding molecules with 5- and 6-member rings, though with features we didn’t expect, like halogens. On the other hand, ScholarBERT does a much better job when we reduce the search space to those with structural similarity. We see that molecules with 5-member rings, for instance, are found to be similar structurally and in how they are described in the literature via ScholarBERT.

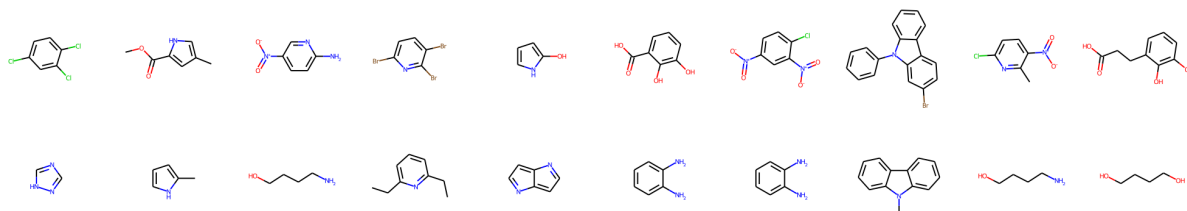


Figure 3. Each column shows a Known molecule on the bottom and its top candidate molecule from the Random set on the top

Based on our empirical data, computing the energy capacity (wt%H₂) and energy penalty (kJ/mol/H₂) of adding and removing H₂ to the molecule (which are the quantitative “success metrics” for this project) of a candidate molecule using traditional quantum chemistry takes around 30 seconds per molecule on a 64-core Intel Xeon Phi 7230 processor, whereas the proposed LLM approach can screen around 100 molecules per second on a V100 GPU, achieving a 3000 times speedup.

One sentence summaries

- Problem/Task* Recommending hydrogen carrier molecules.

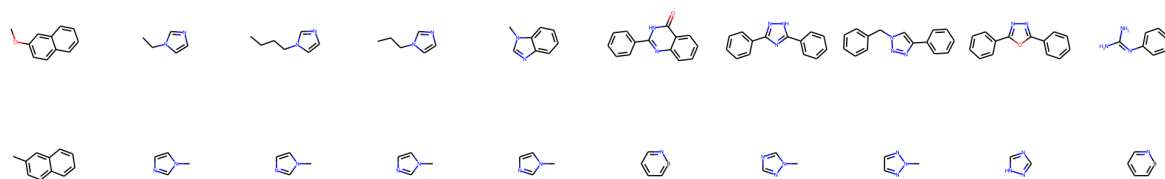


Figure 4. Each column shows a Known molecule on the bottom and its top candidate molecule from the Relevant set on the top

b. Approach Ranking of molecules based on similarity of embeddings created with a language model by showing aggregating the embeddings of mentions of the molecules in scientific texts.

c. Results and Impact Approach can recommend molecules with a success rate better than random.

d. Challenges and Future Work More benchmarks compared to conventional generative modeling are needed.

D. Text template paraphrasing with LLMs

1. Problem

Text data is much trickier to augment for machine learning applications due to the discrete nature of the data modality. There are some traditional augmentation approaches for these tasks. However, they can be inefficient or still need extensive manual checks to be sure they deliver the desired results, especially for scientific or chemistry applications.

2. Solution

To automate high-quality text data augmentations, LLMs have been explored by Dai and his coworkers [31] as a very recent and promising solution to this problem. We investigated such a setup in the scope of the OpenBioML chemistry NLP project (<https://github.com/OpenBioML/chemnlp>) to paraphrase text templates for the insertion of chemical raw data into natural language for LIFT. [27]

An example prompt is shown below. The outlined prompt setup has after “Question:” the desired task with additional information and after “Sentence:” the starting text template for the paraphrasing. The “Question:” and “Answer:” headers are not used if the LLM interface uses a chat interface, i.e., with OpenAI GPT-4.

Example Prompt

Question: Please paraphrase the sentence below ten times without changing the original meaning and the placeholder in the curly {} brackets. Please use all the placeholders in the curly {} brackets for every rephrased sentence.

Sentence: The molecule with the {SMILES_description} representation {SMILES_data} has a {target_name} of {target_value}{target_units}.

Answer:

Example Response by GPT-4

1. The {SMILES_description} representation of the molecule is {SMILES_data}, and its {target_name} is {target_value}{target_units}.
2. With a {SMILES_description} representation of {SMILES_data}, the molecule has a {target_name} of {target_value}{target_units}.
3. The molecule, represented by {SMILES_description} as {SMILES_data}, possesses a {target_name} of {target_value}{target_units}.
4. A molecule having a {SMILES_description} representation of {SMILES_data} also has a {target_name} equal to {target_value}{target_units}.
5. The {SMILES_description}-represented molecule with {SMILES_data} exhibits a {target_name} of {target_value}{target_units}.
6. For the molecule with a {SMILES_description} representation of {SMILES_data}, its {target_name} measures {target_value}{target_units}.
7. The {target_name} of the molecule having a {SMILES_description} representation of {SMILES_data} is {target_value}{target_units}.
8. Represented by {SMILES_description} as {SMILES_data}, the molecule’s {target_name} is {target_value}{target_units}.

9. The molecule's {target_name} is {target_value}{target_units} when its {SMILES_description} representation is {SMILES_data}.
10. A {target_name} of {target_value}{target_units} is associated with the molecule whose {SMILES_description} representation is {SMILES_data}.

Example Response by Anthropic Claude v1

Here are 10 paraphrased sentences of the original sentence with the placeholders retained:

1. The molecule represented by the {SMILES_description} code {SMILES_data} has a {target_name} of {target_value}{target_units}.
2. The {SMILES_description} notation {SMILES_data} denotes a molecule with a {target_name} equal to {target_value}{target_units}.
3. A {target_name} of {target_value}{target_units} is possessed by the molecule depicted using {SMILES_description} as {SMILES_data}.
4. {SMILES_data} signifies the {SMILES_description} of the molecule which has a {target_name} of {target_value}{target_units}.
5. The {SMILES_description} {SMILES_data} characterizes the molecule that exhibits a {target_name} of {target_value}{target_units}.
6. A {target_name} equal to {target_value}{target_units} belongs to the molecule expressed with {SMILES_description} code {SMILES_data}.
7. {SMILES_data} denotes the {SMILES_description} for the molecule possessing a {target_name} of {target_value}{target_units}.
8. The molecule defined by {SMILES_description} {SMILES_data} presents a {target_name} of {target_value}{target_units}.
9. {SMILES_description} {SMILES_data} specifies the molecule which displays a {target_name} of {target_value}{target_units}.
10. A {target_name} amounting to {target_value}{target_units} is a property of the molecule represented with {SMILES_description} {SMILES_data}.

Example Response by GPT-J

The molecule with the {SMILES_description} representation {SMILES_A} has a {target_name} of {target_value}{target_units}. {SMILES_A} has a {target_name} of {target_units}. The molecule with the {target_units} representation {target_name} has a TARGET_UNITS of {target_value}{target_units}. {target_name} of {target_units} have a TARGET_UNITS of {target_value}{target_units}.

In the above answer, there is the {SMILES_description} representation of {SMILES_query}, but we don't use it in the sentence yet. And there is no curly brackets for the expected answer.

3. *Impact*

The outlined approach allows to automatically create new paraphrased high-quality prompts for LIFT LLM training data very efficiently. With the additional paraphrased text templates, overfitting to special text passages should be avoided. We explore this setup in follow-up work in more detail.

4. *Lessons learned*

The outlined paraphrasing setup works well for the latest state-of-the-art models, e.g., OpenAI’s GPT-4 and Anthropic’s Claude v1. Less capable open-source models seem to lack the understanding of this paraphrasing task. Still, new and upcoming open-source LLM efforts could change that soon, enabling a cost-effective and broader application of this setup.

One sentence summaries

- a. Problem/Task* Generation of many text-templates for language-interfaced fine-tuning of LLMs
- b. Approach* Prompting of LLM to rephrase templates (with template syntax similar to Jinja).
- c. Results and Impact* Large models (GPT-4, Claude), in contrast to smaller ones, can successfully rephrase templates, offering a potential avenue for data-augmentation.
- d. Challenges and Future Work* As next step, ablation studies need to be carried out that test the effect of data augmentation by template rephrasing on regression and classification case studies.

E. GA without genes

We investigate the ability for a LLM to work in parallel with genetic algorithms (GAs) for molecular property optimization. By employing a LLM to guide genetic algorithm operations, it could be possible to produce better results using fewer generations. We hypothesize that a GA can take advantage of the “smart” randomness of the outputs of the LLM. This work explores the potential of LLMs to improve molecular fragmentation, mutation, variation, and reproduction processes and the ability of a LLM to gather information from a simplified molecular-input line-entry system (SMILES) string [5, 6] and an associated score to produce new SMILES strings. Although computational efficiency is not the primary focus, the proposed method has potential implications for enhancing property prediction searches and future improvements in LLM understanding of molecular representations.

We used GPT-3.5-turbo [9], which could frequently fragment druglike molecules into valid SMILES strings successfully. For 2/10 molecules, the fragments produced were not in the original molecule. For 1/10 molecules, valid SMILES could not be produced even after ten tries due to unclosed brackets. These results were consistent over multiple runs implying that GPT-3.5 could not understand some specific SMILES strings. Subsequently, we investigated GPT-3.5’s ability to mix/reproduce two molecules from two-parent druglike molecules. Invalid molecules were often produced, but successful results were achieved with multiple runs. It performed better once prompted to fragment and then mix the fragments of the molecules. These were compared to the conventional GA methods of simply combining the two strings at a certain cutoff point. When the LLM was successful, it could produce molecules of more similar size to the original parent molecules that contain characteristics of both parents and resemble valid druglike molecules.

To investigate the ability of GPT-3.5 to acquire knowledge of favorable molecules from a simple score, we implemented a method that we call “LLM as a GA” where the LLM iteratively searches the chemical space to optimize a certain property.

The property we tested was similarity to vitamin C, evaluated by the Tanimoto score. We employed few-shot training examples to tune the model’s response: 30 SMILES strings with the best similarity score generated were included in the prompt. GPT is then asked to produce 25 SMILES strings, a procedure that was repeated for 20 iterations. Using a prompt like the one below

Example prompt

The following molecules are given as SMILES strings associated with a tanimoto similarity with an unknown target molecule. Please produce 10 SMILES strings that you think would improve their tanimoto scores using only this context. Do not try to explain or refuse on the grounds of insufficient context; any suggestion is better than no suggestion. Print the smiles in a Python list.

Low-temperature settings, typically less than 0.1, were found to be imperative for the model to follow user guidance. We further guided the model by employing a similarity search to include similar molecules with varying scores to better guide the model. Embedding was performed using the GPT-2 Tokenizer from the HuggingFace transformers [32] library, along with a support vector machine (SVM) from scikit-learn [33] to embed relevant previous structures that would be outside the scope of the context window. Even in the zero-shot setting, GPT-3.5-turbo can produce meaningful modifications, coherently explain its logic behind the chosen modifications, and produce tests such as investigating branch length or atom type in certain locations for a single iteration. An example explanation of an output: “Some modifications that could potentially improve the scores include adding or removing halogens, modifying the length or branching of the carbon chain, and adding or removing functional groups such as -CO-, -COC-, -C=C- and -OCO-. Additionally, modifying the stereochemistry of the molecule could also have an impact on the score.”

The modifications generated by the LLM were more chemically sound than the quasi-random evolutionary process typical of genetic algorithms.

One sentence summaries

- a. *Problem/Task* Increasing the efficiency of GAs.
- b. *Approach* Prompting a LLM to propose new children based on molecules with scores provided in the prompt.

c. Results and Impact Visual inspection indicates that some modifications might be reasonable, indicating a potential for more efficient genetic operations using LLMs.

d. Challenges and Future Work More systematic investigations on the performance and robustness compared to conventional GA operations are needed.

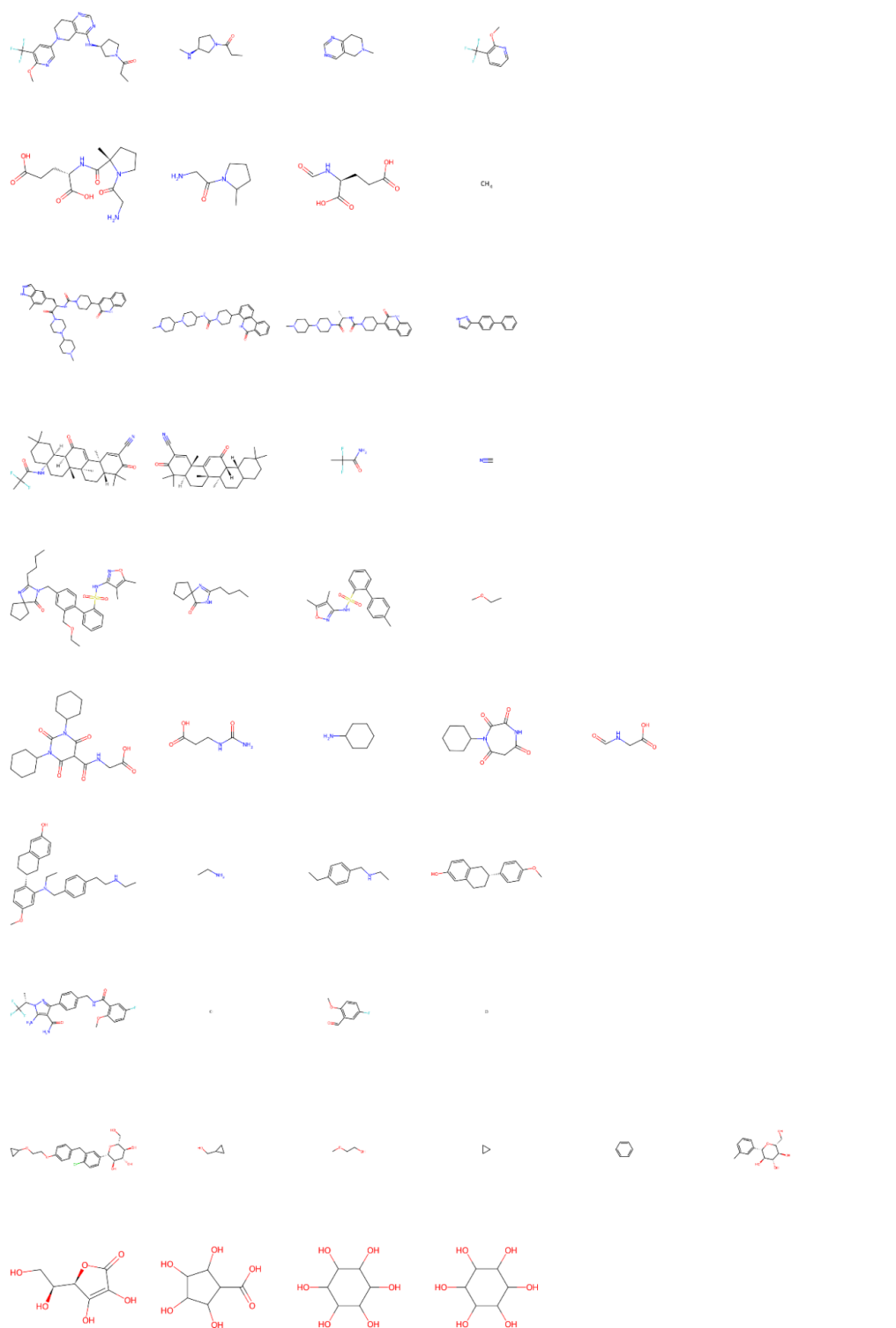


Figure 5. Using GPT to fragment molecules. Original molecules are in column one with LLM created fragment to the right. The LLM can frequently fragment molecules into valid SMILES strings successfully. $\frac{2}{10}$ times fragments produced were not in the original molecule (rows 6 and 10). For $\frac{1}{10}$ molecules, valid SMILES were able to be produced even after ten attempts (row 8)

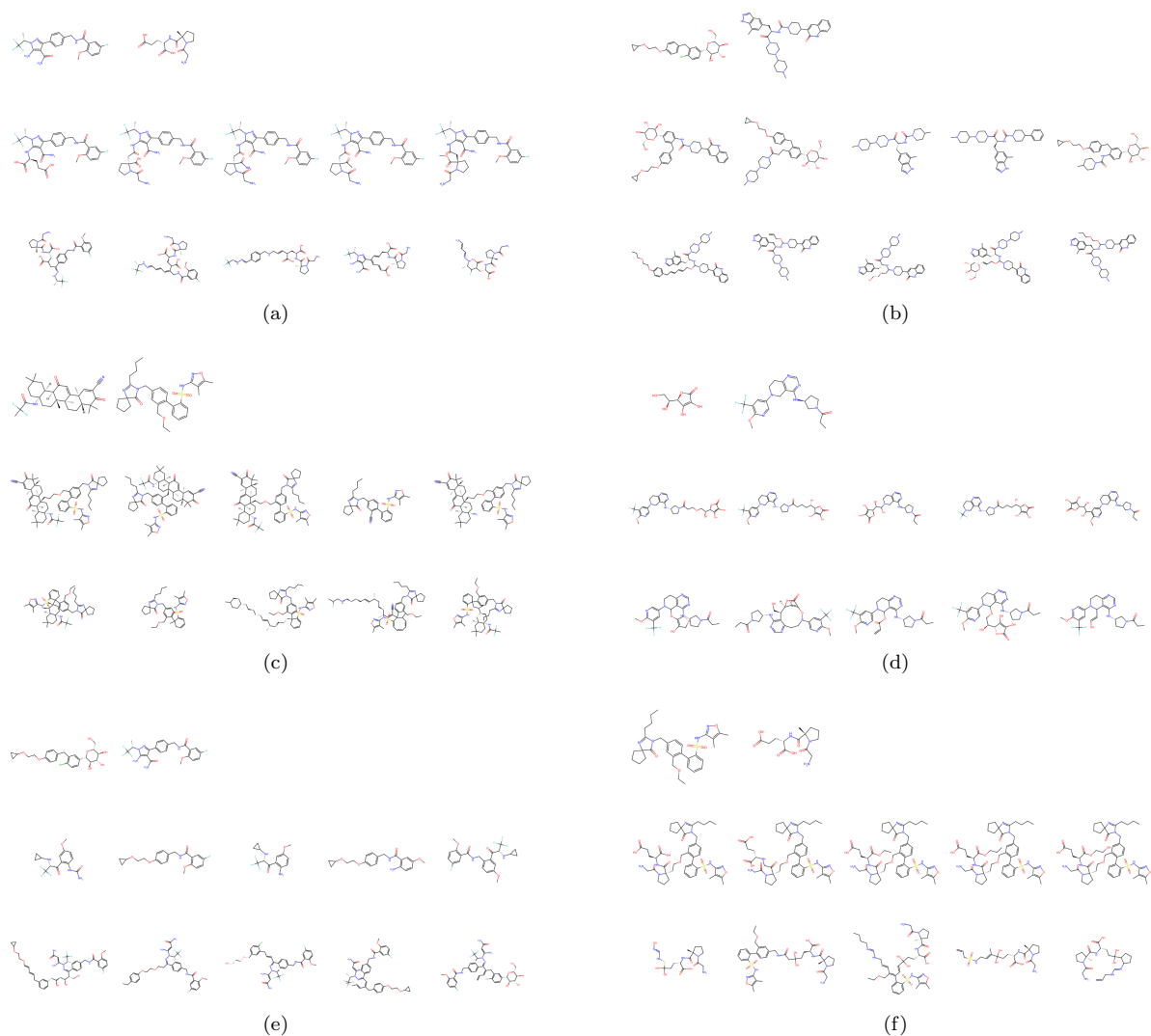


Figure 6. Using GPT-3.5-turbo to reproduce/mix molecules. Two original parent molecules on 1st row, followed by LLM created children, followed by conventional GA string splicing children for comparison

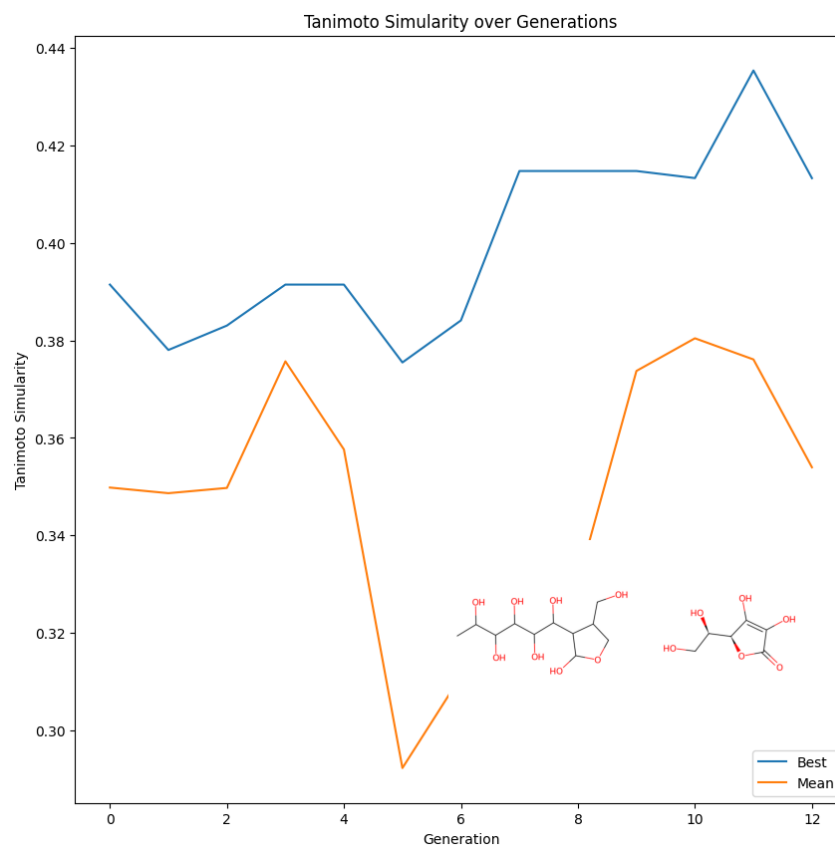


Figure 7. Tanimoto similarity to vitamin C as a function of GA generations. Conventional GA run for 30 generations and the best score (most similar to vitamin C) of each generation is given to the LLM as a LLM along with its associated Tanimoto similarity score to Vitamin C. LLM was then asked to create new molecules and improve the score for 12 generations. Multiple new best molecules were found using LLM as shown by the blue line.

II. Automation and novel interfaces

A. Using chain-of-thought and chemical tools to answer materials questions

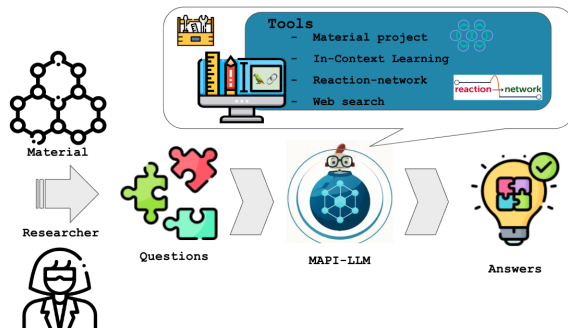


Figure 8. *Schematic overview of the MAPI-LLM workflow.* It uses LLMs to process the user’s input and decide which available tools (e.g., Materials Project API, and Google Search) to use following an iterative chain-of-thought procedure. In this way, it can answer questions such as “Is the material AnByCz stable?”.

LLMs have demonstrated remarkable success in various tasks [34–36]. Recently, LLMs have gained attention in chemistry, demonstrating exceptional ability to model chemical systems [37] and predicting tabular data [8, 22, 27]. Predicting the properties of materials is challenging since it requires computationally intensive techniques, such as density functional theory (DFT) [38–40]. Data-driven models offer a viable option to balance accuracy and computational time. Here we presented the MAPI-LLM, a multi-task package that employs LangChain [41] agents with access to multiple tools to address users’ questions about materials.

It has been shown that providing chemistry-specific tools to an LLM allows the LLM to solve chemistry problems with significantly higher accuracy [42]. In a similar manner, we developed tools to iteratively query the Materials Project (MAPI) dataset [43] and utilize the reaction-network package [44], among others. MAPI-LLM can process user prompts in natural language using LLMs and follow a chain of thought (COT) [45] approach to determine the most suitable tools and inputs to answer the prompt. Due to MAPI-LLM’s design, more tools can be added as needed, and tools can be combined (multiple tools can be used for a given prompt), opening the door for a large variety of applications. Figure 8 illustrates MAPI-LLM’s capabilities. The code for the app is available in https://github.com/maykcaldas/MAPI_LLM, and a graphical user interface (GUI) is implemented in https://huggingface.co/spaces/maykcaldas/MAPI_LLM.

An important feature implemented into MAPI-LLM is a technique known as ICL [9], which allows the model to learn from the context within the prompt. For example, users can use MAPI-LLM’s tool to query the MAPI dataset, first triggering the dataset search in the COT. However, if the desired material is not found in the dataset, MAPI-LLM still has access to other tools (such as ICL) to build context around the user prompt and adjust the COT actions to make a prediction. Another interesting tool is the ability to use the reaction-network package [44], which is a package for predicting inorganic reaction pathways. We showed the promising capabilities of MAPI-LLM by simply asking for reactions that use a given material as reactants or products. It can suggest such reactions for material synthesis or decomposition.

We built from the knowledge that LLMs are suitable for such tasks of interest in this application, for instance, classification and regression tasks [8]. Nevertheless, this application still needs a systematic validation of its predictions, such as the reinforcement learning from human feedback (RLHF) implementation in GPT-3.5 [46].

One sentence summaries

- a. *Problem/Task* Answering complex materials science questions based on reliable data and tools.
- b. *Approach* LLM-based agent in the ReAct framework that has access to tools such as the Materials Project API and uses ICL to answer questions for materials that are not in the materials project.
- c. *Results and Impact* Coupling of tools allows answering questions that none of the tools or LLMs alone could solve by themselves, providing a very accessible interface to materials informatics tools.

d. Challenges and Future Work If a description of tools is incorporated in the prompt, this limits the number of tools that can be coupled. In addition, LLM agents still tend to not perform equally well on all prompts, and systematic investigation to better understand this and to increase the robustness is needed.

B. sMolTalk

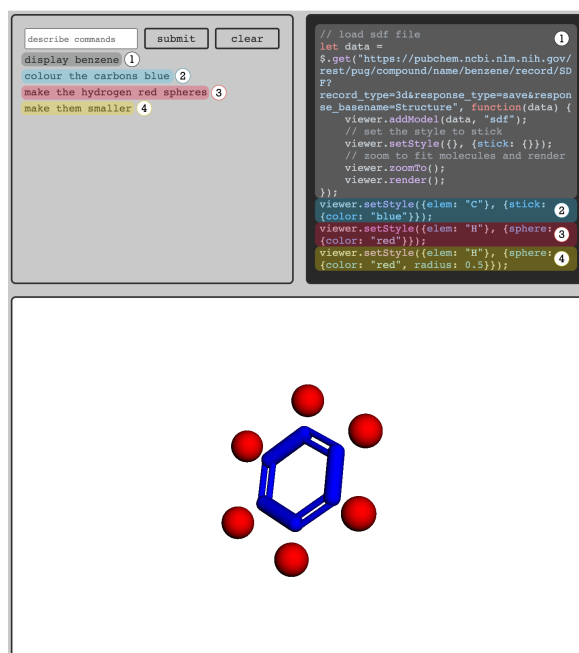


Figure 9. *The sMolTalk interface.* Based on few-shot prompting LLMs can create code for visualization tools such as `3dmol.js`.

Since the advent of 3D visualization methods, chemists have employed computers to display their molecules of interest to better understand their underlying structure and properties. Nevertheless, a lot of chemists are not equipped with the required coding skills to use and customize their visualizations. Depending on the package, and its associated documentation, chemists might end up spending hours to days learning the details of the specific visualization software.

We developed a natural language interface that generates code for `3dmol.js`, an open-source visualization JavaScript library [47], meaning the visualizations are run in a web browser (Figure 9). The user input is fed into ChatGPT API, using the GPT-3.5-turbo model. We use in-context learning (few-shot prompting), giving several examples of the user input with the expected JavaScript code that manipulates the `3dmol.js` viewer. Before the user submits further commands, we update the prompt with the current state of the viewer.

The current implementation might lead to a one-stop solution for visualizing and retrieving properties for molecules. This would accelerate chemists’ workflow for querying information about molecules. Furthermore, if an LLM is able to control structural software, it might be possible to perform reasoning on the molecular structure itself. For instance, in drug discovery, one may ask what functional group of the ligand needs to be changed for binding affinity to the protein to increase. Another example might involve proteins, looking at what amino acid residues could be mutated to cysteines in order to create new disulfide bonds between chains. This would presumably require specific fine-tuning and equipping the LLM with more tools. The approach of generating code and structural reasoning might be similar but is most likely going to require a different set of tools that were specifically developed for protein structure manipulation (such as `PyMol` [48], or `MolStar` [49]). Then, another set of highly accurate tools for binding affinity predictions or protein folding is also required. The major problem encountered is prompt leakage, where examples from in-context learning would leak into the actual LLM output. For the best evaluation, it is best to have as few and as different examples as possible. Moreover, although OpenAI’s GPT models can sometimes correctly recall protein data bank (PDB) IDs of proteins or Chemical Abstract Services (CAS) numbers of compounds, it’s not reliable, making tooling the models with API calls to PubChem, or the PDB, much more robust. We are currently developing an agent based on the ReAct approach [50] tooling with these APIs so that correct structures are always retrieved (i.e., to avoid the LLM needs to remember internally all such IDs). This framework would

then help us iteratively add tools to the agent, creating a chatbot one can query about any molecule of interest, including the structural reasoning task mentioned above. Lastly, we hypothesize we could improve the generation of `3dmol.js` code by using self-instruct fine-tuning. Using an external LLM with access to the documentation would create a dataset that could be used for fine-tuning. The same approach might be utilized for generating code for any other type of software, not just visualization packages. Therefore, such LLM could control molecular dynamics software, such as LAMMPS [51], or GROMACS [52].

One sentence summaries

a. Problem/Task Making bioinformatics tools, in particular the visualization software `3dmol.js` accessible to non-experts.

b. Approach Chat-interface by prompting a LLM to produce commands to `3dmol.js`, which are then passed to the software.

c. Results and Impact The LLM can, without consulting the documentation, generate code that often successfully performs the requested actions, demonstrating that LLM might help make tools more accessible by providing access to complex interfaces via natural language.

d. Challenges and Future Work The approach suffers from robustness issues, which might be overcome with conditioning on a knowledge base, more advanced decoding strategies, or error handling (e.g., giving the agent access to error messages and logs and prompting it to reflect).

C. whinchat: A Conversational ELN Interface

Laboratory data management systems such as ELNs and laboratory information systems (LIMS) can record rich data about experimental samples and the experiments that have been performed on them. Then, the user interfaces for these systems typically allow researchers to explore their samples with various pre-programmed visualizations. However, it is difficult for a traditional user interface to have enough flexibility to capture the richness and diversity of real, interconnected, experimental data. Here, our aim was to investigate whether an LLM could process digital laboratory data and serve as a flexible, conversational interface to a data management system. As a proof-of-concept for this approach, we introduced an automated data management assistant into the open-sourced “Datalab” software, which we have been developing to address needs in experimental materials chemistry (<https://github.com/the-grey-group/datalab>). Datalab can be used to capture data and metadata about materials chemistry samples, their synthesis constituents, and resulting devices. The system tracks each sample’s provenance through the research lifecycle and makes this data available via a JavaScript object notation (JSON) API that can be explored programmatically or with a Vue.js-based web user interface (UI). During the hackathon, we embedded a prompted LLM into the UI and gave it access to raw responses from datalab’s JSON API. We used the most recent publicly available models from OpenAI (gpt-3.5-turbo) via their Python API. We found that the LLM does a very good job at interpreting JSON and inferring the meaning of fields (including linked data) and can:

1. *Contextualize existing data*: Explain a set of related experiments from linked JSON responses, resolve acronyms/ short-hand used by experimentalists or create simple text summaries.

Contextualizing existing data

Assistant Hello! I am your virtual assistant and I am here to help you manage your experimental data. How can I assist you today?

User Can you provide a brief summary of this experiment?

Assistant Based on the provided JSON data, this experiment involves the synthesis and characterization of NaCoO₂ as a battery cathode material. The synthesis method used was solid-state reaction, and the starting materials used were sodium carbonate and cobalt (II,III) oxide. The synthesized NaCoO₂ was then used to make an electrode, which was characterized using X-ray diffraction. The electrode was then incorporated into a coin cell battery, which was cycled electrochemically and characterized using various techniques.

2. *Generate text and graphical representations of the data*: Aggregate data from different fields in the JSON response and convert to formats such as tables (potentially formatted for a particular journal), CSV, and even mermaid.js graph visualizations.
3. *Suggest future experiments*: The assistant readily identifies related materials to study, additives to try, additional characterization techniques, and synthesis protocols.

Suggest experiments

User: What NMR nuclei could I probe in this sample?

Assistant: Based on the information provided in the JSON file, the sample is NaCoO₂, which contains sodium, cobalt, and oxygen. Therefore, the NMR nuclei that could be probed in this sample are:

- (a) Sodium-23 (²³Na) NMR: This can provide information about the local environment and coordination of sodium ions in the NaCoO₂ lattice.
- (b) Cobalt-59 (⁵⁹Co) NMR: This can provide information about the local environment and coordination of cobalt ions in the NaCoO₂ lattice.
- (c) Oxygen-17 (¹⁷O) NMR: This can provide information about the local environment and coordination of oxygen ions in the NaCoO₂ lattice.

group’s Wiki and papers. Furthermore, while we have only given the model data from a single project so far (up to 8 samples/starting materials, 1700 tokens of JSON), it would be of great interest to provide the model with a larger context across multiple projects to attempt to facilitate cross-fertilization of ideas. One notable challenge in this area is the limited context size of currently available LLM models (e.g., 4097 tokens for GPT-3.5-turbo). Therefore, future work will investigate larger models (e.g., GPT-4 with 30K token context), as well as approaches to give existing LLMs access to larger context (e.g., an embedding-based approach or allowing an LLM agent to query the OpenAPI directly as needed). At present, we note that the scientific usefulness of this assistant is highly task- and model-dependent; however, any additional interface that can lower the barrier to improving data capture and dissemination in the field should be investigated further and will be a future development target for Datalab.

One sentence summaries

- a. Problem/Task* Providing very flexible access to data in ELNs/LIMS.
- b. Approach* Prompting of a large language model with questions provided in a chat interface and context coming from the response of the API of an LLM.
- c. Results and Impact* The system can successfully provide a novel interface to the data and let user interact with it in a very flexible and personalized way, e.g, creating custom summaries or visuals for which the developers did not implement specific tools.
- d. Challenges and Future Work* Since the current approach relies on incorporating the response of the ELN/LIMS into the prompt, this limits how much context (i.e., how many experiments/samples) the system can be aware of. One potential remedy is to use retrieval-augmented generation, where the entries are embedded in a vector store and the agent will be able to query this database on put (parts of) the most relevant entries into the prompt.

D. BOLLaMa

The field of chemistry is continuously evolving towards sustainability, with the optimization of chemical reactions being a key component [53]. The selection of optimal conditions, such as temperature, reagents, catalysts, and other additives, is challenging and time-consuming due to the vast search space and high cost of experiments [54]. Expert chemists typically rely on previous knowledge and intuition, leading to weeks or even months of experimentation [55].

Bayesian optimization (BO) has recently been applied to chemistry optimization tasks, outperforming humans in optimization speed and quality of solutions [55]. However, mainstream access to these tools remains limited due to requirements for programming knowledge and the numerous parameters these tools offer. To address this issue, we developed BOLLaMa. This artificial intelligence (AI)-powered chatbot simplifies BO for chemical reactions with an easy-to-use natural language interface, which facilitates access to a broader audience.

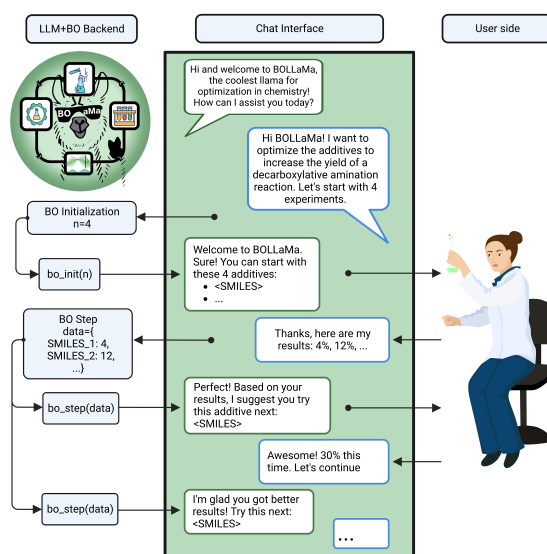


Figure 11. *Schematic overview of BOLLaMa.* A LLM can act as an interface to a BO algorithm. An experimental chemist can bootstrap an optimization and then, via a chat interface, update the state of the simulation to which the bot responds with the recommended next steps.

BOLLaMa combines LLMs with BO algorithms to assist chemical reaction optimization. The user-friendly interface allows even those with limited technical knowledge to engage with the tool. BOLLaMa’s current implementation provides two main tools: the initialization function and the optimization step function [56], that are retrieved on LLM-demand as shown in Figure 11.

The primary contribution of this project is democratizing access to advanced BO techniques in chemistry, promoting widespread adoption of sustainable optimization tools, and impacting sustainability efforts within the community. This approach can be further enhanced to provide a more comprehensive assistant experience, such as with additional recommendations or safety warnings, and improve the explainability of the BO process to foster user trust and informed decision-making.

Key insights gained from this project include the critical role of accessibility in developing expert tools and the potential of LLMs in chemistry through various agent architectures [50]. In addition, the initial BO tool adapted for BOLLaMa was designed for closed-loop automated laboratories, emphasizing the need for accessible tools catering to diverse user backgrounds.

One sentence summaries

a. Problem/Task Giving scientists without coding and machine learning expertise access to Bayesian optimization.

b. Approach LLM as a chat-interface for a Python package for Bayesian optimization by using ReAct-like approach in which the LLM has access to text-description of relevant functions (such as initialization and stepping of the BO run).

c. Results and Impact The chat interface can successfully initialize a BO run and then convert observations reported in natural language into calls to the stepping function of the BO tool.

d. Challenges and Future Work As most LLM agents, the tools suffers from robustness issues and the correct functioning cannot be guaranteed for all possible prompts.

III. Knowledge Extraction

A. InsightGraph

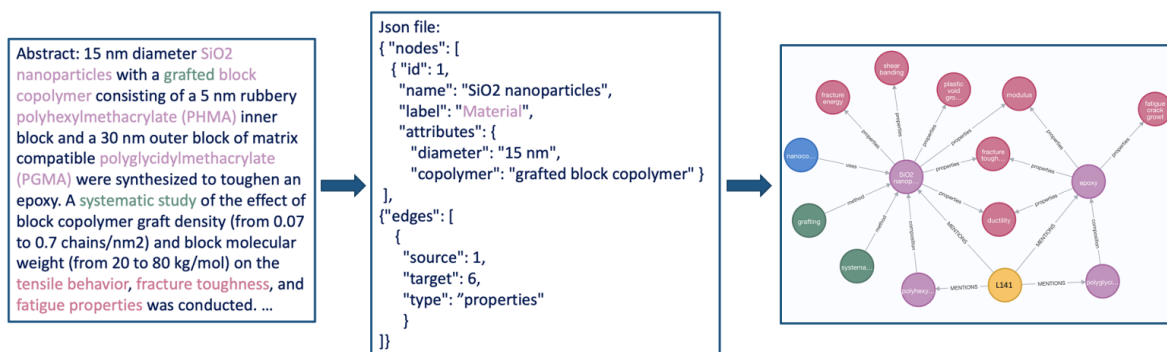


Figure 12. *The Insight Graph interface.* A suitably prompted LLM can create knowledge graph representations of scientific text that can be visualized using tools such as neo4j’s visualization tools. [57]

The traditional method of performing a literature review involves months of reading relevant articles to find crucial information on material properties, structure, reaction pathways, and applications. Knowledge graphs are sources of structured information that enable data visualization, data discovery, insights, and downstream machine-learning tasks. Knowledge graphs extracted from published scientific literature covering broad materials science domains [58] as well as more-focused domains such as polymer nanocomposites [59] empower material scientists to discover new concepts and accelerate research. Until recently, capturing complex and hierarchical relationships for a knowledge graph within the materials science literature was a time-consuming effort, often spanning multi-disciplinary collaborations and many Ph.D. years. By leveraging zero to few-shot training and pre-trained LLMs, it is now possible to rapidly extract complex scientific entities with minimal technical expertise [58, 60, 61]. We envision that knowledge graphs built by LLMs based on scientific publications can offer a concise and visual means to launch a literature review.

To demonstrate a proof of concept of a zero-shot entity and relationship extraction, we identified 200 abstracts on polymer-nanocomposite materials for which detailed structured information was already available [62]. Each abstract was fed as a prompt to GPT-3.5-turbo, a language model powering the popular ChatGPT web application by OpenAI. The instructions in our prompt consisted of an example JSON containing high-level schema and information on possible entities and pairwise relationships. The nodes and relationships in the output JSON response were then stored in a neo4j graph database using Cypher, a graph query language (Figure 12). [57] The zero-shot capabilities of the model allowed the specification of an arbitrary entity and relationship types depending upon the information contained in the text. Given that this required a change in the neo4j pipeline every time the prompt changed, we found it necessary to constrain the JSON schema to a standard format.

While large language models on their own are prone to hallucinations, leveraging them with guidance to create structured databases empowers chemists/materials scientists with no expertise in natural language processing to search and build on existing knowledge leading to new insights. The speed at which LLMs can create structured graphs dramatically exceeds the years required for humans to manually curate data into existing knowledge graphs. Access to structured databases will accelerate the pace of data-driven material science research, synthesizing details embedded in dispersed scientific publications. Additionally, other scientific fields could benefit from a similar use of LLMs to extract entities and relationships to build knowledge graphs.

Owing to the non-deterministic nature of LLMs, we found that the output response would vary even when the same prompt was provided. An instruction constraining the JSON schema minimized the variability. A systematic study comparing different foundation models, prompt techniques (zero-shot, one-shot, few-shot), prompt chaining, and the role of fine-tuning is needed to evaluate the precision and recall of extracted entities

and relationships. Notably, pairwise links between the nodes are not often enough to model the complex nature of materials requiring improvement in the input schema.

One sentence summaries

- a. Problem/Task* Extraction of entities and their relationships from text.
- b. Approach* Prompting of GPT-3.5-turbo prompted with abstract and example JSON and the task to extract entities and their relationships in a structure as provided in the example.
- c. Results and Impact* The approach can successfully create meaningful JSON data structures with extracted entities and their relationships for hundreds of abstracts.
- d. Challenges and Future Work* The non-deterministic behavior of LLMs can lead to variability and fragile behavior. To better understand this as well as the performance of this approach, more systematic benchmarking is needed.

B. Extracting Structured Data from Free-form Organic Synthesis Text

A suspension of **compound 63** (0.22 g, 0.93 mmol) and **64** (0.33 g, 0.92 mmol) in THF/triethylamine (11 mL, 10/1) was stirred at room temperature for 48 h. After this time, a clear solution was formed. The solvent was removed under reduced pressure and the residue was purified by flash chromatography (silica gel, 10:1:0.1 chloroform/methanol/concentrated ammonium hydroxide) to provide the guanidine **65** (0.3 g, 60%) as a yellow solid. ¹H NMR (300 MHz, DMSO-d₆) δ 1.42 (s, 9H), 1.55 (m, 4H), 2.19 (s, 6H), 2.58 (m, 4H), 3.99 (m, 2H), 6.83 (d, 2H), 7.12 (d, 2H), 7.40 (br s, 2H), 9.02 (m, 2H).

```

{
  "m1_m2_m5": {
    "components": [
      {
        "identifiers": [{"type": "NAME", "value": "compound 63"}],
        "amount": {"mass": {"value": 0.22, "units": "GRAM"}},
        "reactionRole": "REACTANT"
      },
      {
        "identifiers": [{"type": "NAME", "value": "compound 64"}],
        "amount": {"mass": {"value": 0.33, "units": "GRAM"}},
        "reactionRole": "REACTANT"
      },
      {
        "identifiers": [{"type": "NAME", "value": "THF"}],
        "amount": {"volume": {"value": 11, "units": "MILLILITER"}},
        "reactionRole": "SOLVENT"
      }
    ]
  },
  "m3": {
    "components": [
      {
        "identifiers": [{"type": "NAME", "value": "triethylamine"}],
        "amount": {"volume": {"value": 11, "units": "MILLILITER"}},
        "reactionRole": "SOLVENT"
      }
    ]
  }
}

```

Figure 13. *The Organic Synthesis Parser interface*. The top part shows text describing an organic reaction (<https://open-reaction-database.org/client/id/ord-1f99b308e17340cb8e0e3080c270fd08>), which the finetuned LLM converts into structured JSON (bottom). A demo application can be found at https://qai222.github.io/LLM_organic_synthesis/.

a. Problem As data-driven approaches and machine learning (ML) techniques gain traction in the field of organic chemistry and its various subfields, it is becoming clear that, as most data in chemistry is represented by unstructured text, the predictive power of these approaches is limited by the lack of structured, well-curated data. Due to the large corpus of organic chemistry literature, manual conversion from unstructured text to structured data is unrealistic, making software tools for this task necessary to improve or enable downstream applications, such as reaction prediction and condition recommendation.

b. Solution In this project, we leverage the power of fine-tuned LLMs to extract reactant information from organic synthesis text to structured data. 350 reaction entries were randomly selected from the Open Reaction Database (ORD) [63]. The field of `reaction.notes.procedure_details` is used as the input (prompt), and the field of `reaction.inputs` is used as the output (completion). 300 of these prompt-completion pairs were used to fine-tune a GPT-3 (OpenAI Davinci) model using the OpenAI command line interface (version 0.27.2), and the rest were used for evaluation. In addition to this, we also explored fine-tuning the Alpaca-LoRA model [16, 64, 65] for this task. All data and scripts used in this project are available in the GitHub repository.

c. Results and Discussion Surprisingly, the pre-trained language model (OpenAI Davinci), fine-tuned with only 300 prompt-completion pairs, is capable of generating valid JSON complying with the ORD data model. For the 50 prompt-completion pairs in evaluation, 93% of the components in reaction inputs were correctly extracted from the free text reaction description by the GPT-3 based model. The model also associates existing properties, such as volume or mass used in the reaction, to these components. In addition to recognizing in-text chemical entities (such as molecule names), as shown in Figure 13, tokens referencing external chemical entities (compound numbers) can also be captured by the model. On the other hand, while completing the prompts with extracted chemical information, the fine-tuned Alpaca-LoRA model was unable to properly construct a valid JSON complying with the ORD data model.

Despite these encouraging preliminary results, there are still challenges to a robust synthesis text parser. One of them is the ambiguous and often artificial boundary between descriptions of reactions and workups, which leads to misplaced chemical entities in the structured data, e.g., a solvent used in the extraction of products is instead labeled as a reaction solvent. The aforementioned external reference problem, where a compound number in the procedure is only explicitly identified in an earlier section of the manuscript, can only be solved by prompting the LLM with multiple paragraphs or even the entire document, adding more irrelevant tokens to the prompt. It is also important to prevent the LLM from “auto-completing” extracted named entities with information outside the prompt, e.g., the chemical is extracted as “sodium chloride” in the completion while it is only specified as “chloride” in the prompt.

One sentence summaries

- d. Problem/Task* Extraction of structured reaction condition and procedure data from text.
- e. Approach* Fine-tuning of LLMs on hundreds of prompt (unstructured text)- completion (extracted structured data) pairs.
- f. Results and Impact* OpenAI’s davinci model can extract the relevant data with a success rate of 93 %.
- g. Challenges and Future Work* Parameter efficient fine-tuning could not match the performance of OpenAI’s models. In addition, there are instances in which the LLM goes beyond the specified tasks (e.g., modifies/“autocompletes”) extracted entries, which can lead to fragile systems.

C. TableToJson: Extracting structured information from tables in scientific papers

Much of the scientific information published in research articles is presented in an unstructured format, primarily as free text, making it a difficult input for computational processing. However, relevant information in scientific literature is not only found in text form. Tables are commonly employed in scientific articles, e.g., to collect precursors and raw materials' characteristics, synthesis conditions, synthesized materials' properties, or chemical process results. Converting this information into a structured data format is usually a manual time-consuming and tedious task. Neural-network-based table extraction methods and optical character recognition (OCR) [66], which can convert typed, handwritten, or printed documents into machine-encoded text, can be used to extract information from tables in PDF files. However, it is often not straightforward to extract the data in the desired structured format. Nonetheless, structured data is essential for creating databases that aggregate research results, and enable data integration, comparison, and analysis.

In this context, JSON is a widely adopted structured data format due to its simplicity, flexibility and compatibility with different programming languages and systems. However, obtaining structured data following a specific JSON schema with models can be challenging. The generated JSON needs to be syntactically correct and conform to a schema that defines the JSON's structure. Models typically do not provide structured output that perfectly matches the desired JSON schema. Some manual post-processing or data transformation is often necessary to map the extracted information to the appropriate schema fields.

In this work, we have studied two approaches to generate structured JSON from data contained in tables of scientific papers focused on different research topics within the field of chemistry [67–73]. The Python `json` module was used to parse JSON data and validate the outputs.

As a first approach, the OpenAI `text-davinci-003` model was used to generate structured JSON from data in tables. The input to the LLM is the HyperText Markup Language (HTML) code of the table, obtained directly from the digital object identifier (DOI) of the article using the Python `selenium` library, while the output of the model is the data extracted in JSON form (Figure 14). The OpenAI `text-curie-001` model, although not tested in this work, can also be utilized if the number of input tokens, considering both the HTML text of the table and the schema, meets the requirements of this model (maximum 2049 input tokens, compared to 4097 for `text-davinci-003`).

The use of the OpenAI model to generate structured JSON was compared with a second approach, i.e., the use of `jsonformer` (<https://github.com/lrgs/jsonformer>), which implements a data processing pipeline that combines the model generation with appropriate data transformation. This method introduces an efficient way to generate structured JSON using LLMs by generating only the content tokens and filling in the fixed tokens. This avoids generating a complete JSON string and parsing it. This approach ensures that the produced JSON is always syntactically correct and aligns with the specified schema. [74]

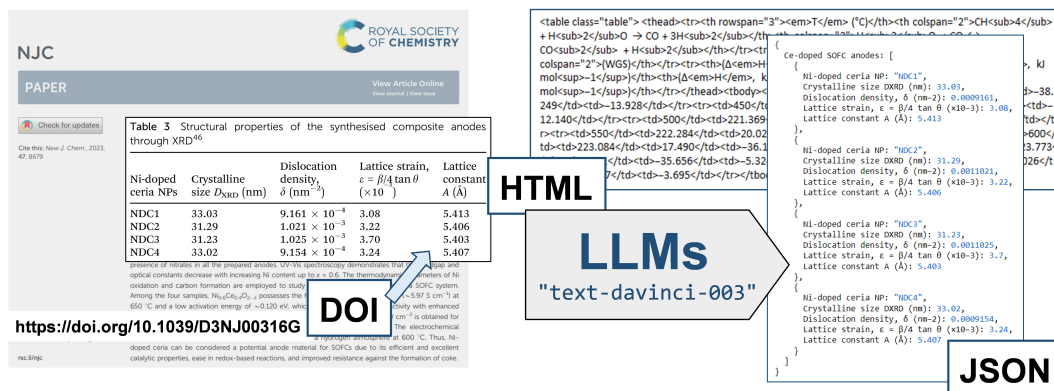


Figure 14. *TableToJson*. Extraction of structured information from scientific data in tables using LLMs. The input to the LLM model is the HTML code of a table contained in a scientific paper. The output of the LLM model is data structured in JSON form. Results can be visualized in this demo app: <https://vgvinter-tabletojson-app-kt5aiv.streamlit.app/>.

In our first approach, we directly asked the OpenAI `text-davinci-003` model to generate a JSON object according to a desired JSON schema provided in the model prompt. The table content was also included in

the prompt as HTML code. The accuracy in the prediction, calculated as the percentage of schema values generated correctly, is shown in Figure 15. In all examples, the OpenAI model was queried with a simple prompt, and it correctly extracted all the data in the table and inserted every value into the corresponding position in the schema, with 100% accuracy, providing as output a JSON object. This model also correctly generated both string and number values according to the type assigned in the schema. However, in two of the examples, the OpenAI model did not generate the JSON object name specified in the schema when the corresponding name was not found in the table, generating only the list of components. This was solved by modifying the object name in the schema to a term that more closely aligned with the content of the table. It appears that when the model could not establish a clear relationship between the provided name and the table content, it disregards that part of the schema during generation. These results indicate that the OpenAI `text-davinci-003` model is able to convert scientific data from tables of research papers to a structured format following the approach used in this work, where the desired JSON schema was included in the model prompt. Nevertheless, the model retains a certain degree of freedom to modify the requested scheme if it considers that something may be wrong.

	text-davinci-003 (schema in prompt)		text-davinci-003 + jsonformer	
	<pre>prompt = "Generate a JSON object extracting the information from this table in html code: " + HTML_table + "Generate the JSON result with the following JSON schema and give only the JSON as output: " + JSON_schema</pre>			
	JSON follows schema	% values extracted ok	JSON follows schema	% values extracted ok
carbon materials for CO ₂ adsorption	✓	100%	✓	100%
MOFs properties	✓	100%	✓	100%
supercapacitor performance	✓	100%	✓	100%
catalysts for CO ₂ conversion	✗ → ✓ a	100%	✓	94% b
biomass properties	✓	100%	✓	100%
anode materials for SOFCs	✗ → ✓ a	100%	✓	80% → 100% c
perovskite cathodes for SOFCs	✓	100% d	✓	46% → 60% → 86-100% d
MOFs properties: providing a wrong schema to the model	✓ (new schema is created following the table)	100%	✓	(values in table are inserted in the provided wrong schema)

a | the OpenAI model did not generate the JSON object name provided in the schema when this name was not included in the table, generating only the list of components; this was solved by using an object name closer to the table contents

b | errors in the generation of compounds formulas due to the "-" character (e.g., "Fe\u2013" and "MnFe\u2013N2" instead of "Fe-PYL" and "MnFe-N2")

c | errors in generating numbers with powers (e.g., 9.161×10^{-4}); this was solved with an explanation in the prompt: "if you find numbers as $1.025 \times 10⁻³$, this means $1.025e^{-3}$ "

d | for this table (below) the OpenAI model generated all values correctly; jsonformer failed to generate molecule names (e.g., "Pr1-xSrxCo1-"), strings with the "-" character (e.g., "129\u2013369d") and numbers with powers (e.g., "5.93\u2013710"); after solving the generation of wrong names the accuracy increased from 46% to 60%, after solving the generation of numbers with powers it increased up to 86%, but the issues with the "-" character could not be solved systematically and the resulting accuracy varied between 86% and 100% for several attempts of JSON generation.

Composition	σ_e (Scm ⁻¹)	σ_t (Scm ⁻¹)	CTE (10 ⁻⁶ K ⁻¹)	References
La _{1-x} Sr _x MnO ₃	130–300	5.93×10^{-7}	11–13	[90]
La _{1-x} Sr _x CoO ₃	1200–1600	0.22	19–20	[91,92]
La _{1-x} Sr _x FeO ₃	129–369	$0.205-5.6 \times 10^{-3}$	12.2–16.3	[[93], [94], [95]]
La _{1-x} Sr _x CoFeO ₃	87–1050	$0.058-8 \times 10^{-3}$	14.8–21.4	[96,97]
Pr _{1-x} Sr _x Co _{1-y} Fe _y O ₃	76–950	$1.5 \times 10^{-3}-4.4 \times 10^{-5}$	12.8–21.3	[95,98]

Figure 15. *TableToJson*. Results of the structured JSON generation of tables contained in scientific articles. Two approaches are compared: (i) the use of an OpenAI model prompted with the desired JSON schema, and (ii) the use of an OpenAI model together with `jsonformer`.

The second approach used to generate structured information was a version of the `jsonformer` approach adapted for use with OpenAI LLMs (<https://github.com/martinezpl/jsonformer/tree/add-openai>), with the implementation of the inclusion of the table text as an input parameter to the `jsonformer` function.

Detection of strings indicating null values was also added when the schema type is number, as “nan”, “NaN”, “NA”, and “NAN” entries are common in research data tables. The OpenAI `text-davinci-003` model was used. In this case, the model was prompted with the desired JSON schema and the HTML code of the studied table. `Jsonformer` reads the keys from the JSON schema and only delegates the generation of the value tokens to the language model, ensuring that a valid JSON is generated by the LLM model.

For this approach, the accuracy in the prediction is also shown in Figure 15. The use of the OpenAI `text-davinci-003` model together with `jsonformer` generated valid JSON objects with 100% accuracy for most of the tables evaluated using a simple prompt. Figure 16 shows the results of one of the examples studied, where using a simple descriptive prompt denoting the type of input text, this approach correctly generated structured data JSON from a table with a complex header. However, it was detected that when the values to be generated contain special characters or specific texts, a more detailed prompt with some simple examples, but without finetuning, can be necessary to provide good results, as shown in Figure 17 for a special numeric notation that included power numbers.

Sample	Ultimate analysis					Proximate analysis				HHV		H/O	He density
	C	N	H	S	O ₂	MC	Ash	VM	FC ₂	Empty Cell	Empty Cell	ratio	(g/cm ³) ^ρ
AS	49.44	0.31	5.85	0.05	42.90	6.5	1.45	78.9	19.6	19.565	2.16	1.252	
CHE	50.22	0.34	5.55	0.01	43.41	8.4	0.47	81.2	18.3	19.109	2.03	1.268	
CHET	51.30	0.40	5.40	0.02	42.59	8.2	0.29	80.0	19.7	19.588	2.01	1.275	
CS	47.96	2.74	5.93	0.21	35.26	6.7	7.90	70.4	21.7	19.067	2.67	1.156	
GP	45.50	1.82	5.05	0.17	34.73	11.6	12.73	67.6	19.7	18.682	2.31	1.210	
OS	51.21	0.29	6.01	0.03	41.88	4.3	0.58	81.5	17.9	20.511	2.28	1.241	
PCL	52.89	0.44	6.06	0.03	39.46	10.1	1.12	76.5	22.4	20.976	2.44	1.237	

prompt="Generate an object with the following schema extracting the information from the provided table in html code:"

```
{
  Biomass type: [
    {
      Sample: "AS",
      Ultimate Analysis (wt%, db): {
        C: 49.44,
        N: 0.31,
        H: 5.85,
        S: 0.05,
        O: 42.9
      },
      Proximate Analysis (wt%, db): {
        MC (wt%): 6.5,
        Ash: 1.45,
        VM: 78.9,
        FC: 19.6
      },
      HHV (MJ/kg, db): 19.565,
      H/O: 2.16,
      He density (g/cm3): 1.252
    },
    {
      Sample: "CHE",
      Ultimate Analysis (wt%, db): {
        C: 51.3,
        N: 0.4,
        H: 5.4,
        S: 0.02,
        O: 42.59
      },
      Proximate Analysis (wt%, db): {
        MC (wt%): 8.2,
        Ash: 0.29,
        VM: 80.0,
        FC: 19.7
      },
      HHV (MJ/kg, db): 19.588,
      H/O: 2.01,
      He density (g/cm3): 1.275
    },
    {
      Sample: "CS",
      Ultimate Analysis (wt%, db): {
        C: 47.96,
        N: 2.74,
        H: 5.93,
        S: 0.21,
        O: 35.26
      },
      Proximate Analysis (wt%, db): {
        MC (wt%): 6.7,
        Ash: 7.9,
        VM: 70.4,
        FC: 21.7
      },
      HHV (MJ/kg, db): 19.067,
      H/O: 2.67,
      He density (g/cm3): 1.156
    }
  ]
}
```

Figure 16. *TableToJson*. Structured JSON generation of tables contained in scientific articles using a prompt with a simple description of the type of input text. One example is shown for a table that contains data on properties of biomass materials [71].

As shown in Figure 15, in one of these examples, an accuracy of 94% was obtained from a table containing a few catalyst names that included the “-” character, and those values were erroneously generated. In another example, an accuracy of 80% was initially obtained due to errors in the generation of numbers with powers (e.g., 9.161×10^4), which could be solved by adding an explanation in the prompt: “if you find numbers as 1.025×10^{³} , this means $1.025e^{-3}$ ”, increasing the accuracy to 100%.

Next, a table with more complex content (long molecule names, hyphens, power numbers, subscripts, and superscripts...) was selected (Figure 15), resulting in an accuracy of 46% in the JSON generation, meaning that only 46% of the schema values were correctly generated. The erroneous generation of long formula or molecule names with a mixture of letters and numbers as subscripts could be solved by increasing the value of the `max_string_token_length` argument of the `jsonformer` function to get a longer response where the end of the string can be detected more easily, which increased the accuracy to 60%. `Jsonformer` also showed some issues in this example in generating power numbers, which are represented as $10^{⁻ⁿ}$ in the input HTML text. As mentioned above, this was solved by adding a specific explanation in the prompt, increasing the accuracy to 86%. A specific explanation was also included in the prompt to address the issues related to the presence of hyphens in the text. Still, this problem could not be solved systematically, and the resulting accuracy varied between 86% and 100% for several JSON generation attempts. In this particular case, the generated value provided by the model included Unicode text instead of the “-” character (and usually several “\” characters). An instruction to “decode Unicode characters in your response”

was then included in the prompt. Although this solution sometimes yielded satisfactory results, it did not systematically guarantee correct output. These results indicate that the OpenAI model combined with `jsonformer` can provide wrong outputs when the values to be generated contain some special characters, such as the “_” character in this example. This issue requires further investigation to be improved.

Dislocation density, δ (nm ⁻²)	prompt="Generate an object with the following schema extracting the information from the provided table in html code:"	prompt="Generate an object with the following schema extracting the information from the provided table in html code (if you find numbers as $1.025 \times 10^{⁻³}$, this means $1.025e-3$):"
9.161×10^{-4}	<pre>{ "Ce-doped SOFC anodes": [{ "Ni-doped ceria NP": "NDC1", "Crystalline size DXR0 (nm)": 33.03, "Dislocation density, δ (nm⁻²): 9.161, "Lattice strain, $\epsilon = \beta/4 \tan \theta$ (x10⁻³): 3.08, "Lattice constant A (Å)": 5.413 }, { "Ni-doped ceria NP": "NDC2", "Crystalline size DXR0 (nm)": 31.29, "Dislocation density, δ (nm⁻²): 1.021, "Lattice strain, $\epsilon = \beta/4 \tan \theta$ (x10⁻³): 3.22, "Lattice constant A (Å)": 5.406 }, { "Ni-doped ceria NP": "NDC3", "Crystalline size DXR0 (nm)": 31.23, "Dislocation density, δ (nm⁻²): 1.025, "Lattice strain, $\epsilon = \beta/4 \tan \theta$ (x10⁻³): 3.7, "Lattice constant A (Å)": 5.403 }, { "Ni-doped ceria NP": "NDC4", "Crystalline size DXR0 (nm)": 33.03, "Dislocation density, δ (nm⁻²): 9.154, "Lattice strain, $\epsilon = \beta/4 \tan \theta$ (x10⁻³): 3.24, "Lattice constant A (Å)": 5.407 }] }</pre>	<pre>{ "Ce-doped SOFC anodes": [{ "Ni-doped ceria NP": "NDC1", "Crystalline size DXR0 (nm)": 33.03, "Dislocation density, δ (nm⁻²): 0.0009161, "Lattice strain, $\epsilon = \beta/4 \tan \theta$ (x10⁻³): 3.08, "Lattice constant A (Å)": 5.413 }, { "Ni-doped ceria NP": "NDC2", "Crystalline size DXR0 (nm)": 31.29, "Dislocation density, δ (nm⁻²): 0.0011021, "Lattice strain, $\epsilon = \beta/4 \tan \theta$ (x10⁻³): 3.22, "Lattice constant A (Å)": 5.406 }, { "Ni-doped ceria NP": "NDC3", "Crystalline size DXR0 (nm)": 31.23, "Dislocation density, δ (nm⁻²): 0.0011025, "Lattice strain, $\epsilon = \beta/4 \tan \theta$ (x10⁻³): 3.7, "Lattice constant A (Å)": 5.403 }, { "Ni-doped ceria NP": "NDC4", "Crystalline size DXR0 (nm)": 33.03, "Dislocation density, δ (nm⁻²): 0.0009154, "Lattice strain, $\epsilon = \beta/4 \tan \theta$ (x10⁻³): 3.24, "Lattice constant A (Å)": 5.407 }] }</pre>

Figure 17. *TableToJson*. Structured JSON generation of a table contained in a scientific article using a standard prompt and a prompt with a few simple examples of the special numeric notation found in some of the cells of the input table [72].

Lastly, for one of the examples, a test was performed by providing a wrong schema to the model (Figure 15). In this case, as expected, `jsonformer` inserted the values contained in the table into the given wrong schema in a more or less ordered fashion, generating an invalid output. However, the OpenAI model created a new schema according to the table structure and headers, providing a valid result, and confirming its freedom to decide what may be wrong with the user’s query. An example of these results is shown in Figure 18.

The two approaches used in this work showed a good performance in the generation of JSON format when the data contained in the table are regular strings or numbers, with an accuracy of 100% in most of the examples. The results of this work show that, although the OpenAI `text-davinci-003` is able to easily extract structured information from tables and give a valid JSON output, this approach cannot guarantee that the outputs will always follow a specific schema. On the other hand, although `jsonformer` may present problems when special characters need to be generated, some of these issues have been solved with careful prompting, and others could probably be solved with further research. It can be concluded that `jsonformer` can be a powerful tool for the generation of structured data from unstructured information in most tables, ensuring the generation of valid JSON syntax as the output of LLMs that always complies with the provided schema. The use of `jsonformer` could facilitate and promote the creation of databases and datasets for numerous topics within the field of chemistry, especially in experimental domains, where the availability of structured data is very scarce.

One sentence summaries

- a. *Problem/Task* Extracting structured data in a JSON-schema-compliant form from HTML tables.

table from the "MOFs properties" example							
MOFs	BET (m ² /g)		V _{pore} (cc/g)			ρ (g/cm ³)	metal (mmol/cc)
	N ₂	calcd	N ₂	CH ₄	calcd		
HKUST-1	1850	2064	0.78	0.78	0.78	0.883	4.38
Ni-MOF-74	1350	1240	0.51	0.52	0.49	1.206	7.74

schema from the "catalysts CO ₂ " example	text-davinci-003 + jsonformer	text-davinci-003
<pre> json_schema_catalystCO2 = { "type": "object", "properties": { "Carbon-supported catalysts CO2": { "type": "array", "items": { "type": "object", "properties": { "catalyst": {"type": "string"}, "Synthesis method": {"type": "string"}, "Promoter incorporation": {"type": "string"}, "X CO2 (%)": {"type": "number"}, "S CO (%)": {"type": "number"}, "S C1 (%)": {"type": "number"}, "S C2-C4 (%)": {"type": "number"}, "S C5+ (%)": {"type": "number"}, "Temperature (°C)": {"type": "number"}, "P (MPa)": {"type": "number"}, } } } } } </pre>	<pre> { Carbon-supported catalysts CO2: [{ catalyst: "HKUST-1", Synthesis method: "N2", Promoter incorporation: "calcd", X CO2 (%): 1850, S CO (%): 2064, S C1 (%): 0.78, S C2-C4 (%): 0.78, S C5+ (%): 0.78, Temperature (°C): 0.883, P (MPa): 4.38 }, { catalyst: "Ni-MOF-74", Synthesis method: "N2", Promoter incorporation: "calcd", X CO2 (%): 1350, S CO (%): 1240, S C1 (%): 0.51, S C2-C4 (%): 0.52, S C5+ (%): 0.49, Temperature (°C): 1.206, P (MPa): 7.74 }] } </pre>	<pre> { MOFs: "HKUST-1", N2: 1850, calcd: 2064, VporeN2: 0.78, VporeCH4: 0.78, calcdVpore: 0.78, pg/cm3: 0.883, metalmmol/cc: 4.38 }, { MOFs: "Ni-MOF-74", N2: 1350, calcd: 1240, VporeN2: 0.51, VporeCH4: 0.52, calcdVpore: 0.49, pg/cm3: 1.206, metalmmol/cc: 7.74 } </pre>

Figure 18. *TableToJson*. Results of the structured JSON generation of a table after providing the model with a wrong schema. The output generated using the OpenAI model together with `jsonformer` is shown on the left (values in the table are inserted in the provided wrong schema), while the output generated using directly the OpenAI model is shown on the right (a new schema is created following the table content).

b. Approach Two approaches were compared: Direct prompting of OpenAI's `text-davinci-003` model with the input table and the JSON schema, as well as the `Jsonformer` approach, which only samples from a subset of tokens in field-wise generation steps.

c. Results and Impact Both approaches can extract data in schema-compliant from tables with high success rates. Due to hard-coded decoding rules, `Jsonformer` failed in some cases.

d. Challenges and Future Work While the `Jsonformer` approach can guarantee valid syntax, it can fail in cases that were not considered in the development of the decoding rules. Hence, future work is needed for increasing the general applicability of constrained decoding strategies.

D. AbstractToTitle & TitleToAbstract: text summarization and text generation

1. Problem

Text summarization and text generation are some of most the common tasks in natural language processing (NLP). Often it is tricky to obtain well-defined and curated datasets for these tasks. Also, evaluating the performance of an NLP model is challenging because there is no unique way to summarize and generate text. Luckily, there are many publicly available manuscripts for chemistry and materials science in open access platforms such as arXiv and PubChem. These datasets can be used along with LLMs to solve problems such as: 1) given title of the article; generate an abstract, 2) given an abstract; generate a title, which should be similar to the one available in the literature. Such models can greatly help in technical writing, especially with a human expert in the loop.

2. Solution

The above challenging tasks require an organized infrastructure for curated data and tools. JARVIS-ChemNLP [75] is an integrated library for materials chemistry text data for NLP tasks such as text classification, token classification, abstractive summarization, text generation, and integrating with existing DFT databases. ChemNLP uses transformer models to generate titles from abstracts and vice versa for arXiv condensed matter category articles.

There is no unique approach for both tasks. Nevertheless, highly qualified scientists and engineers generate such abstracts and titles and can act as a valuable benchmark to evaluate model performance. Moreover, developing metrics for text generation is a difficult task. Here, for the sake of simplicity, we choose the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metrics for evaluating the model performance.

ChemNLP uses models such as a pre-trained Text-to-Text Transfer Transformer (T5) developed by Google for text summarization. The pre-trained model is further fine-tuned to produce summaries of abstracts and analyze the performance by analyzing its closeness to the title of the given paper.

For the text generation, ChemNLP fine-tunes LLMs such as the pre-trained GPT-2-medium model from the HuggingFace library.

For these tasks, the community can contribute their models on a benchmarking platform called the JARVIS-Leaderboard (https://pages.nist.gov/jarvis_leaderboard/). Specifically, the TextGen task is available at: https://pages.nist.gov/jarvis_leaderboard/AI/TextSummary/arxiv_summary_text/ and TextSummary task is available at https://pages.nist.gov/jarvis_leaderboard/AI/TextGen/arxiv_gen_text/

3. Example

Title to Abstract example for `cond-mat/0105402`. [76]

Question

MgB2 under pressure: phonon calculations, Raman spectroscopy, and optical reflectance can be described as

Fine-tuned GPT-2-medium model output

The amplitude of phonon vibrations in a material exhibiting an unconventional lattice structure has been measured in the low pressure region of the granular MgB2 phase. The peak in the high pressure region is the first measurement of phonon amplitude ever presented in a granular superconducting material, and it indicates a potential source of superconductivity. Superconducting Transition from Fluctuations to Brillouin T-Wings in CeFo Josephson Junctions can be described as We report electronic and numerical calculations for solving the electron-wave (E-W) conduction spectra of a quasi-two-dimensional superconducting monolith CeFo Josephson junctions: The amplitude of

phonon vibrations in the low pressure region (KPSM), and in the high pressure region of the superconducting density wave (SDW) densities, are used to understand the superconductivity.

Actual abstract

The effect of pressure on optical phonon frequencies of MgB₂ has been calculated using the frozen-phonon approach based on a pseudopotential method. Gruneisen parameters of the harmonic mode frequencies are reported for the high-frequency zone-center E_{2g} and B_{1g} and the zone-boundary E_{2u} and B_{2u} modes at A . Anharmonic effects of phonon frequencies and the implications of the calculated phonon frequency shifts for the pressure dependence of the superconducting transition temperature of MgB₂ are discussed. Also reported are Raman and optical reflectance spectra of MgB₂ measured at high pressures. The experimental observations in combination with calculated results indicate that broad spectral features we observed in the Raman spectra at frequencies between 500 and 900 cm⁻¹ cannot be attributed to first-order scattering by zone-center modes, but originate in part from a chemical species other than MgB₂ at the sample surface and in part from a maximum in the MgB₂ phonon density of states. Low-temperature Raman spectra taken at ambient pressure showed increased scattering intensity in the region below 300 cm⁻¹.

Interestingly, the generated abstract contains grammatically and syntactically incorrect sentences. We suspect that this is due to our use of a small, outdated, base model. However, more systematic analysis will need to be performed in future work.

One sentence summaries

- a. Problem/Task* Text summarization and generation, in specific, a summary of an abstract into a title and generation of an abstract conditioned on a title.
- b. Approach* Fine-tuning of transformer models such as T-5 and GPT-2 on data from arXiv.
- c. Results and Impact* Initial exploration indicates that transformer models might be suitable for this task.
- d. Challenges and Future Work* More systematic analysis, including rating of the generated titles and abstracts by domain experts is required to identify the limitations of this approach.

IV. Education

A. i-Digest

a. Problem Over the last few years, especially during the Covid period, most of us had to switch to the online mode of working in our day-to-day jobs. And even today, the online mode of working has, to some extent, stayed on as it turned out to be convenient for both employers and employees. One clear example can be found in the field of education, where the use of video lectures became the norm for teaching students in universities and schools. Likewise, podcasts and three-minute thesis videos, which communicate important scientific information to society at large, have grown tremendously [77, 78]. This has led to a situation where, at present, we have an enormous amount of important scientific information stored in the form of videos and audio all over the internet. A current challenge is to summarize and make use of this knowledge efficiently. Some efforts in this direction have been made by using AI Youtube summarizers and QnA Bots [79]. We would like to build upon such efforts and create a tool for the field of education.

b. Solution We present a tool that self-guides students and other users toward a better understanding of the content of a video lecture or a podcast. In order to accomplish this, we used publicly available LLMs like Open AI’s Whisper [80] and GPT-3.5-turbo model. All the user needs to do is provide a link to the lecture video or audio file. After only a short time, the overview page shows some technical keywords on which the video is based, a short but comprehensive summary, and some questions for the user to assess his or her understanding of the concepts discussed in the video/audio (Figure 19). Additionally, for chemistry enthusiasts, if some chemical elements/molecules are discussed in the content, we link them to online databases. At the backend, we first convert the video to audio using Pytube (In the case of a podcast, this step is not needed). Then we use the Whisper model to transcribe the audio to text. Next, we make use of the OpenAI GPT-3.5-turbo model to obtain a short summary and a set of questions based on the text. Finally, we extract the name of chemical elements/molecules and list the PubChem database entry for that element/molecule on the overview page. [81–83] The web interface was made using the open-source app framework Streamlit [84].

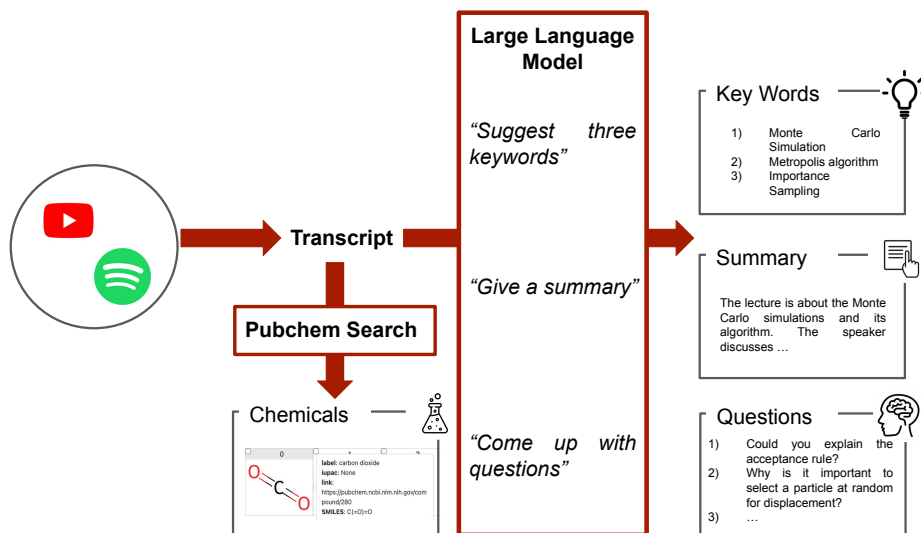


Figure 19. A schematic of the i-digest interface. On providing a link to an online video or audio, i-digest generates some technical keywords, a short but comprehensive summary, and a list of questions based on the content in the video/audio. Additionally, chemicals discussed in the content are linked to online databases such as PubChem.

c. Impact We strongly believe that extracting important scientific information in terms of short lecture notes and questions would help to push forward the field of education towards creating and using resources more efficiently. Moreover, by providing additional links to resources, e.g., databases, journals, and books,

we provide an opportunity for the user to go beyond the content of the lecture and spark interest in a more detailed understanding of the topic. Specifically, this would help researchers/teachers/professors to create new course content or to update/modify already available content. In general, our tool covers a broad range of users, from the youngest learner to the chemistry novice who wants to kickstart his research, all the way to professors, course creators, and lifetime learners.

d. Lessons learned Working together with colleagues can be fun and enriching and often help to solve big problems. This hackathon taught us that even in one day, coming together can help achieve something significant.

One sentence summaries

e. Problem/Task Provide students with automatically generated active learning tasks for lecture recordings.

f. Approach Transcription of videos using OpenAI's Whisper model, prompting of OpenAI's GPT-3.5-turbo model to produce a short summary and questions based on the transcript, as well as to extract mentions of chemicals in the text.

g. Results and Impact The system can transcribe the text, generate meaningful questions, and successfully extract mentions of chemicals.

h. Challenges and Future Work It is difficult to systematically evaluate the performance of this system due to the lack of suitable benchmarks/eval. An obvious extension of this approach is to condition it on further material (e.g., lecture notes and books). In addition, one might automatically score the answers and show them at the beginning and at the end of the video. This would allow us to evaluate the learning of the students and to guide them to the relevant material in case a question was not answered correctly.

V. Meta analysis of the workshop contributions

We have a female/male ratio of about 30 % among the workshop participants who co-authored this paper. We have participants from 22 different institutions in 8 countries.

Most teams combine expertise from different institutions (Figure 21), in several cases beyond academia (Figure 22). Around 20% of the teams are international, with participants from two countries (Figure 23).

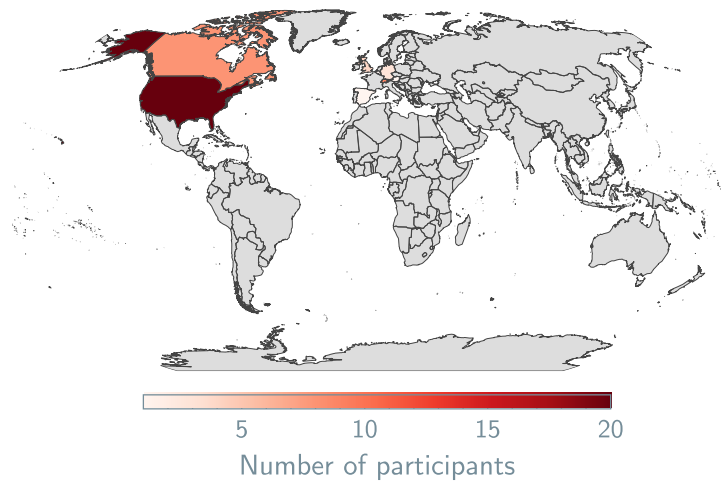


Figure 20. Worldmap (Robin projection) with the number of participants shown in color.

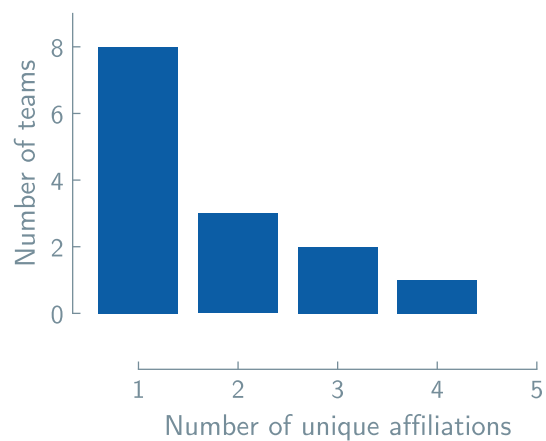


Figure 21. Histogram of the number of unique affiliations per team.

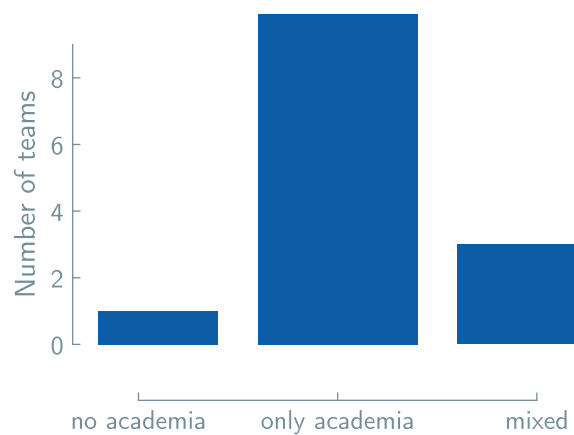


Figure 22. Number of teams with participants only from academia or academia and industry/nonprofit, respectively. We counted national labs as “academia”.

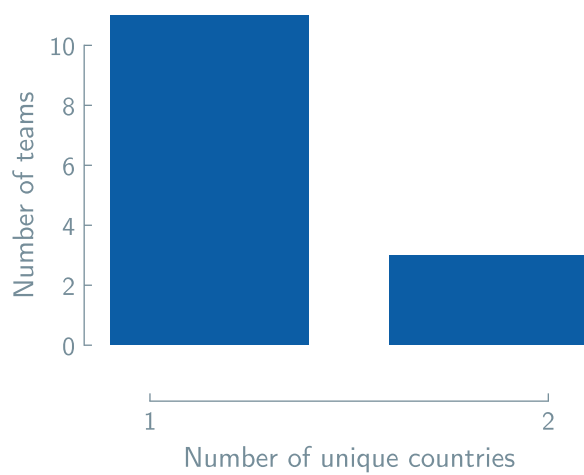


Figure 23. Histogram of the number of unique countries per team.

-
- [1] Ward, L.; Blaiszik, B.; Foster, I.; Assary, R. S.; Narayanan, B.; Curtiss, L. Machine learning prediction of accurate atomization energies of organic molecules from low-fidelity quantum chemical calculations. *MRS Commun.* **2019**, *9*, 891–899.
- [2] Curtiss, L. A.; Redfern, P. C.; Raghavachari, K. Gaussian-4 theory using reduced order perturbation theory. *J. Chem. Phys.* **2007**, *127*, 124105.
- [3] Ramakrishnan, R.; Dral, P. O.; Rupp, M.; Von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Sci. Data* **2014**, *1*, 1–7.
- [4] Narayanan, B.; Redfern, P. C.; Assary, R. S.; Curtiss, L. A. Accurate quantum chemical energies for 133000 organic molecules. *Chem. Sci.* **2019**, *10*, 7449–7455.
- [5] Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31–36.
- [6] Krenn, M.; Häse, F.; Nigam, A.; Friederich, P.; Aspuru-Guzik, A. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Mach. Learn.: Sci. Technol.* **2020**, *1*, 045024.
- [7] Krenn, M.; Ai, Q.; Barthel, S.; Carson, N.; Frei, A.; Frey, N. C.; Friederich, P.; Gaudin, T.; Gayle, A. A.; Jablonka, K. M., et al. SELFIES and the future of molecular string representations. *Patterns* **2022**, *3*, 100588.
- [8] Jablonka, K. M.; Schwaller, P.; Ortega-Guerrero, A.; Smit, B. Is GPT-3 all you need for low-data discovery in chemistry? *ChemRxiv preprint 10.26434/chemrxiv-2023-fw8n4* **2023**,
- [9] Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems* **2020**, *33*, 1877–1901.
- [10] Ramakrishnan, R.; Dral, P. O.; Rupp, M.; Von Lilienfeld, O. A. Big data meets quantum chemistry approximations: the Δ -machine learning approach. *J. Chem. Theory Comput.* **2015**, *11*, 2087–2096.
- [11] Gupta, A. K.; Raghavachari, K. Three-Dimensional Convolutional Neural Networks Utilizing Molecular Topological Features for Accurate Atomization Energy Predictions. *J. Chem. Theory Comput.* **2022**, *18*, 2132–2143.
- [12] Mangrulkar, S.; Gugger, S.; Debut, L.; Belkada, Y.; Paul, S. PEFT: State-of-the-art Parameter-Efficient Fine-Tuning methods. <https://github.com/huggingface/peft>, 2022.
- [13] Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv preprint: Arxiv-2106.09685* **2021**,
- [14] Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language Models are Unsupervised Multitask Learners. **2019**, https://d4mucfpxsywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.
- [15] Kojima, T.; Gu, S. S.; Reid, M.; Matsuo, Y.; Iwasawa, Y. Large Language Models are Zero-Shot Reasoners. **2023**.
- [16] Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint:2302.13971* **2023**,
- [17] Lin, Z.; Akin, H.; Rao, R.; Hie, B.; Zhu, Z.; Lu, W.; Smetanin, N.; Verkuil, R.; Kabeli, O.; Shmueli, Y., et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **2023**, *379*, 1123–1130.
- [18] Andrew, R. Global Co2 Emissions From Cement Production. 2017; <https://zenodo.org/record/831455>.
- [19] Lookman, T.; Balachandran, P. V.; Xue, D.; Yuan, R. Active learning in materials science with emphasis on adaptive sampling using uncertainties for targeted design. *npj Comput. Mater.* **2019**, *5*.
- [20] Völker, C.; Firdous, R.; Stephan, D.; Kruschwitz, S. Sequential learning to accelerate discovery of alkali-activated binders. *Journal of Materials Science* **2021**, *56*, 15859–15881.
- [21] Völker, C.; Benjami Moreno Torres; Tehseen Rug; Firdous, R.; Ghezal Ahmad; Zia, J.; Lüders, S.; Scaffino, H. L.; Höpler, M.; Böhmer, F.; Pfaff, M.; Stephan, D.; Kruschwitz, S. Green building materials: a new frontier in data-driven sustainable concrete design. Preprint 10.13140/RG.2.2.29079.85925. **2023**.
- [22] Ramos, M. C.; Michtavy, S. S.; Porosoff, M. D.; White, A. D. Bayesian Optimization of Catalysts With In-context Learning. *arXiv preprint: Arxiv-2304.05341* **2023**,
- [23] Rao, G. M.; Rao, T. D. G. A quantitative method of approach in designing the mix proportions of fly ash and GGBS-based geopolymer concrete. *Aust. J. Civ. Eng.* **2018**, *16*, 53–63.
- [24] OpenAI, Text-davinci-003. <https://platform.openai.com/models/text-davinci-003>.
- [25] Bousquet, A. lolopy. <https://pypi.org/project/lolopy/>, 2017; Accessed: 2023-02-27.
- [26] Heinisch, O. Steel, R. G. D., and J. H. Torrie: Principles and Procedures of Statistics. (With special Reference to the Biological Sciences.) McGraw-Hill Book Company, New York, Toronto, London 1960, 481 S., 15 Abb. 81 s 6 d. *Biometrische Zeitschrift* **1962**, *4*, 207–208.
- [27] Dinh, T.; Zeng, Y.; Zhang, R.; Lin, Z.; Gira, M.; Rajput, S.; Sohn, J.-Y.; Papailiopoulos, D.; Lee, K. LIFT: Language-Interfaced Fine-Tuning for Non-Language Machine Learning Tasks. *arXiv preprint: Arxiv-2206.06565*. **2022**.

- [28] Herhold, P.; Farnworth, E. The Net-Zero Challenge: Fast-Forward to Decisive Climate Action. World Economic Forum, available at: https://www3.weforum.org/docs/WEF_The_Net_Zero_Challenge.pdf (accessed 4 October 2021). 2020.
- [29] Hong, Z.; Ajith, A.; Pauloski, G.; Duede, E.; Malamud, C.; Magoulas, R.; Chard, K.; Foster, I. ScholarBERT: Bigger is Not Always Better. arXiv preprint: Arxiv-2205.11342. 2022.
- [30] Kim, S.; Thiessen, P. A.; Bolton, E. E.; Chen, J.; Fu, G.; Gindulyte, A.; Han, L.; He, J.; He, S.; Shoemaker, B. A., et al. PubChem substance and compound databases. *Nucleic acids research* **2016**, *44*, D1202–D1213.
- [31] Dai, H. et al. AugGPT: Leveraging ChatGPT for Text Data Augmentation. arXiv preprint: Arxiv-2302.13007. 2023.
- [32] Wolf, T. et al. Transformers: State-of-the-Art Natural Language Processing. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Online, 2020; pp 38–45.
- [33] Pedregosa, F. et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.
- [34] Rajpurkar, P.; Jia, R.; Liang, P. Know What You Don’t Know: Unanswerable Questions for SQuAD. **2018**,
- [35] Zhang, J.; Chang, W.-C.; Yu, H.-F.; Dhillon, I. Fast multi-resolution transformer fine-tuning for extreme multi-label text classification. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 7267–7280.
- [36] White, A. D.; Hocky, G. M.; Gandhi, H. A.; Ansari, M.; Cox, S.; Wellawatte, G. P.; Sasmal, S.; Yang, Z.; Liu, K.; Singh, Y., et al. Assessment of chemistry knowledge in large language models that generate code. *Digital Discovery* **2023**,
- [37] Schwaller, P.; Laino, T.; Gaudin, T.; Bolgar, P.; Hunter, C. A.; Bekas, C.; Lee, A. A. Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction. *ACS Central Science* **2019**, *5*, 1572–1583.
- [38] Schwabe, T.; Grimme, S. Theoretical thermodynamics for large molecules: walking the thin line between accuracy and computational cost. *Acc. Chem. Res.* **2008**, *41*, 569–579.
- [39] Skyner, R. E.; McDonagh, J. L.; Groom, C. R.; van Mourik, T.; Mitchell, J. B. O. A review of methods for the calculation of solution free energies and the modelling of systems in solution. *Phys. Chem. Chem. Phys.* **2015**, *17*, 6174–6191.
- [40] Schleder, G. R.; Padilha, A. C. M.; Acosta, C. M.; Costa, M.; Fazzio, A. From DFT to machine learning: recent approaches to materials science—a review. *J. Phys. Mater.* **2019**, *2*, 032001.
- [41] Chase, H. LangChain. 2022; <https://github.com/hwchase17/langchain>.
- [42] Bran, A. M.; Cox, S.; White, A. D.; Schwaller, P. ChemCrow: Augmenting large-language models with chemistry tools. *arXiv preprint: Arxiv-2304.05376* **2023**,
- [43] Jain, A.; Ong, S. P.; Hautier, G.; Chen, W.; Richards, W. D.; Dacek, S.; Cholia, S.; Gunter, D.; Skinner, D.; Ceder, G.; Persson, K. A. Commentary: The Materials Project: A materials genome approach to accelerating materials innovation. *APL Materials* **2013**, *1*, 011002.
- [44] McDermott, M. J.; Dwaraknath, S. S.; Persson, K. A. A Graph-Based Network for Predicting Chemical Reaction Pathways in Solid-State Materials Synthesis. *Nat. Commun.* **2021**, *12*, 3097.
- [45] Shao, Z.; Gong, Y.; Shen, Y.; Huang, M.; Duan, N.; Chen, W. Synthetic Prompting: Generating Chain-of-Thought Demonstrations for Large Language Models. **2023**,
- [46] Gao, L.; Schulman, J.; Hilton, J. Scaling Laws for Reward Model Overoptimization. *ARXIV.ORG* **2022**,
- [47] Rego, N.; Koes, D. 3Dmol.js: molecular visualization with WebGL. *Bioinformatics* **2014**, *31*, 1322–1324.
- [48] Schrödinger, L.; DeLano, W. PyMOL. <http://www.pymol.org/pymol>.
- [49] Sehnal, D.; Bittrich, S.; Deshpande, M.; Svobodová, R.; Berka, K.; Bazgier, V.; Velankar, S.; Burley, S. K.; Koča, J.; Rose, A. S. Mol* Viewer: modern web app for 3D visualization and analysis of large biomolecular structures. *Nucleic Acids Res.* **2021**, *49*, W431–W437.
- [50] Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; Cao, Y. ReAct: Synergizing Reasoning and Acting in Language Models. *arXiv preprint: Arxiv-2210.03629* **2023**,
- [51] Thompson, A. P.; Aktulga, H. M.; Berger, R.; Bolintineanu, D. S.; Brown, W. M.; Crozier, P. S.; in ’t Veld, P. J.; Kohlmeyer, A.; Moore, S. G.; Nguyen, T. D.; Shan, R.; Stevens, M. J.; Tranchida, J.; Trott, C.; Plimpton, S. J. LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Comp. Phys. Comm.* **2022**, *271*, 108171.
- [52] Abraham, M. J.; Murtola, T.; Schulz, R.; Páll, S.; Smith, J. C.; Hess, B.; Lindahl, E. GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX* **2015**, *1-2*, 19–25.
- [53] Volk, A. A.; Epps, R. W.; Yonemoto, D. T.; Masters, B. S.; Castellano, F. N.; Reyes, K. G.; Abolhasani, M. AlphaFlow: autonomous discovery and optimization of multi-step chemistry using a self-driven fluidic lab guided by reinforcement learning. *Nat. Commun.* **2023**, *14*, 1403.
- [54] Griffiths, R.-R. et al. GAUCHE: A Library for Gaussian Processes in Chemistry. 2022; <http://arxiv.org/abs/2212.04450>, arXiv:2212.04450 [cond-mat, physics:physics].
- [55] Shields, B. J.; Stevens, J.; Li, J.; Parasram, M.; Damani, F.; Alvarado, J. I. M.; Janey, J. M.; Adams, R. P.; Doyle, A. G. Bayesian reaction optimization as a tool for chemical synthesis. *Nature* **2021**, *590*, 89–96.

- [56] Ranković, B.; Griffiths, R.-R.; Moss, H. B.; Schwaller, P. Bayesian optimisation for additive screening and yield improvements in chemical reactions – beyond one-hot encodings. ChemRxiv preprint 10.26434/chemrxiv-2022-nll2j. 2022.
- [57] Neo4j, Neo4j - The World's Leading Graph Database. 2012; <http://neo4j.org/>.
- [58] Venugopal, V.; Pai, S.; Olivetti, E. MatKG: The Largest Knowledge Graph in Materials Science—Entities, Relations, and Link Prediction through Graph Representation Learning. *arXiv preprint:2210.17340* **2022**,
- [59] McCusker, J. P.; Deagen, M.; Fateye, T.; Wallace, A.; Rashid, S. M.; McGuinness, D. L. Creating and Visualizing the Materials Science Knowledge Graph with Whyis. ISWC (Posters/Demos/Industry). 2021.
- [60] Dunn, A.; Dagdelen, J.; Walker, N.; Lee, S.; Rosen, A. S.; Ceder, G.; Persson, K. A.; Jain, A. Structured information extraction from complex scientific text with fine-tuned large language models. *arXiv preprint: Arxiv-2212.05238* **2022**,
- [61] Badhwar, S. Smart Manufacturing - A Case for Creating a Knowledge Network Using Data Mining. 2022.
- [62] McCusker, J. P.; Keshan, N.; Rashid, S.; Deagen, M.; Brinson, C.; McGuinness, D. L. NanoMine: A knowledge graph for nanocomposite materials science. The Semantic Web—ISWC 2020: 19th International Semantic Web Conference, Athens, Greece, November 2–6, 2020, Proceedings, Part II. 2020; pp 144–159.
- [63] Kearnes, S. M.; Maser, M. R.; Wlekliński, M.; Kast, A.; Doyle, A. G.; Dreher, S. D.; Hawkins, J. M.; Jensen, K. F.; Coley, C. W. The Open Reaction Database. *J. Am. Chem. Soc.* **143**, 18820–18826.
- [64] Taori, R.; Gulrajani, I.; Zhang, T.; Dubois, Y.; Li, X.; Guestrin, C.; Liang, P.; Hashimoto, T. B. Stanford Alpaca: An Instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca, 2023.
- [65] Alpaca-LoRA. <https://github.com/tloen/alpaca-lora>.
- [66] Colter, Z.; Fayazi, M.; Youbi, Z. B.-E.; Kamp, S.; Yu, S.; Dreslinski, R. Tablext: A combined neural network and heuristic based table extractor. *Array* **2022**, *15*, 100220.
- [67] Mamaghani, Z. G.; Hawboldt, K. A.; MacQuarrie, S. Adsorption of CO₂ using biochar - Review of the impact of gas mixtures and water on adsorption. *J. Environ. Chem. Eng.* **2023**, *11*, 109643.
- [68] Peng, Y.; Krungleviciute, V.; Eryazici, I.; Hupp, J. T.; Farha, O. K.; Yildirim, T. Methane Storage in Metal–Organic Frameworks: Current Records, Surprise Findings, and Challenges. *J. Am. Chem. Soc.* **2013**, *135*, 11887–11894.
- [69] Sahoo, B.; Pandey, V.; Dogonchi, A.; Mohapatra, P.; Thatoi, D.; Nayak, N.; Nayak, M. A state-of-art review on 2D material-boosted metal oxide nanoparticle electrodes: Supercapacitor applications. *J. Energy Storage* **2023**, *65*, 107335.
- [70] Suppiah, D. D.; Daud, W. M. A. W.; Johan, M. R. Supported Metal Oxide Catalysts for CO₂ Fischer–Tropsch Conversion to Liquid Fuels-A Review. *Energy Fuels*. **2021**, *35*, 17261–17278.
- [71] González-Vázquez, M.; García, R.; Gil, M.; Pevida, C.; Rubiera, F. Comparison of the gasification performance of multiple biomass types in a bubbling fluidized bed. *Energy Convers. Manag.* **2018**, *176*, 309–323.
- [72] Mohsin, M.; Farhan, S.; Ahmad, N.; Raza, A. H.; Kayani, Z. N.; Jafri, S. H. M.; Raza, R. The electrochemical study of NixCe_{1-x}O_{2-δ} electrodes using natural gas as a fuel. *New J. Chem.* **2023**, *47*, 8679–8692.
- [73] Kaur, P.; Singh, K. Review of perovskite-structure related cathode materials for solid oxide fuel cells. *Ceram. Int.* **2020**, *46*, 5521–5535.
- [74] Sengottuvelu, R. jsonformer. <https://github.com/1rgs/jsonformer>, 2018.
- [75] Choudhary, K.; Kelley, M. L. ChemNLP: A Natural Language Processing based Library for Materials Chemistry Text Data. *arXiv preprint arXiv:2209.08203* **2022**,
- [76] Kunc, K.; Loa, I.; Syassen, K.; Kremer, R.; Ahn, K. MgB₂ under pressure: phonon calculations, Raman spectroscopy, and optical reflectance. *arXiv preprint cond-mat/0105402*
- [77] FameLab International — Cheltenham Festivals. <https://www.cheltenhamfestivals.com/famelab>, last accessed 2023-05-30.
- [78] MT 180 - My Thesis in 180 Seconds. <https://www.epfl.ch/campus/events/events/public-events/my-thesis-in-180-seconds>, last accessed 2023-07-07.
- [79] CLIPDIGEST. <https://clipdigest.com/>, last accessed 2023-05-30.
- [80] Radford, A.; Kim, J. W.; Xu, T.; Brockman, G.; McLeavey, C.; Sutskever, I. Robust speech recognition via large-scale weak supervision. arXiv preprint: ArXiv-2212.04356. 2022.
- [81] Kim, S.; Chen, J.; Cheng, T.; Gindulyte, A.; He, J.; He, S.; Li, Q.; Shoemaker, B. A.; Thiessen, P. A.; Yu, B.; Zaslavsky, L.; Zhang, J.; Bolton, E. E. PubChem 2023 update. *Nucleic Acids Res.* **2022**, *51*, D1373–D1380.
- [82] Kim, S.; Chen, J.; Cheng, T.; Gindulyte, A.; He, J.; He, S.; Li, Q.; Shoemaker, B. A.; Thiessen, P. A.; Yu, B.; Zaslavsky, L.; Zhang, J.; Bolton, E. E. PubChem 2019 update: improved access to chemical data. *Nucleic Acids Res.* **2018**, *47*, D1102–D1109.
- [83] Kim, S.; Thiessen, P. A.; Cheng, T.; Yu, B.; Bolton, E. E. An update on PUG-REST: RESTful interface for programmatic access to PubChem. *Nucleic Acids Res.* **2018**, *46*, W563–W570.
- [84] Streamlit. <https://streamlit.io/>.

Acronyms

AI: artificial intelligence.

API: application programming interface.

BO: Bayesian optimization.

CAS: Chemical Abstract Services.

COT: chain of thought.

DFT: density functional theory.

DOI: digital object identifier.

ELN: electronic lab notebook.

GA: genetic algorithm.

GPR: Gaussian process regression.

GPT: generative pretrained transformer.

GUI: graphical user interface.

HTML: HyperText Markup Language.

ICL: in-context learning.

ID: inverse design.

InChI: international chemical identifier.

JSON: JavaScript object notation.

LIFT: language-interfaced fine-tuning.

LIMS: laboratory information system.

LLM: large language model.

LoRA: low-rank adaptors.

MAD: median absolute deviation.

MAE: mean absolute error.

MAPI: Materials Project API.

ML: machine learning.

NER: named entity recognition.

NLM: national library of medicine.

NLP: natural language processing.

OCR: optical character recognition.

ORD: Open Reaction Database.

PDB: protein data bank.

PEFT: parameter efficient fine-tuning.

RF: random forest.

RLHF: reinforcement learning from human feedback.

ROUGE: Recall-Oriented Understudy for Gisting Evaluation.

SELFIES: self-referencing embedded strings.

SMILES: simplified molecular-input line-entry system.

SVM: support vector machine.

UI: user interface.

