IOP Publishing J. Phys. Mater. **8** (2025) 035006 https://doi.org/10.1088/2515-7639/ade1fa

Journal of Physics: Materials

CrossMark

OPEN ACCESS

RECEIVED
25 March 2025

REVISED
16 May 2025

ACCEPTED FOR PUBLICATION
6 June 2025

PUBLISHED
24 June 2025

**PAPER**

# Retrieval augmented generation for building datasets from scientific literature

**Piyush Ranjan Maharana**[1,*], **Ashwini Verma**[1,2] **and Kavita Joshi**[1,2,*]

[1] Physical and Materials Chemistry Division, CSIR-National Chemical Laboratory, Pune 411008, India
[2] Academy of Scientific and Innovative Research (AcSIR), Ghaziabad 201002, India
[*] Authors to whom any correspondence should be addressed.

**E-mail:** pr.maharana@ncl.res.in and k.joshi@ncl.res.in

**Keywords:** LLM, RAG, dataset building, hydrogen storage, materials

## Abstract

In this work, we show that employing retrieval augmented generation (RAG) with a large language model (LLM) enables us to extract accurate data from scientific literature and construct datasets. The rapid growth in publications necessitates the automation of extraction of structured data as it is crucial for training machine learning(ML) models. The pipeline developed is simple and can be adjusted accordingly with natural language as input. Quantization enables us to run LLMs on consumer hardware and remove the reliance on closed-source models. Both Llama3-8B and Gemma2-9B with RAG give structured output consistently and with high accuracy as compared to direct prompting. Using the newly developed protocol, we created a data set of metal hydrides for solid-state hydrogen storage from paper abstracts. The accuracy of the generated dataset was >88% in the cases tested. Further, we demonstrate that the generated dataset is ready-to-use for ML models by testing it with HYST to predict the $H_2 wt\%$ at a given temperature. Thus, we demonstrate a pipeline to create datasets from scientific literature at minimal computational cost and high accuracy.

## 1. Introduction

High-quality data sets are critical for machine learning (ML) applications in the current data-driven paradigm of science. Advances in computational power and electronic structure methods have enabled the creation of large materials databases. Materials Project [1], open quantum materials Database (OQMD) [2], novel-materials-discovery (NOMAD) [3], joint automated repository for various integrated simulations (JARVIS) [4], open catalyst dataset (OC20) [5] are examples of computational databases which have led to screening of potential materials such as photocatalysts [6], carbon capture [7], bcc alloys [8], perovskites [9], electro-catalysts [10], multi-component crystalline solids [11] etc. OC20 which contains 1,281,040 density functional theory relaxations across a wide range of materials was used in Lan *et al* to achieve a $\sim$2000$\times$ speedup in computation time for identifying low energy adsorbate-surface configurations, paving the way for accelerated catalyst discovery [12]. The success of computational materials databases cannot be understated, being the driver of the current materials discovery revolution. However, at the same time, we need to be aware of the physical and practical limitations of computational methods when linking experiments and computations [13]. While popular and valuable, and efforts being made for improvement, these models often lack the complexity of real-life experimental conditions. Variables like temperature, pressure, electric/magnetic field, defects or addition of other elements in a material, effect of solvent are often difficult to represent computationally. Without experimental relevance to anchor the simulations, computation will just simulate ever-more-idealized systems [14, 15]. Effective data-driven discovery will require the integration of both experimental and simulation data to provide a more comprehensive picture. To bridge this gap, efforts are being made to develop experimental databases. A step in this direction are efforts like the High-Throughput Experimental database containing experimental synthesis and characterization data of thin films which will alleviate the issue of method verification and reproducibility [16]. Additionally, repositories like the Harvard Dataverse (https://dataverse.harvard.edu/) provide an online platform for researchers to share, preserve, cite, explore, and analyse research data. Being open to all researchers, it is a great way to contribute to the accessibility and integration of experimental data. While these repositories help structure new research data, historical/legacy data extraction still remains crucial. Although data sharing practices will become more widespread over time, currently vast amounts of research data remain locked within published papers. The immense influx of publications necessitates automated tools for data extraction, as manual methods are no longer feasible. There are exceedingly more efforts towards developing strategies and tools to automate data extraction. ChemData Extractor [17], a text mining toolkit has been used to automatically generate databases for yield strength and grain size [18], thermoelectric materials [19], refractive indices, Curie and Neel temperatures [20] and dielectric constants [21]. These methods require time and effort either in fine tuning models and making parsing rules as every new extraction task requires the user to define new grammatical and syntactic rules. The current state of the art (SOTA) in natural language processing(NLP) are transformers [22]. The early models ELMo [23], GPT [24], and BERT [25], demonstrated that unsupervised pre-training of language models on extensive text corpora greatly enhances performance across various NLP tasks such as named entity recognition, text classification, dependency parsing, hence there was a shift towards using these models for data extraction. BERT (Bidirectional Encoder representations) is a popular language model based on the transformer architecture [22]. BERTbase (110 million parameters) and BERT Large (340 million parameters) were pre-trained on the English Wikipedia and Toronto BookCorpus and they required further training on scientific text to be useful for data extraction as direct application gave sub-optimal results. It has been fine-tuned on various different text mining tasks in different fields such as BatteryBERT [26], SciBERT [27], MatSciBERT [28] and BioBERT [29]. Shetty *et al* [30] trained MaterialsBERT to extract material property from polymer literature abstracts while BatteryBERT by Huang and Cole [26] performs battery database enhancement, database creation and extractive question-answering for battery device component classification. Venugopal and Olivetti utilized BERT and GPT-3.5 for building MatKG, an extensive knowledge graph of materials science that captures a diverse range of entities and relationships from literature (5 million scientific papers) [15].

Fine-tuning large language models (LLMs) is resource and time consuming. For example SciBERT was pre-trained on a corpus of 1.14 M Papers (3.17 billion words) constituting of computer science and biomedical domain [27]. Gupta *et al* built MatSciBERT by further training with an addition of $\sim$150 K papers (3.45 billion words) [28]. Regarding the computational resources needed for training these models, MatSciBERT required fifteen days on two NVIDIA V100 32GB GPUs [28] while BatteryBERT needed eight NVIDIA DGX A100 GPUs [26]. All these requirements for extensive pre-training data and compute power, makes it difficult to get started with or may not be accessible to the majority of researchers. Also fine tuning makes the model field specific and difficult to transfer to other domains. Recently, a no fine-tuning approach was used by Polak and Morgan, where they constructed datasets for critical cooling rates of metallic glasses and yield strengths of high entropy alloys using their method ChatExtract which relies on GPT-4 [31]. This method requires successive prompts and also relies on a closed source model. Currently, the focus is shifted to using autoregressive LLMs for doing natural language tasks. LLMs are mainly transformer based language models that contain tens to hundreds of billions of parameters, pre-trained on massive amounts of text data. Some of them are open source such as Llama3 [32], Gemma [33] and others are closed (only accessible through API) such as GPT-4 [34] and Claude 3.5 Sonnet [35]. The performance of these models for various NLP tasks is impressive featuring broad capabilities and achieving SOTA. This ability of general purpose natural language understanding and natural language generation is acquired as LLMs' billions of parameters are trained on massive amounts of text data [36]. Even emergent abilities such as in-context learning, instruction following and multi-step reasoning are present in current LLMs that were absent in previous language models. However, they are still plagued by significant challenges. LLMs are generative text models and they hallucinate i.e make up false information in the text generated. LLMs have a knowledge cut-off date due to their fixed training data, hindering their ability to provide up-to-date information, especially in evolving scientific fields. While prompt engineering can improve reasoning, LLMs struggle with tasks needing current data and real-time calculations. Retrieval augmented generation (RAG) addresses this problem by incorporating external, up-to-date information, proving effective in areas like biomedical and clinical question-answering. For example, RAG supports personalized treatment, emergency triage, and disease management in clinical decision-making, showing better performance than using a base LLM [37]. Another bottleneck is their large sizes require considerable hardware just for model loading and even more for fine-tuning or pre-training. Although many closed source models from private companies like OpenAI and Anthropic can be accessed through APIs, the fees required can quickly become a problem for large scale information extraction over a huge text corpus. RAG is an information retrieval approach designed to overcome the limitations of using a LLM directly [38]. By augmenting the prompt of the LLM, it will retrieve the specific text chunks related to the query thereby increasing the relevance of the extracted data. With the rapid increase in the number of publications, instead of fine-tuning a model on a topic of interest, it is better to have a general model to be able to adapt to the incoming information. Fine-tuning is resource and time consuming and requires extensive preparation of training data, which may not be accessible to the majority of researchers. In this work we implement a RAG-LLM based approach to automate the extraction of accurate data from scientific literature. The substantial size of LLMs makes it hard to run on consumer hardware but the recent introduction of quantization has made this achievable. By reducing the precision of weight values, one can conserve memory and accelerate inference, all while maintaining most of the model's performance. With the emergence of ever better performing open source LLMs, there is a diverse ecosystem of LLMs that can compete with the current state-of-the-art models. Removing the reliance on closed source models, we are not limited by API call's rate or any fees and can rely on open source LLMs. To understand the memory requirements of these models we can take GPT3 and Llama-2 as an example. GPT3 requires 350 GB VRAM while Llama-2-70b requires 140 GB VRAM at 16-bit precision whereas depending on the base model, GPT-generated unified format (GGUF) quantized models might only need ∼10 GB VRAM to run inference. This allows the model to run on consumer hardware or on free GPUs available (Google Colab, Kaggle, Lightning AI). It has been found that model weights can be quantized to 8-bit or 4-bits without a significant loss in performance [39]. As an acceptable compromise between accuracy and memory, we have tested quantized versions of two open source LLMs, Llama3-8B [32] and Gemma2-9B [40]. Clever prompting tricks can generate desired output from LLMs but it is not always consistent and hence getting structured output from LLMs still remains difficult or requires a lot of post processing. In the present work, we demonstrate that both models with RAG give structured and highly accurate output consistently and hence enable us to construct ready-to-use datasets.

## 2. Methodology

We begin by taking abstracts from the published literature and then use an LLM to process and analyse these abstracts. The LLM's responses are enhanced with RAG so that they are more context aware. Finally we construct the dataset and then test its applicability by using the extracted data as input to an ML model.

### 2.1. Extracting abstracts

The abstracts were retrieved from Web of Science by doing an advanced search with appropriate keywords. Keywords can be selected according to the dataset to be created. Since our aim is to build a dataset of metal hydrides for solid-state hydrogen storage, the following filters were applied for the literature search: ((ALL = (metal hydrides)) AND (ALL = (hydrogen storage)) AND (ALL = (wt%)) which resulted into shortlisting 1611 articles after the search. This filtering step is crucial as it preserves resources and saves time and compute by not running the LLM on irrelevant text. We did not apply additional filtering or exclusion criteria, such as removing duplicates or screening for topic relevance, as the search query was already sufficiently specific to our research focus. We did not perform any preprocessing or text cleaning, since the abstracts were relatively short and well-formatted. Relevant information is added in the text for clarity.

### 2.2. Selecting LLM

LLMs were selected based on their VRAM size as the goal was to implement this extraction task with minimal memory requirements. The performance of LLMs is measured by comparing their ability to do certain tasks through standardized benchmarks such as (1) massive multitask language understanding (MMLU) consists of 15 908 questions covering 57 subjects and tests the general understanding of an LLM [41]; (2) artificial general intelligence (AGIEval) tests models on human-centric standardized exams such as entrance exams and math competitions. It tests a model's ability to handle simple reasoning and deduction on its set of 8062 questions [42]; (3) discrete reasoning over the content of paragraphs (DROP) evaluates models capability to navigate through references and execute operations like addition or sorting [43]; (4) HumanEval tests models on a set of 164 handwritten programming problems that evaluate for language comprehension, algorithms, and mathematics [44]; and (5) MATH consists of 12 500 challenging competition mathematics problems and tests the mathematical ability of LLMs [45]. In figure 1 performance of the two LLMs we selected in this work are compared with SOTA LLMs. Though LLM benchmarks are undeniably useful, it is the best practice to utilize them as guidance or strong indicators of a model's capabilities, rather than absolute determinants. We focused on open source models which were a good compromise between accuracy and size. Having low VRAM enables the code to run on free resources such as Google Colaboratory's Tesla T4 GPU which has 16GB VRAM. Based on the hardware, we selected Llama3-8B [32] and Gemma2-9B [40]. GGUF quantized versions of the models (Q40) were used for RAG while for direct prompting the bnb-4bit quantized versions were used. Quantization does reduce overall performance as the model size becomes smaller, but what we show in this work is that coupling the LLMs with RAG will help us claw back performance.

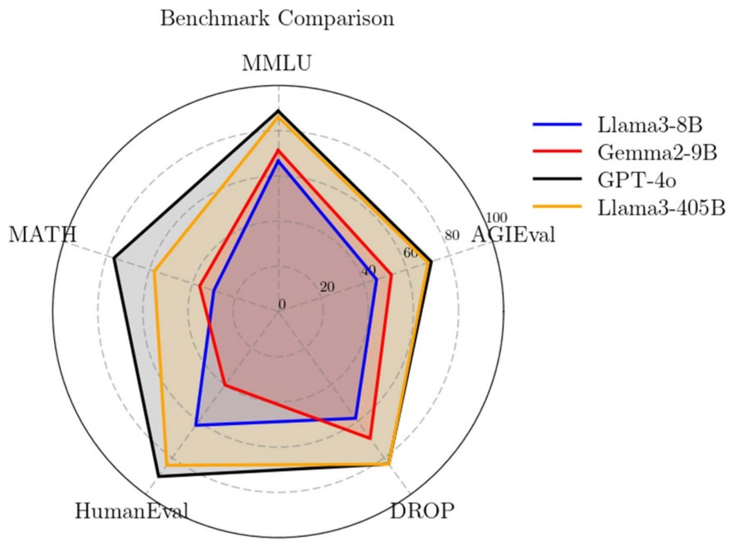
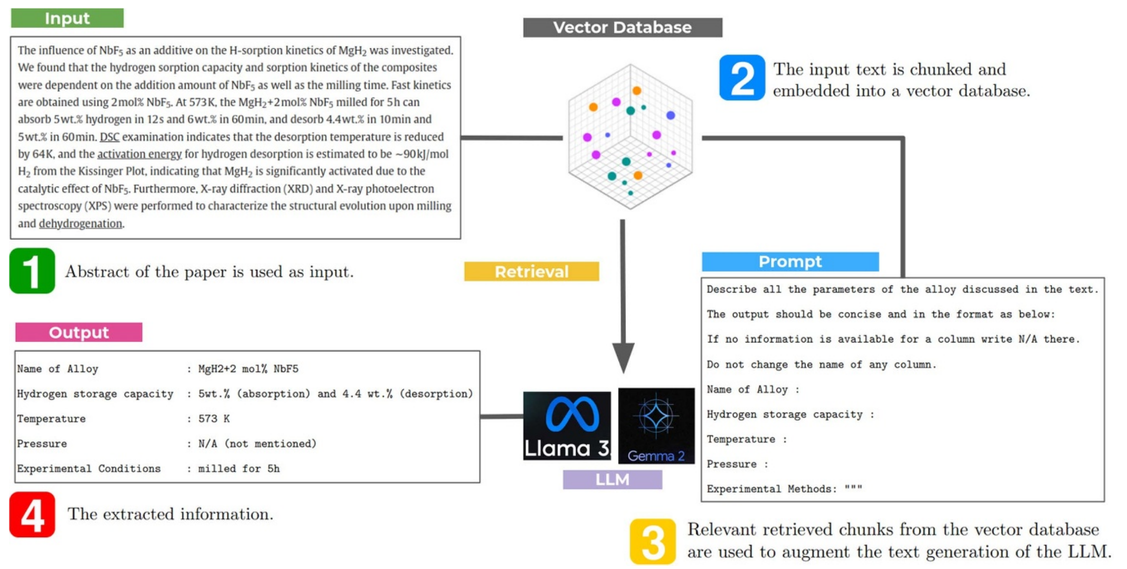


**Figure 1.** Performance of the selected LLMs on different benchmarks compared with GPT-4o [46] and Llama3-405B [47] (both currently in the state-of-the-art). On most benchmarks GPT-4o outperforms other LLMs. Gemma2-9B shows an overall better performance than Llama3-8B. Our task is to enhance the performance of the small models but still work on consumer hardware.

### 2.3. RAG

RAG enhances LLM's performance by retrieving relevant context for their prompts. RAG retrieves relevant text passages to augment the LLM's prompt, leading to more accurate and contextually appropriate responses. The initial step is data preparation for the RAG framework. In this work, we use plain text in the form of paper abstracts obtained from Web of Science. The vector representations of the associated text is stored in a vector store index for similarity comparison during the retrieval phase. To create these representations from text, embedding models are used. These models are designed to convert text into dense vector representations in a high-dimensional space. These vectors capture the semantic meaning of the text and further enable operations such as similarity comparisons. In the embedding space similar items will be positioned closer to each other as compared to less similar items. We have opted for the BAAI/bge-base-en-v1.5 (109 M) embedding model due to its efficient size and proven performance [48]. It provides sufficient embedding quality for our abstract data extraction, offering a good balance between size and performance. In the retrieval stage the user query is converted into a vector representation using the embedding model. Similarity scores between the query vector and the vectorized chunks in the vector store decide the ordering of the chunks. In RAG the top $k$ chunks of the input text that are similar to the user query from the vector store index are selected. These chunks are then used for generating the answer by the LLM. Identifying the optimal chunk size of the text is crucial to ensure accuracy in the final result during retrieval. After experimenting with different values and combinations we finalized on 256 and 20 tokens for the chunk size and overlap respectively for making the vector store and $k = 3$ for querying the chunks. These values depend on the length of the text and need to be adapted as per the task.

In figure 2 we demonstrate the workflow of data extraction using RAG employed in this work. In the first step, the abstract is chunked, indexed, and embedded into a vector store. Finally the chunks are used to augment the prompt at the time of generating the text from the LLM. The prompt that goes into the query engine can be modified as per requirement. In our present work our goal is to extract data about hydrogen storage alloys (composition of the alloy, pressure, temperature, synthesis method, and associated hydrogen storage capacity). We did a few experiments with slightly different variants of this prompt so that the output is in the desired format. The LLM inference time depends on the number of tokens generated. So, it is important to engineer the prompt in such a way that the output is restricted to the required information. With RAG we do not rely solely on the text generation ability of an LLM on the data it was trained on, but we augment it with the actual documents/text we want to query to retrieve accurate and reliable data. Also the performance of LLMs degrades with the increase in input length [49]. As our RAG implementation can be interacted with natural language, it is effective in creating databases and also querying research papers directly for any other information that the user wants.


Input
Vector Database
2 The input text is chunked and embedded into a vector database.
1 Abstract of the paper is used as input.
Retrieval
Prompt
Output
Llama 3
Gemma 2
LLM
4 The extracted information.
3 Relevant retrieved chunks from the vector database are used to augment the text generation of the LLM.

The influence of $NbF_5$ as an additive on the H-sorption kinetics of $MgH_2$ was investigated. We found that the hydrogen sorption capacity and sorption kinetics of the composites were dependent on the addition amount of $NbF_5$ as well as the milling time. Fast kinetics are obtained using 2mol% $NbF_5$. At 573K, the $MgH_2$+2mol% $NbF_5$ milled for 5h can absorb 5wt.% hydrogen in 12s and 6wt.% in 60min, and desorb 4.4wt.% in 10min and 5wt.% in 60min. DSC examination indicates that the desorption temperature is reduced by 64K, and the activation energy for hydrogen desorption is estimated to be ~90kJ/mol $H_2$ from the Kissinger Plot, indicating that $MgH_2$ is significantly activated due to the catalytic effect of $NbF_5$. Furthermore, X-ray diffraction (XRD) and X-ray photoelectron spectroscopy (XPS) were performed to characterize the structural evolution upon milling and dehydrogenation.

```
Describe all the parameters of the alloy discussed in the text.
The output should be concise and in the format as below:
If no information is available for a column write N/A there.
Do not change the name of any column.
Name of Alloy :
Hydrogen storage capacity :
Temperature :
Pressure :
Experimental Methods: """
```

```
Name of Alloy              : MgH2+2 mol% NbF5
Hydrogen storage capacity  : 5wt.% (absorption) and 4.4 wt.% (desorption)
Temperature                : 573 K
Pressure                   : N/A (not mentioned)
Experimental Conditions    : milled for 5h
```

**Figure 2.** Workflow of data extraction using RAG in this work. The abstract is chunked, indexed, and embedded into a vector store/database. The chunks are then used to augment the prompt at the time of generating the text from the LLM. The extracted information can be easily parsed and put into a dataset.

## 3. Results and discussion

### 3.1. Extraction

We have employed this framework (Llama3-8B with RAG) to construct a dataset of metal hydrides used for solid-state hydrogen storage by extracting information from selected abstracts. Hydrogen is one of the promising candidates for replacing fossil fuels in the current climate crisis. However its wide scale utilization and realistic adoption is hindered by the challenges faced to store it safely and efficiently. Research on solid-state hydrogen storage spans a huge chemical space, but existing experimental database like HydPARK database [50] (last updated in 2002), is outdated. Our efforts of manual data extraction, such as HyStor [51] to update the HydPARK, highlighted the critical need for automated data extraction to effectively screen for suitable metal hydrides. For demonstration of the framework we have used only abstracts to extract the information. However, the framework is generalized and could be used to extract information from the complete text of the published paper. Given the prompt, we test the output of the LLMs for the metal hydrides abstracts. LLM inference was done on a Tesla T4 GPU in Google Colab. In direct prompting we use the alpaca prompt format. The basic format of the prompt is shown below.

**Alpaca Prompt**

```
alpaca_prompt = """Below is an instruction that describes a task, paired with an input that
provides further context. Write a response that appropriately completes the request.
### Instruction:
{}
### Input:
{}
### Response:
{}"""
```

in which the input text is the abstract and the instruction is the following text.

**Instruction**

```
Describe all the parameters of the material discussed in the text.
If no information is available just write "N/A".
The output should be concise and in the format as below:

Name of Alloy                       :
Hydrogen storage capacity           :
Temperature                         :
Pressure                            :
Experimental Conditions             :
```

The ‘instruction’ in the alpaca format for direct prompting is also used as the prompt for RAG. Following are the outputs for the LLMs on a single abstract for both direct prompting and RAG.

**Llama3-8B**

**Prompting**

```
Name of Alloy               : Mg2NiH4
Hydrogen storage capacity   : 6.7 wt %
Temperature                 : 320 C-degrees
Pressure                    : 50 atm H-2
Experimental Conditions     : 300 s, 320 C-degrees, 50 atm H-2
```

**RAG**

```
Name of Alloy               : Mg+10 wt % FeNi@3DG composite
Hydrogen storage capacity   : 6.7 wt %
Temperature                 : 320°C (during hydrogen absorption)
Pressure                    : 50 atm H-2 (during hydrogen absorption) and 0.5 atm H-2
(during hydrogen desorption)
Experimental Conditions     : Ball milling
```

**Gemma2-9B**

**Prompting**

```
Name of Alloy                : Mg2NiH4
Hydrogen storage capacity    : 6.7 wt %
Temperature                  : 320 C-degrees
Pressure                     : 50 atm H-2
Experimental Conditions      : 320 C-degrees, 50 atm H-2
```

**RAG**

```
Name of Alloy                : Mg+10 wt % FeNi@3DG composite
Hydrogen storage capacity    : 6.7 wt % H-2 absorbed, 6.5 wt % desorbed
Temperature                  : 320 C-degrees
Pressure                     : 50 atm H-2 for absorption, 0.5 atm H-2 for desorption
Experimental Conditions      : Ball milling
```

Both the LLMs are able to extract the data in a structured format but in the case of RAG the alloy name mentioned in the abstract is extracted correctly by the LLM. We also note that the columns needed can be changed quite easily in one single prompt making the extraction process simple and flexible as compared to the ChatExtract method by Polak and Morgan that requires multiple prompts [31].

### 3.2. Evaluation

We have used this framework to extract information from the 1611 abstracts. These abstracts were selected by using 'metal hydrides', 'hydrogen storage' and '*wt*%' as keywords on Web of Science. Out of 1611 entries, name of alloy could be extracted from 1433 abstracts, hydrogen storage capacity could be filtered out from 1195 abstracts, value of temperature from 1203 and that of pressure from 550 abstracts. Finally, experimental methods are extracted from 1438 entries. In the generated dataset 350 rows have all the entries, i.e. name of alloy, corresponding $H_2$*wt*%, temperature, pressure, and experimental methods. A subset of this dataset is shown in table 1 and the complete extracted dataset is provided in the Github repository. Further, 836 alloys have $H_2$*wt*% at a given temperature for a given composition whereas 1072 entries have information about composition and corresponding $H_2$*wt*%. We would like to note that the missing information is not necessarily because the framework fails to capture it , but mostly it is not included in the abstract. Crucially, HydPARK caps out at year 2002 and the abstracts that we queried from Web of Science are much more recent (1563 were after 2002).

Sometimes the framework fails to capture the accurate information. An example of incorrect and hallucinated response is shown in figure 3. In the first abstract we see that although the alloy name is correct but the LLM hallucinates the value of the hydrogen storage capacity (7.5 *wt*% instead of 5.3 and 5.6 *wt*%). In the second abstract too although there is a mention of 'decreases with increasing Fe content' earlier in the text the last line mentions the storage capacity for the alloy which the LLM misses.

The most reliable although time consuming evaluation is the acceptance rate by users or comparisons made by humans. To evaluate the performance of the RAG-LLM framework, we manually validated a subset of the extracted data. From our total dataset of 1611 abstracts, we randomly selected 250 samples (15.5%) for manual verification. This sample size was chosen to achieve a 90% confidence level ($Z$ score $= 1.645$) with a 5% margin of error. For this purpose we focused on three columns: Name of Alloy, Hydrogen storage capacity and Temperature as this will enable us to use the generated dataset to test our previously developed ML models. The responses of the two LLMs were tested on these 250 sampled abstracts on two different scenarios: Direct Prompting and with RAG. The validation process involved comparing the system-extracted information against the source abstracts. The criteria for considering a output as accurate, incomplete and hallucinated is as follows:

- Accurate: The extracted information is correct.
- Incomplete/Incorrect: Property is not mentioned or is mentioned but the information is incomplete/consists of only unrelated text.
- Hallucination: Hallucination of the data. No relation to the input text.

The results on the test set for Gemma2-9B and Llama3-8B are shown in figures 4 and 5 respectively. We see a marked increase in accurate responses and reduction in hallucinations with RAG as compared to direct prompting. Both models show substantial improvements across all three metrics (Name of the alloy, $H_2$*wt*%, and temperature) when RAG is implemented. Alloy name accuracy increased significantly: Gemma2-9B from 65.2% to 90%, and Llama3-8B from 80.4% to 93.6%. $H_2$*wt*% accuracy rose from 72% to 95.2% for Gemma2-9B and from 82.4% to 88.0% for Llama3-8B. Temperature accuracy improved from 76.0% to 96.8% for Gemma2-9B and from 77.2% to 96.8% for Llama3-8B. The highest increase is seen in correctly extracting the name of alloy in both models suggesting that RAG provides the models with the necessary context to accurately identify alloy names. The evaluation dataset, including the 250 manually validated samples for each of the two LLMs with their corresponding source abstracts and extraction results, is provided for reproducibility and future comparative studies.

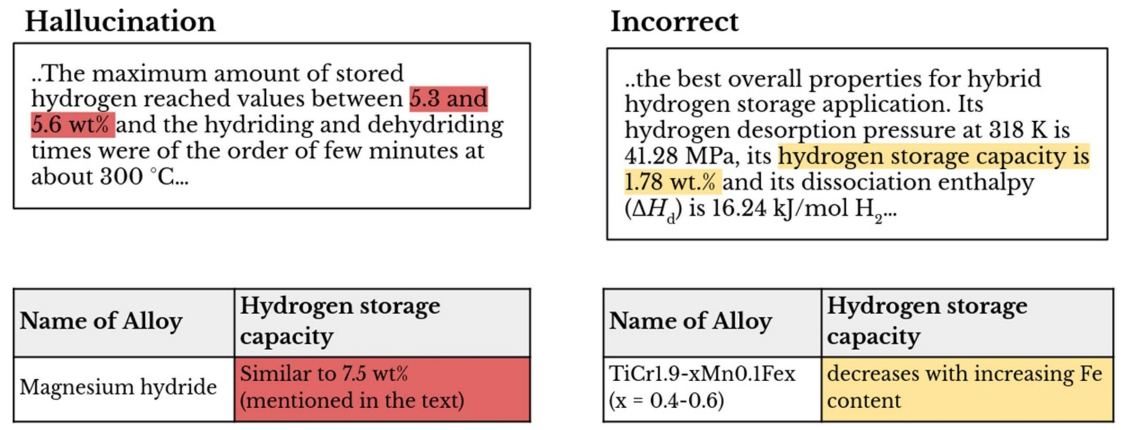
Finally, to demonstrate the correctness of extracted data, we show the distribution of the $H_2$*wt*% in the dataset in figure 6. The upper tail of the distribution primarily features ammonia-borane ($NH_3BH_3$), borohydrides ($LiBH_4$, $Mg(BH_4)_2$, $Ca(BH_4)_2$), complex hydrides ($LiAlH_4$), and their derivatives. The central peak is dominated by magnesium-containing materials or composites, such as $MgH_2 + Li_3AlH_6$, Mg–Ti–Cr–Nb, $MgH_2$–H–$V_2O_5$, and $Li_2MgN_2H_2$. The slow kinetics and unfavourable thermodynamics of $MgH_2$ render it unsuitable for practical applications. The abundance of $MgH_2$-based materials stems from various modifications and additions aimed at enhancing its usability. Below 3 wt%, the dataset primarily comprises Ti, La, and rare earth metal-based alloys, including TiZrCrMnFeNi, $Ti_{33}V_{37}Mn_{30}$, TiZrNbTa, $LaNi_5$, $La_{0.7}Mg_{0.3}Ni_{3.4-x}Co_{0.6}Mn_x$, and $Ce_2MgNi_2$.

**Table 1.** A subset of the generated dataset showing hydrogen storage properties of various alloys and compounds extracted from the paper's abstracts.

| Name of Alloy | Hydrogen storage capacity | Temperature | Pressure | Experimental Methods |
|---|---|---|---|---|
| N-ethylcarbazole (NEC) | 6.1 wt % | 226 °C (start), 300 °C (30 min) | N/A | Ball milling, addition of sodium fluotitanate |
| $Mg_{95}Ni_5$ | 3.37 wt.% (abs.), 1.02 wt.% (des.) | 473 K, 373 K | N/A | Mechanical milling with tetrahydrofuran solution of PMMA under argon |
| $Zr_8Ni_{21}$ | 6.5 wt% (for 3 and 5 wt% additive) | 250 °C, 300 °C | 10 bar | Ball milling $MgH_2$ with $BaMnO_3$ additive |
| 52Ti-12 V-36Cr doped with $Zr_7Ni_{10}$ | 3.4 wt.% (single melt), 3.2 wt.% (co-melt) | Room temp. (desorption) | N/A | Single melt and co-melt methods, XRD patterns, hydrogenation |
| $La_{0.8}Mg_{0.2}Ni_{3.8}$ | 0.08–0.15 wt% | 0 °C–100 °C | N/A | Varying PVDF content, $(NH_4)_2CO_3$ concentration, and temperature |
| $Ti_{1+x}Cr_{1.2}Mn_{0.2}Fe_{0.6}$ | 1.61 wt% | 318 K | 37.69 MPa | Induction levitation melting |
| $(Ni\text{-}V_2O_3)$ @C-containing sample, $MgH_2$ | 5.50 wt% at 25 °C, 6.05 wt% at 275 °C | 25 °C–275 °C | N/A | Milling with $MgH_2$ |
| $MgH_2$-Ni-MOF@Pd | 2.62 wt% at 100 °C, 6.06 wt% at 150 °C | 100 °C | N/A | Under catalytic effect of Ni-MOF@Pd |
| MGH@NGNSs nanocomposite | 6.5 wt% (abs.), 5.5 wt% (des.) | 200 °C (abs.), 300 °C (des.) | N/A | Wet-chemical method, nano-size effect of MGHs, NGNSs |
| $(1\text{-}y)MgH_2+yTiH_2$ | 4.9 wt% H | 573 K | N/A | Reactive ball milling under hydrogen gas, limited reaction time |
| 0.71 $LiBH_4$-0.29 $NaBH_4$ | ∼3.5 wt.% $H_2$ | N/A | N/A | Melt infiltration into CMK-3 type carbon pores or graphitic carbon discs |

### 3.3. Testing

To demonstrate the applicability of the extracted data, we employed our pre-trained ML model, HYST (**Hy**drogen **st**orage capacity predictor) [51], to predict hydrogen storage capacities ($H_2wt\%$) for the compositions within the dataset. Our extracted dataset is composed of various material types, including complex hydrides, metal hydrides, oxides, and carbides. However, we focused solely on metal hydrides, as HYST is trained exclusively on metal hydrides with some features available for only 28 elements [52]. After filtering for relevant compositions, we identified 71 compositions compatible with HYST. Following duplicate removal, this list was refined to 59 unique compositions. Before being used as test set for the HYST model, all 59 compositions were manually verified, and no hallucinated or incorrect entries were found. Figure 7 illustrates the validation of HYST's predictions against experimental data. The blue bars represent experimentally measured hydrogen storage capacities, while the red bars show ML-predicted values. While the ML predictions generally align with experimental trends, some deviations highlight areas for improvement. The diverse compositions tested help evaluate the model's performance and limitations. This



Hallucination

..The maximum amount of stored hydrogen reached values between 5.3 and 5.6 wt% and the hydriding and dehydriding times were of the order of few minutes at about 300 °C...

| Name of Alloy | Hydrogen storage capacity |
|---|---|
| Magnesium hydride | Similar to 7.5 wt% (mentioned in the text) |

Incorrect

..the best overall properties for hybrid hydrogen storage application. Its hydrogen desorption pressure at 318 K is 41.28 MPa, its hydrogen storage capacity is 1.78 wt.% and its dissociation enthalpy ($\Delta H_d$) is 16.24 kJ/mol $H_2$...

| Name of Alloy | Hydrogen storage capacity |
|---|---|
| TiCr1.9-xMn0.1Fex (x = 0.4-0.6) | decreases with increasing Fe content |

**Figure 3.** An example of a hallucination and a incorrect/incomplete response from the LLM. Incomplete responses happen more often when there are multiple alloys mentioned in the abstract.

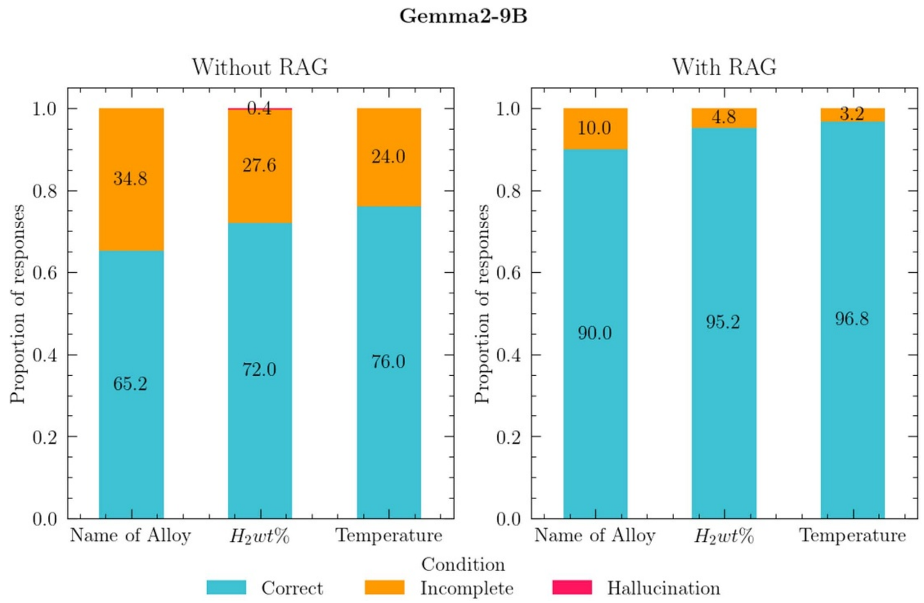


**Figure 4.** Accuracy of the Gemma2-9B model on the test set for direct prompting (without RAG) and RAG. Gemma2-9B initially does not show much hallucination but the number of correct responses increase with RAG.

dataset can be incorporated into HYST for model improvement, and expanding the feature set could allow inclusion of additional material classes. The current dataset of 59 points can be further incorporated into the HYST model to enhance its predictive capabilities. Similarly, by expanding the feature set, we can include additional material classes (complex hydrides, oxides, and carbides) in the model training, transforming HYST into a more comprehensive and generalized hydrogen weight capacity predictive tool. Our work is a step towards automation of dataset building from the published literature, addressing a critical gap in existing databases and potentially improving future ML models' predictive capabilities.

## 4. Limitations

LLMs are nondeterministic by nature, so the results will slightly change on each run. In RAG the optimal chunk size and chunk overlap change according to the use case, so it may not be immediately better than the direct prompting method and require tuning. The evaluations were run on metal hydride abstracts and further testing is required to see whether the increase in accuracy will be present for other scientific domains and for longer texts as well. In the case of multiple alloys mentioned in the abstract, it becomes difficult to extract and this is one of the reasons of incomplete/incorrect responses. We'll try to address the effects of all these variables on data extraction performance and the efficacy of RAG against direct prompting in future work. On a more technical note Q40 quantization scheme was used for RAG as Ollama

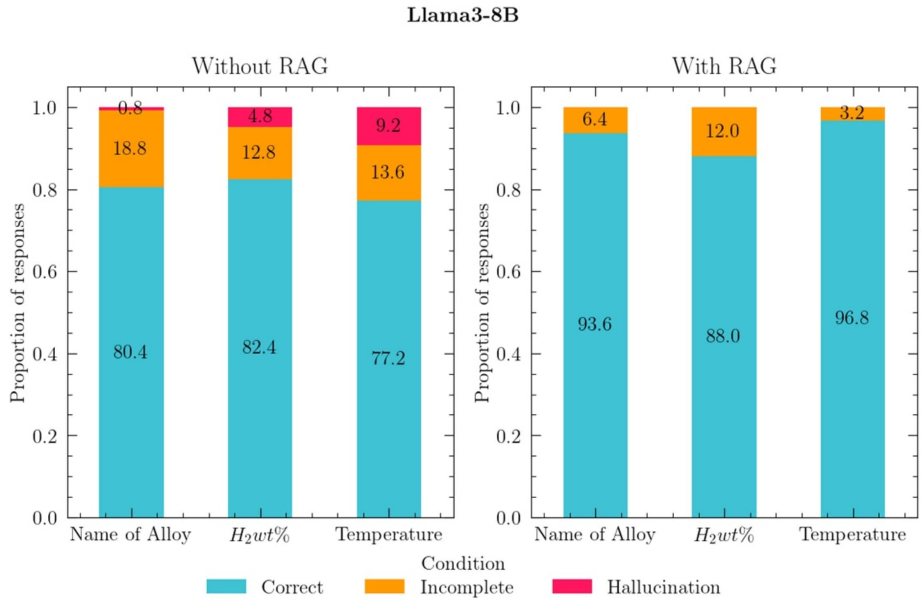


**Figure 5.** Accuracy of the Llama3-8B model on the test set for direct prompting (without RAG) and RAG. Llama3-8B initially shows more hallucination than Gemma2-9B but upon the use of RAG, a similar trend in accuracy is observed.

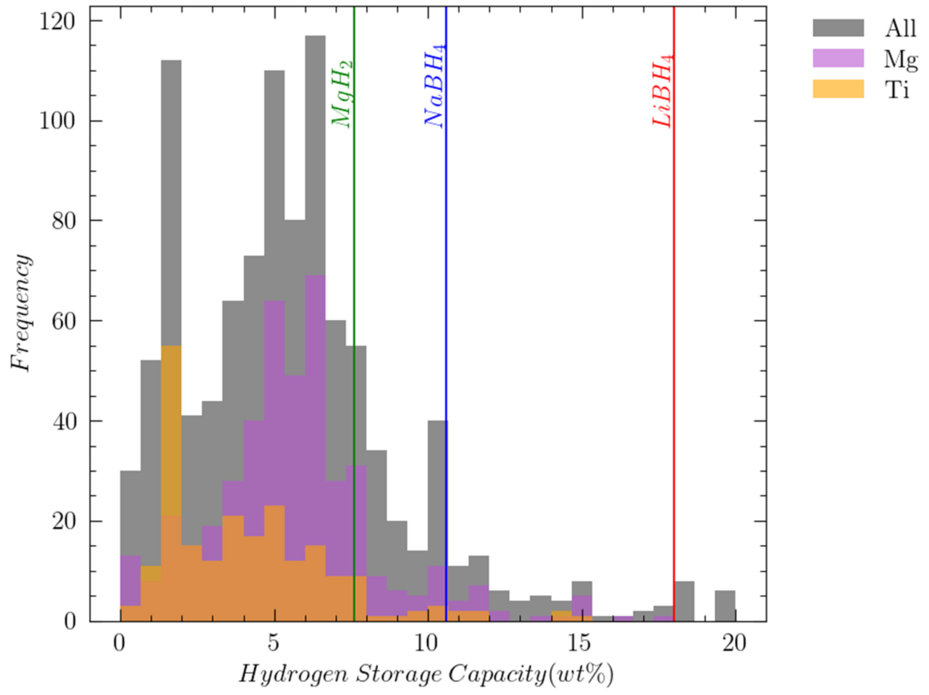


**Figure 6.** Distribution of hydrogen storage capacity($H_2$ wt%) in the metal hydrides dataset extracted from the abstracts by the pipeline described in this work. The majority of hydrides are concentrated within the 3–8 wt% range. Three prominent metal hydrides and their corresponding capacities are highlighted: $LiBH_4$: 18 wt%, $NaBH_4$: 10.6 wt%, $MgH_2$: 7.6 wt%. The distribution of compositions containing Mg and Ti are also shown.

(https://ollama.com/) was utilized to run the LLMs. While for direct prompting we used the bnb 4-bit quantized versions of the LLMs, and ran the inference using Unsloth (https://github.com/unslothai/unsloth). Although both of them are 4-bit quantizations, there are slight differences in the way they quantize the weights in the transformer architecture. This can alter the output of the LLMs and affect the evaluations.

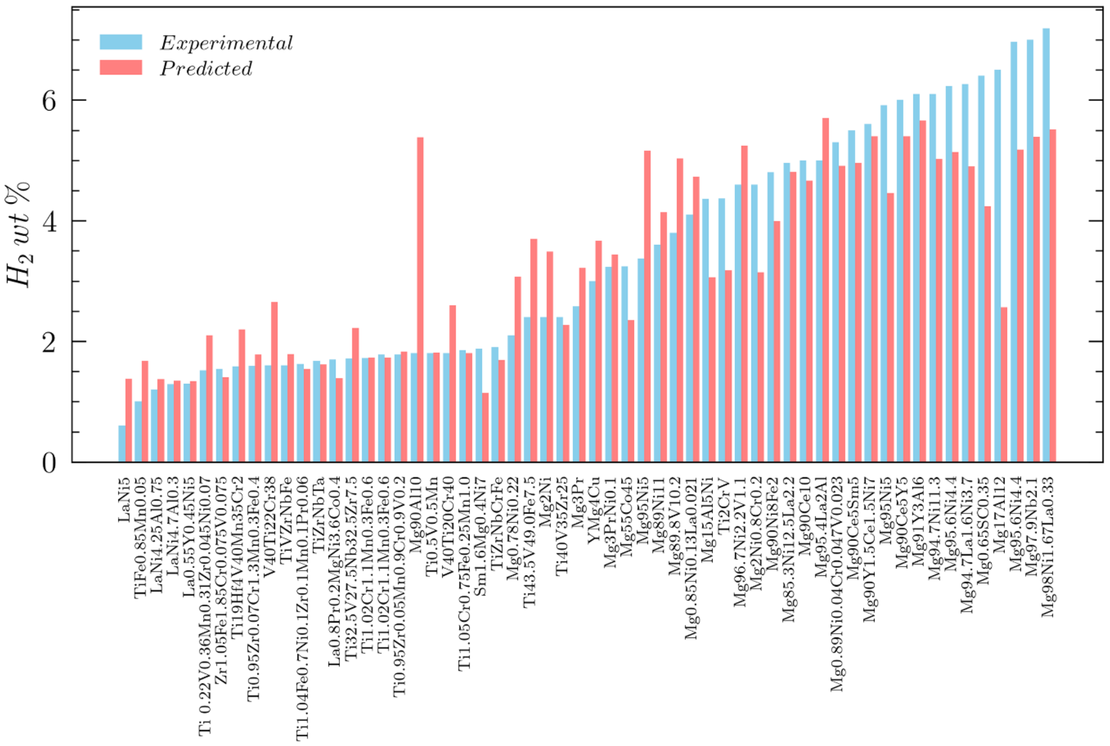


**Figure 7.** Comparison of experimental (blue bars) and machine learning-predicted (red bars) hydrogen storage capacities by HYST($H_2$ wt%) for various alloys from the dataset. As the input data points were taken without much post processing from the created dataset, this shows the applicability of our framework for ML models.

## 5. Conclusion

Automating data extraction using LLMs will make scientific information more accessible. However, these models still face challenges such as hallucinations and unboundedness. The advent of open-source LLMs and quantization makes it possible to run inference on consumer hardware. Using retrieval-augmented generation along with quantized LLMs enables us to extract consistent and highly accurate structured output, paving the way for creation of high-quality datasets that are critical for machine learning applications. Using the framework, we created a dataset of metal hydrides for hydrogen storage with an overall >88% accuracy. Also, we see an increase in the number of accurate responses and a reduction in incorrect/hallucinated responses upon deploying RAG as compared to direct prompting in quantized LLMs. For Gemma 2-9B and Llama 3-8B we see a reduction in the number of incorrect responses by 24.8% and 12.4% respectively. Our framework successfully extracted information from 1611 abstracts about the material compositions along with their hydrogen storage capacities and experimental conditions, such as temperature and pressure. The data extracted serve as a valuable resource for training machine learning models to predict hydrogen storage capacities based on chemical compositions and experimental parameters. Using HYST, we were able to test the suitability of the data as input for ML models. The results presented here show that this framework is useful for rapidly extracting information from literature. With the advancement of ever more powerful and efficient language models the accuracy would increase even further.

## Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://github.com/catastropiyush/RAG-dataset-gen.

## Acknowledgments

Authors gratefully acknowledge financial support from CSIR network Project, HCP-44-05 (HEART) and SERB POWER Grant SPG/2022/000890.

## Conflict of interest

There are no conflicts to declare.

## CRediT authorship contribution statement

**Piyush Ranjan Maharana:** Conceptualization, Methodology, Analysis, Writing original draft

**Ashwini Verma :** Running the HYST model

**Kavita Joshi :** Conceptualization, Writing and editing, Supervision, Project administration, and Funding acquisition

## ORCID iD

Kavita Joshi https://orcid.org/0000-0001-6079-4568

## References

[1] Jain A *et al* 2013 Commentary: the materials project: a materials genome approach to accelerating materials innovation *APL Mater.* **1** 011002

[2] Saal J E, Kirklin S, Aykol M, Meredig B and Wolverton C 2013 Materials design and discovery with high-throughput density functional theory: the open quantum materials database (OQMD) *JOM* **65** 1501–9

[3] Draxl C and Scheffler M 2019 The NOMAD laboratory: from data sharing to artificial intelligence *J. Phys. Mater.* **2** 036001

[4] Choudhary K *et al* 2020 The joint automated repository for various integrated simulations (JARVIS) for data-driven materials design *npj Comput. Mater.* **6** 173

[5] Chanussot L *et al* 2021 Open catalyst 2020 (OC20) dataset and community challenges *ACS Catal.* **11** 6059–72

[6] Yan Q *et al* 2017 Solar fuels photoanode materials discovery by integrating high-throughput theory and experiment *Proc. Natl Acad. Sci.* **114** 3040–3

[7] Dunstan M T, Jain A, Liu W, Ong S P, Liu T, Lee J, Persson K A, Scott S A, Dennis J S and Grey C P 2016 Large scale computational screening and experimental discovery of novel materials for high temperature $CO_2$ capture *Energy Environ. Sci.* **9** 1346–60

[8] Nyshadham C, Rupp M, Bekker B, Shapeev A V, Mueller T, Rosenbrock C W, Csányi G, Wingate D W and Hart G L W 2019 Machine-learned multi-system surrogate models for materials prediction *npj Comput. Mater.* **5** 51

[9] Balachandran P V, Emery A A, Gubernatis J E, Lookman T, Wolverton C and Zunger A 2018 Predictions of new $ABO_3$ perovskite compounds by combining machine learning and density functional theory *Phys. Rev. Mater.* **2** 043802

[10] Hong W T, Welsch R E and Shao-Horn Y 2016 Descriptors of oxygen-evolution activity for oxides: a statistical evaluation *J. Phys. Chem. C* **120** 78–86

[11] Natarajan A R and Van der Ven A 2018 Machine-learning the configurational energy of multicomponent crystalline solids *npj Comput. Mater.* **4** 56

[12] Lan J, Palizhati A, Shuaibi M, Wood B M, Wander B, Das A, Uyttendaele M, Zitnick C L and Ulissi Z W 2023 AdsorbML: a leap in efficiency for adsorption energy calculations using generalizable machine learning potentials *npj Comput. Mater.* **9** 172

[13] Verma P and Truhlar D G 2020 Status and challenges of density functional theory *Trends Chem.* **2** 302–18

[14] Horton M K, Dwaraknath S and Persson K A 2021 Promises and perils of computational materials databases *Nat. Comput. Sci.* **1** 3–5

[15] Venugopal V and Olivetti E 2024 MatKG: an autonomously generated knowledge graph in material science *Sci. Data* **11** 217

[16] Horton M K and Woods-Robinson R 2021 Addressing the critical need for open experimental databases in materials science *Patterns* **2** 100411

[17] Swain M C and Cole J M 2016 Chemdataextractor: a toolkit for automated extraction of chemical information from the scientific literature *J. Chem. Inf. Model.* **56** 1894–904

[18] Kumar P, Kabra S and Cole J M 2022 Auto-generating databases of yield strength and grain size using chemdataextractor *Sci. Data* **9** 292

[19] Sierepeklis O and Cole J M 2022 A thermoelectric materials database auto-generated from the scientific literature using chemdataextractor *Sci. Data* **9** 648

[20] Court C J and Cole J M 2018 Auto-generated materials database of Curie and Néel temperatures via semi-supervised relationship extraction *Sci. Data* **5** 1–12

[21] Zhao J and Cole J M 2022 A database of refractive indices and dielectric constants auto-generated using chemdataextractor *Sci. Data* **9** 192

[22] Vaswani A *et al* 2023 Attention is all you need (arXiv:1706.03762)

[23] Peters M E *et al* 2018 Deep contextualized word representations *Proc. 2018 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* vol 1 (Association for Computational Linguistics) pp 2227–37

[24] Radford A *et al* 2018 Improving language understanding by generative pre-training (available at: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf)

[25] Devlin J *et al* 2019 Bert: pre-training of deep bidirectional transformers for language understanding *Proc. 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* vol 1 pp 4171–86 (arXiv:1810.04805)

[26] Huang S and Cole J M 2022 BatteryBERT: a pretrained language model for battery database enhancement *J. Chem. Inf. Model.* **62** 6365–77

[27] Beltagy I *et al* 2019 SciBERT: a pretrained language model for scientific text *Proc. 2019 Conf. on Empirical Methods in Natural Language Processing and the 9th Int. Joint Conf. on Natural Language Processing (EMNLP-IJCNLP)* (Association for Computational Linguistics) pp 3615–20

[28] Gupta T, Zaki M, Krishnan N M A and Mausam 2022 MatSciBERT: a materials domain language model for text mining and information extraction *npj Comput. Mater.* **8** 102

[29] Lee J, Yoon W, Kim S, Kim D, Kim S, So C H and Kang J 2020 BioBERT: a pre-trained biomedical language representation model for biomedical text mining *Bioinformatics* **36** 1234–40

[30] Shetty P, Rajan A C, Kuenneth C, Gupta S, Panchumarti L P, Holm L, Zhang C and Ramprasad R 2023 A general-purpose material property data extraction pipeline from large polymer corpora using natural language processing *npj Comput. Mater.* **9** 52

[31] Polak M P and Morgan D 2024 Extracting accurate materials data from research papers with conversational language models and prompt engineering *Nat. Commun.* **15** 1569

[32] Meta AI 2024 Introducing Llama 3: our next-generation large language model (available at: https://ai.meta.com/blog/meta-llama-3/)

[33] Team G *et al* 2024 Gemma: open models based on gemini research and technology (arXiv:2403.08295)

[34] Achiam J *et al* 2023 Gpt-4 technical report (arXiv:2303.08774)

[35] Introducing the next generation of claude — anthropic.com (available at: www.anthropic.com/news/claude-3-family)

[36] Minaee S *et al* 2024 Large language models: a survey (arXiv:2402.06196)

[37] Liu S, McCoy A B and Wright A 2025 Improving large language model applications in biomedicine with retrieval-augmented generation: a systematic review, meta-analysis and clinical development guidelines *J. Am. Med. Inf. Assoc.* **32** 605–615, 01

[38] Gao Y *et al* 2023 Retrieval-augmented generation for large language models: a survey (arXiv:2312.10997)

[39] Dettmers T *et al* 2022 Gpt3. int8: 8-bit matrix multiplication for transformers at scale *Advances in Neural Information Processing Systems* vol 35 pp 30318–32 (available at: https://openreview.net/forum?id=dXiGWqBoxaD)

[40] Team G *et al* 2024 Gemma 2: improving open language models at a practical size (arXiv:2408.00118)

[41] Hendrycks D *et al* 2020 Measuring massive multitask language understanding (arXiv:2009.03300)

[42] Zhong W *et al* 2023 Agieval: a human-centric benchmark for evaluating foundation models (arXiv:2304.06364)

[43] Dua D *et al* 2019 Drop: a reading comprehension benchmark requiring discrete reasoning over paragraphs (arXiv:1903.00161)

[44] Chen M *et al* 2021 Evaluating large language models trained on code (arXiv:2107.03374)

[45] Hendrycks D *et al* 2021 Measuring mathematical problem solving with the math dataset (arXiv:2103.03874)

[46] Hurst A *et al* Gpt-4o system card (arXiv:2410.21276)

[47] Dubey A *et al* 2024 The Llama 3 herd of models (arXiv:2407.21783)

[48] Xiao S *et al* 2023 C-pack: packaged resources to advance general chinese embedding (arXiv:2309.07597)

[49] Levy M *et al* 2024 Same task, more tokens: the impact of input length on the reasoning performance of large language models *Proc. 62nd Annual Meeting of the Association for Computational Linguistics* (Association for Computational Linguistics) pp 15339–53

[50] Hydrogen storage materials database 2019 (available at: https://datahub.hymarc.org/ca/dataset/hydrogen-storage-materials-db/resource/775e172f-9827-44bb-aebe-f5ae488d0419)

[51] Wilson N, Verma A, Maharana P R, Sahoo A B and Joshi K 2024 Hystor: an experimental database of hydrogen storage properties for various metal alloy classes *Int. J. Hydrog. Energy* **90** 460–9

[52] Verma A, Wilson N and Joshi K 2024 Solid state hydrogen storage: decoding the path through machine learning *Int. J. Hydrog. Energy* **50** 1518–28