
AGENT-BASED LEARNING OF MATERIALS DATASETS FROM SCIENTIFIC LITERATURE

Mehrad Ansari^{1,2} and Seyed Mohamad Moosavi^{*,1,2}

¹Acceleration Consortium, University of Toronto, Toronto, Ontario M5S 3E5, Canada

² Department of Chemical Engineering & Applied Chemistry, University of Toronto, Toronto, Ontario M5S 3E5, Canada

ABSTRACT

Advancements in machine learning and artificial intelligence are transforming materials discovery. Yet, the availability of structured experimental data remains a bottleneck. The vast corpus of scientific literature presents a valuable and rich resource of such data. However, manual dataset creation from these resources is challenging due to issues in maintaining quality and consistency, scalability limitations, and the risk of human error and bias. Therefore, in this work, we develop a chemist AI agent, powered by large language models (LLMs), to overcome these challenges by autonomously creating structured datasets from natural language text, ranging from sentences and paragraphs to extensive scientific research articles. Our chemist AI agent, *Eunomia*, can plan and execute actions by leveraging the existing knowledge from decades of scientific research articles, scientists, the Internet and other tools altogether. We benchmark the performance of our approach in three different information extraction tasks with various levels of complexity, including solid-state impurity doping, metal-organic framework (MOF) chemical formula, and property relations. Our results demonstrate that our zero-shot agent, with the appropriate tools, is capable of attaining performance that is either superior or comparable to the state-of-the-art fine-tuned materials information extraction methods. This approach simplifies compilation of machine learning-ready datasets for various materials discovery applications, and significantly ease the accessibility of advanced natural language processing tools for novice users in natural language. The methodology in this work is developed as an open-source software on <https://github.com/AI4ChemS/Eunomia>.

1 Introduction

The past decade's extraordinary achievements in leveraging machine learning for chemical discovery highlight the power of accessible knowledge and structured data [1–3]. However, a significant portion of chemical knowledge, particularly the experimental ones, is scattered across scientific literature in an unstructured format[4]. Researchers face challenges in effectively utilizing existing knowledge for design of experiments, as well as in comprehending the entirety of previous works in a field. Thus, the development of methodologies to extract information from the literature and convert it into structured data will play a fundamental role in advancing the machine learning for molecules and materials.

Natural Language Processing (NLP) is a powerful tool for extracting information from scientific literature. Conventional NLP methods have been used in materials and chemical sciences [5–10] for Named Entity Recognition. However, these methods are limited in other NLP tasks that are needed for a general-purpose data extraction tool, including Co-reference Resolution, Relation Extraction, Template Filling, Argument Mining, and Entity Linking. To better understand these NLP terminologies, let us consider an example taken from an abstract of a materials paper [11] in the field of metal-organic frameworks (MOFs):

*Corresponding author: mohamad.moosavi@utoronto.ca

An isoreticular series of cobalt-adeninate bio-MOFs (bio-MOFs-11–14) is reported. The pores of bio-MOFs-11–14 are decorated with acetate, propionate, butyrate, and valerate, respectively. The nitrogen (N_2) and carbon dioxide (CO_2) adsorption properties of these materials are studied and compared. The isosteric heats of adsorption for CO_2 are calculated, and the $CO_2 : N_2$ selectivities for each material are determined. As the lengths of the aliphatic chains decorating the pores in bio-MOFs-11–14 increase, the BET surface areas decrease from $1148\text{ m}^2\text{ g}^{-1}$ to $17\text{ m}^2\text{ g}^{-1}$ while the $CO_2 : N_2$ selectivities predicted from ideal adsorbed solution theory at 1 bar and 273 K for a 10 : 90 $CO_2 : N_2$ mixture range from 73 : 1 for bio-MOF-11 to 123 : 1 for bio-MOF-12 and finally to 107 : 1 for bio-MOF-13. At 298 K, the selectivities are 43 : 1 for bio-MOF-11, 52 : 1 for bio-MOF-12, and 40 : 1 for bio-MOF-13. The water stability of bio-MOFs-11–14 increases with increasing aliphatic chain length.

- **Named Entity Recognition** involves identifying and classifying the specific entities within the text into predefined categories (i.e., *chemical compounds*: “bio-MOFs-11–14”, “acetate”, *experimental conditions*: “1 bar”, “273 K”, “10 : 90 $CO_2 : N_2$ mixture”).
- **Co-reference Resolution** focuses on finding all expressions that refer to the same entity in the text. As an example, phrases like “these materials”, “each material” are references that relate back to the bio-MOFs-11–14 mentioned in the first sentence.
- **Relation Extraction** involves extracting semantic relationships from the text, which usually occur between two or more entities (i.e., the impact of “aliphatic chain lengths” on “BET surface areas” and “ $CO_2 : N_2$ selectivities”).
- **Template Filing** is an efficient approach to extract and structure complex information from text. As an example: *material name*: bio-MOFs-11–14.
- **Argument Mining** focuses on the automatic identification and extracts the reasoning presented within the text. As an example, the “increase in the water stability” of the mentioned MOFs are connected to the “increasing length of the aliphatic chains”.
- **Entity Linking** takes one step further than named entity recognition and distinguishes between similarly named entities (i.e., the term “bio-MOFs” would be linked to databases or literature that describe these materials in detail).

The emergence of Large Language Models (LLMs) or foundation models, shows a great promise in tackling these complex NLP tasks [7, 12–14]. Huang and Cole [15] fine-tuned a language model (BERT) on battery publications to extract device-level information from a paragraph that contains one device only. Dunn et al. [7] showed that fine-tuned LLMs using 100–1000 data points can perform Relation Extraction as well as template filling, enabling conversion of the extracted information into user-defined output formats. Despite these promising results, these methods require training data, limiting their ease of use and broad applicability. Moreover, LLM based approaches have not been explored for more intricate challenges, such as argument mining and co-reference resolution. These tasks are critical for practically using NLP for automated database development. For example, in one article, multiple materials might be discussed and authors use abbreviations like “compound 1” or simply “1” in the entire research manuscript for referencing after initially defining the chemical compound in the introduction section. Additionally, description of material properties often come with various interpretations, limiting using rigid name entity matching. As implementations of standalone LLMs fall short in addressing these intricate tasks, new methods are needed to enable reliable information extraction. An effective approach is to augment LLMs with domain-specific toolkits. These specialized tools offer precise answers, thus addressing the inherent limitations of LLMs in specific domains, and enhancing their overall performance and applicability [16–19].

In this work, we introduce an autonomous AI agent, *Eunomia*, augmented with chemistry-informed tools, designed to extract materials science-relevant information from unstructured text and convert it into structured databases. With an LLM at its core, our AI agent is capable of strategizing and executing actions by tapping into a wealth of knowledge from academic publications, domain-specific experts, the Internet, and other user-defined resources (see Figure 1). We show that this method streamlines data extraction, achieving remarkable accuracy and performance solely with a pre-trained LLM (GPT-4 [20]), eliminating the need for fine-tuning. It offers adaptability by accommodating a variety of information extraction tasks through natural language text prompts for new output schemas and reducing the risk of hallucinations through a chain-of-verification process. This capability extends beyond what a standalone LLM can offer. *Eunomia* simplifies the development of tailored datasets from literature for domain experts, eliminating the need for extensive programming, NLP, or machine learning expertise.

This manuscript is organized as follows; Benchmarking and evaluating the model performance on three different materials NLP tasks with varying level of complexity are represented in Section 3. This is followed by Section 4, with a discussion on the implications of our findings, the advantages and limitations of our approach, as well as suggested directions for future work. Finally, in Section 5, we describe our methodology on agent’s toolkits and evaluation metrics.

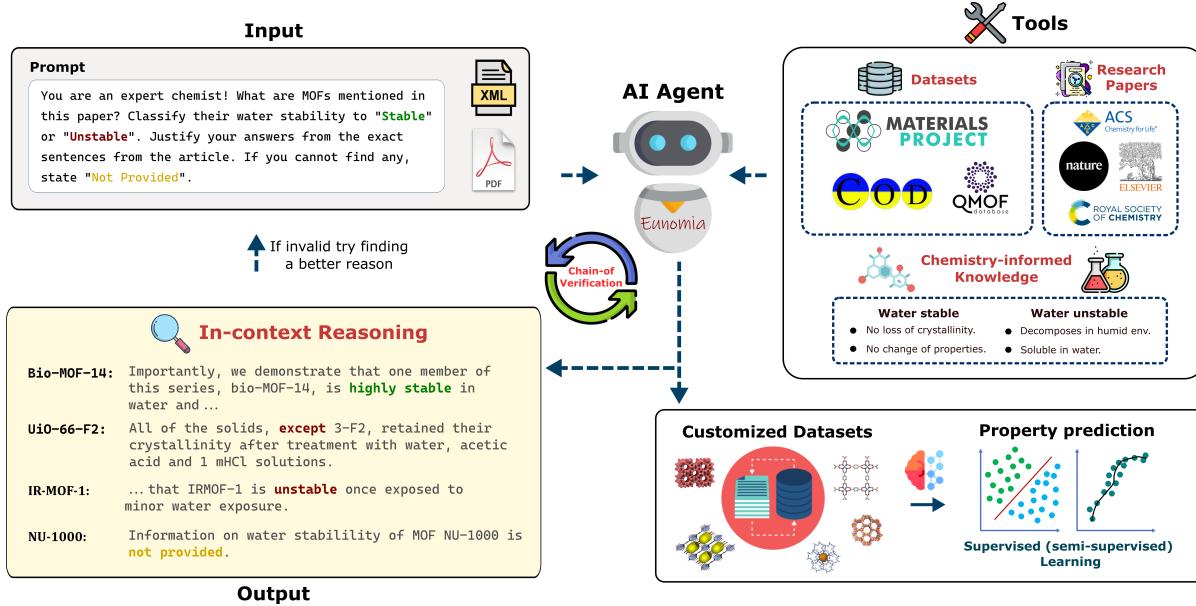


Figure 1: **Agent-based learning framework overview.** The AI agent equipped with various tools (online dataset search, document search, etc.) is tasked to extract information. The example shows the task of identifying all MOFs from a given research article, and predicting their property (e.g. water stability) by providing the reasoning for its decision. This reasoning is the exact in-context sentence from the paper, which is autonomously re-evaluated via the chain-of-verification tool of the agent to ensure its actual logical connection to the water stability property and reduce likelihood of hallucinations. The agent outputs a customized dataset that can be used to develop supervised or unsupervised machine learning methods.

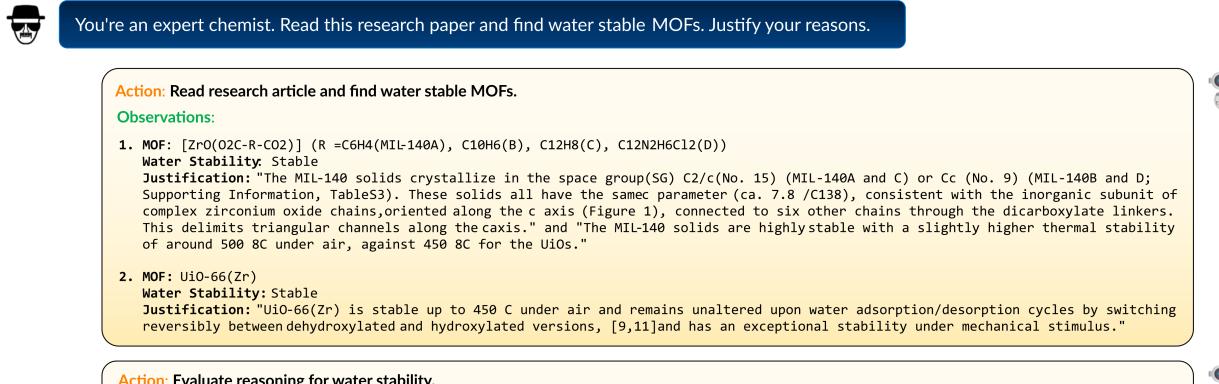
2 AI Agent

In the realm of artificial intelligence, an “agent” is an autonomous entity capable of taking action based on its environment. In this work, we developed a chemist AI agent, *Eunomia*, to autonomously extract information from scientific literature (Figure 1). We use an LLM to serve as the brain of our agent [21]. The LLM is equipped with advanced capabilities like planning and tool use to act beyond just a text generator, and act as a comprehensive problem solver, enabling effective interactions with the environment. We use ReAct architecture [22] for planning, enabling both reasoning and action. Our agent can interact with external sources like knowledge bases or environments to obtain more information. These knowledge bases are developed as toolkits (see method section for details) allowing the agent to extract relevant information from research articles, publicly available datasets, and built-in domain-specific chemical knowledge, ensuring its proficiency in playing the role of an expert chemist. We use OpenAI’s GPT-4 [20] with a temperature of zero as our LLM and LangChain [23] for the application framework development (note the choice of LLM is only a hyperparameter and other LLMs can be also used with our agent). The application of LLMs in developing autonomous agents is a growing area of research [17, 22, 24–27], with a detailed survey available in Ref. [28] for further insights.

In addition to the standard search and text manipulation tools, we have implemented a Chain-of-Verification (CoV) tool to enhance the robustness of our AI agent against hallucination. Hallucination in a LLM refers to the generation of content that strays from factual reality or includes fabricated information [29]. In the CoV approach, the agent iteratively assesses its responses to ensure they remain logically consistent and coherent (see method section 5.1.2 for details). This addition helps particularly with eliminating mistakenly extracted data related to semantically similar properties. An illustrative example is the case of stability of materials, where thermal, mechanical, and chemical stabilities might be confused by the agent. Figure 2 illustrates how CoV process works in action: the agent is tasked

to identify MOFs and the corresponding water stability data in a paper. The agent initially misclassifies a thermally stable MOF as water-stable, but then it corrects this mistake by a comprehensive review using the CoV tool. This tool improves the performance of the agent and ensures robust data extraction.

 Expert Human Chemist
AI Agent Chemist 

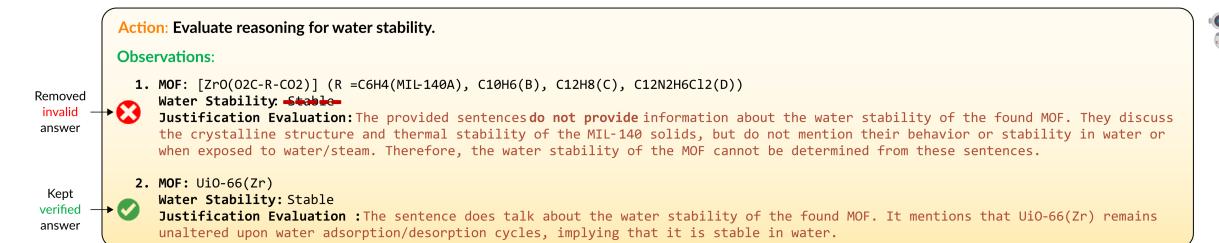


You're an expert chemist. Read this research paper and find water stable MOFs. Justify your reasons.

Action: Read research article and find water stable MOFs.

Observations:

- MOF:** [ZrO(O2C-R-CO2)] (R =C6H4(MIL-140A), C10H6(B), C12H8(C), C12N2H6Cl2(D))
Water Stability: Stable
Justification: "The MIL-140 solids crystallize in the space group(SG) C2/c(No. 15) (MIL-140A and C) or Cc (No. 9) (MIL-140B and D; Supporting Information, TableS3). These solids all have the samec parameter (ca. 7.8 /C138), consistent with the inorganic subunit of complex zirconium oxide chains,oriented along the c axis (Figure 1), connected to six other chains through the dicarboxylate linkers. This delimits triangular channels along the caxis." and "The MIL-140 solids are highlystable with a slightly higher thermal stability of around 500 8C under air, against 450 8C for the UIOs."
- MOF:** Uio-66(Zr)
Water Stability: Stable
Justification: "Uio-66(Zr) is stable up to 450 C under air and remains unaltered upon water adsorption/desorption cycles by switching reversibly between dehydroxylated and hydroxylated versions, [9,11]and has an exceptional stability under mechanical stimulus."

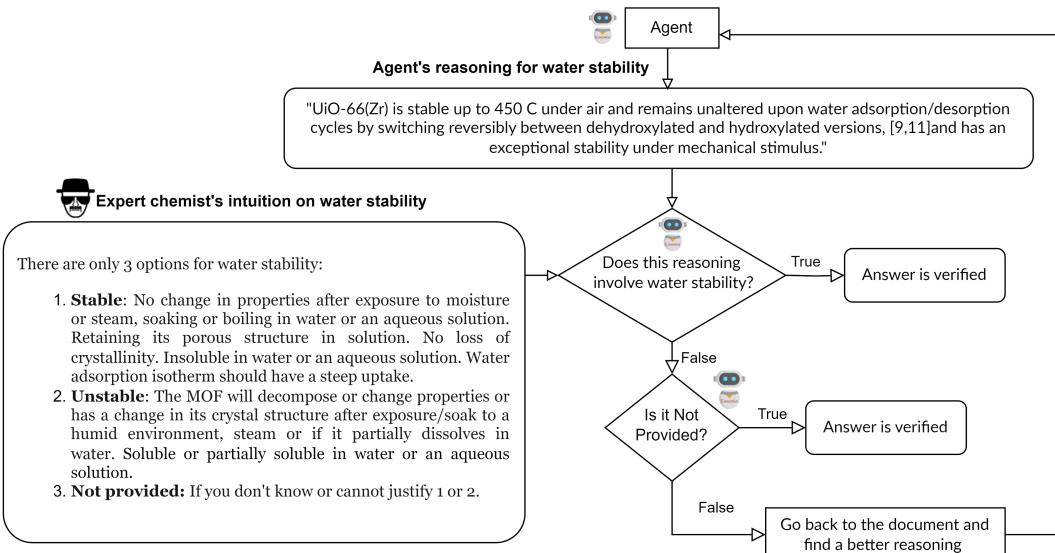


Action: Evaluate reasoning for water stability.

Observations:

Removed invalid answer →  1. MOF: [ZrO(O2C-R-CO2)] (R =C6H4(MIL-140A), C10H6(B), C12H8(C), C12N2H6Cl2(D))
Water Stability: ~~Stable~~
Justification Evaluation: The provided sentences do not provide information about the water stability of the found MOF. They discuss the crystalline structure and thermal stability of the MIL-140 solids, but do not mention their behavior or stability in water or when exposed to water/steam. Therefore, the water stability of the MOF cannot be determined from these sentences.

Kept verified answer →  2. MOF: Uio-66(Zr)
Water Stability: Stable
Justification Evaluation :The sentence does talk about the water stability of the found MOF. It mentions that Uio-66(Zr) remains unaltered upon water adsorption/desorption cycles, implying that it is stable in water.



```

graph TD
    Agent[Agent] --> Reasoning[Agent's reasoning for water stability]
    Reasoning --> Justification["'Uio-66(Zr) is stable up to 450 C under air and remains unaltered upon water adsorption/desorption cycles by switching reversibly between dehydroxylated and hydroxylated versions, [9,11]and has an exceptional stability under mechanical stimulus.'"]
    Justification --> Intuition[Expert chemist's intuition on water stability]
    Intuition --> Decision1{Does this reasoning involve water stability?}
    Decision1 -- True --> Verified1[Answer is verified]
    Decision1 -- False --> Decision2{Is it Not Provided?}
    Decision2 -- True --> Verified2[Answer is verified]
    Decision2 -- False --> GoBack[Go back to the document and find a better reasoning]
  
```

Expert chemist's intuition on water stability

There are only 3 options for water stability:

- Stable:** No change in properties after exposure to moisture or steam, soaking or boiling in water or an aqueous solution. Retaining its porous structure in solution. No loss of crystallinity. Insoluble in water or an aqueous solution. Water adsorption isotherm should have a steep uptake.
- Unstable:** The MOF will decompose or change properties or has a change in its crystal structure after exposure/soak to a humid environment, steam or if it partially dissolves in water. Soluble or partially soluble in water or an aqueous solution.
- Not provided:** If you don't know or cannot justify 1 or 2.

Figure 2: Iterative Chain of Verification (CoV). The agent is tasked with reading a materials research article and predicting water stability of any mentioned MOFs by providing reasoning. In the initial run, the agent confuses water stability with thermal stability and mistakenly predicts the second MOF as water-stable. The CoV tool evaluates the agent's decisions in its previous step by validating the justification against the pre-defined water stability criteria and disregards this prediction.

3 Case studies

We evaluate the performance of our AI agent by benchmarking it across three different materials NLP tasks, with increasing task complexity (Table 1). In our assessment, we include a wide range of text lengths, including sentences, paragraphs and entire manuscript, as well as different NLP tasks discussed in the introduction. The first case study focuses on assessing the agent’s performance on NLP tasks of lower complexity, specifically Named Entity Recognition and Relation Extraction. For this, we use our agent to extract host-to-many dopants relationships from a single sentence. The second case study, with medium NLP complexity, involves obtaining MOFs’ chemical formula and their corresponding guest species from a paragraph with multiple sentences. Finally, the third case study centers on predicting a given property of MOFs based on the context coming from a materials research paper. The property of interest in our work is water stability. This case study aims to, in addition to Named Entity Recognition and Relation Extraction, evaluate the co-reference resolution and argument mining proficiency of our AI agent, tailored for chemists, which involves a high level of NLP complexity. In all case studies, our chemist AI agent, Eunomia, is a zero-shot learner that is equipped with the Doc Search tool (see Section 5.1.1). We have also conducted additional experiments by equipping Eunomia with the chain-of-verification (CoV) tool, as described in Section 5.1.2. This is referred to as Eunomia + CoV from here on.

To fairly compare the performance of our agent with the state-of-the-art fine-tuned LLM methods, the evaluation methodology for the first two case studies mirrors precisely that of Dunn et al. [7] (see Section 5.2 for details), serving as a benchmark reference. In this study, Dunn et al. [7] is referred to as LLM-NERRE, which involves fine-tuning a pre-trained LLM (GPT-3 [30]) on 100-1000 set of manually annotated examples, and then using the model to extract information from unstructured text.

Table 1: Overview of the three case studies based on their context from which data is extracted, NLP tasks and complexity

Case Study	Context	NLP Tasks	Task Complexity
1. Host-to-many Dopants Relation	Sentence	Named Entity Recognition, Relation Extraction	Easy
2. MOF Formula and Guest Species Relation	Paragraph	Named Entity Recognition, Relation Extraction	Medium
3. MOF Property Relation	Research paper	Named Entity Recognition, Relation Extraction, Template Filling, Argument Mining, Entity Linking	Hard

3.1 Case Study 1: Host-to-many Dopants Relation Easy

This case study aims to extract structured information about solid-state impurity doping from a single sentence. The objective is to identify the two entities “host” and “dopant”. “Host” refers to the foundational crystal, sample, or category of material characterized by essential descriptors in its proximate context, such as “ZnO₂ nanoparticles”, “LiNbO₃”, or “half-Heuslers”. “Dopant” means the elements or ions that represent a minority component, deliberately introduced impurities, or specific atomic-scale defects or carriers of electric charge like “hole-doped” or “S vacancies”. A single host can be combined with multiple dopants, through individual doping or simultaneous doping with different species, and one dopant can associate with various host materials. The text may contain numerous dopant-host pairings within the same sentence, and also instances of dopants and hosts that do not interact.

Eunomia shows an excellent performance in this task, demonstrating the effectiveness of our approach in Named Entity Recognition and Relation Extraction. Performance comparison between our chemist AI agent (Eunomia), and LLM-NERRE can be found in Table 2. In this setting, the same above definition of hosts and dopants are passed to Eunomia via the input prompt, while LLM-NERRE is fine-tuned on 413 sentences. The testing set contains 77 sentences. Notably, in both tasks, Eunomia + CoV exceeds the performance of LLM-NERRE in terms of the F1 score. This clearly demonstrates the effectiveness of our approach compared to fine-tuning, which can be labor-intensive and error-prone. We instruct Eunomia not to make up answers, which lead to a more cautious outcome, wherein uncertain or unclear inputs yield no output. As an example in the sentence “An anomalous behavior of the emission properties of alkali halides doped with heavy impurities, stimulated new efforts for its interpretation, invoking delicate and sophisticated mechanisms whose interest transcends the considered specific case.”, the ground-truth host materials is “alkali halides”. However, due to the nature of exact-word matching metric implemented in Ref. [7] a cautious agent with no predictions

Table 2: Performance comparison between LLM-NERRE, Eunomia, and Eunomia + CoV on hosts and dopants relation extraction (Case Study 1). Eunomia embeddings are generated using OpenAI’s text-ada-002. Best scores for each entity are highlighted in bold text.

Model	Entity	Precision (Exact Match)	Recall (Exact Match)	F1 Score (Exact Match)
LLM-NERRE [7]	hosts	0.892	0.874	0.883
Eunomia	hosts	0.753	0.768	0.760
Eunomia + CoV	hosts	0.964	0.853	0.905

LLM-NERRE [7]	dopants	0.831	0.812	0.821
Eunomia	dopants	0.859	0.788	0.822
Eunomia + CoV	dopants	0.962	0.882	0.920

for the host entity will be penalized with two false negatives, one for each word in the ground-truth, leading to lower recall score.

3.2 Case Study 2: MOF Formula and Guest Species Relation Medium

The goal of this case study is to identify MOF formula and guest species from unstructured text, as a paragraph with multiple sentences. The MOF formula refers to the chemical formula of a MOF, which is an important piece of information for characterizing and identifying MOFs. The guest species, on the other hand, are chemical species that have been incorporated, stored, or adsorbed in the MOF. These species are of interest because MOFs are often used for ion and gas separation, and the presence of specific guest species can affect the performance of the MOF. We limit our method to stand-alone Eunomia without CoV due to the complexity of defining a chemistry-informed CoV verification tool for this specific task. It should be noted that Dunn et al. [7] also included results on the identification of synthesis descriptions and applications pertaining to MOFs. However, as the metric of exact-word matching reported in Dunn et al. [7] does not fairly and adequately reflect the model performance for the multi-word (> 2 words) nature of these outputs, we have limited our benchmarking to the MOF formula and guest species identification only.

Table 3: Performance comparison between LLM-NERRE and Eunomia on MOF formula and guest species relation extraction (Case study 2). Eunomia embeddings are generated using OpenAI text-ada-002. Best scores for each entity are highlighted in bold text.

Model	Entity	Precision (Exact Match)	Recall (Exact Match)	F1 Score (Exact Match)
LLM-NERRE [7]	mof formula	0.409	0.455	0.424
Eunomia	mof formula	0.623	0.589	0.606

LLM-NERRE [7]	guest species	0.588	0.665	0.606
Eunomia	guest species	0.429	0.923	0.585

Table 3 shows the performance comparison between Eunomia and LLM-NERRE on the MOF formula and guest species relation extraction task. While Eunomia shows a superior performance on the MOF formula compared to LLM-NERRE, the relatively low performance of both approaches is related to the nature of the exact word matching. Using semantic similarity would be a more appropriate indicator in this context. On the guest species entity, while Eunomia shows a high recall (0.923), precision is relatively poor (0.429). This can be attributed to how the exact-word matching metrics have been defined in Ref. [7], where precision is majorly lowered by the presence of the extra unmatched predicted words (false positives), while recall remains high because all ground truth items were found in the predicted words.

3.3 Case Study 3: MOF Property Relation Hard

This case study aims to mimic a practical scenario of developing datasets from scientific literature, where we evaluate the agent’s performance on extracting MOF’s water stability. To excel in this goal, the agent must identify all MOFs mentioned within the research paper, evaluate their water stability, and support these evaluations using exact sentences derived from the document. Such tasks are inherently linked to the NLP functions of Named Entity Recognition, Co-reference Resolution, Relation Extraction, and Argument Mining. This is particularly a challenging task as researcher report the water stability in various ways, using phrases ranging from “the material remains crystalline in humid conditions” to “the MOF is stable in wide range of pH”, or “the material is not soluble in water”.

For this case study, we created a hand-labeled dataset based on a selection of 101 materials research papers, which contain a selection of 371 MOFs. Three expert chemists manually read through and review each paper, pinpointing the MOFs referenced within. A portion of these articles are selected considering the original work by Burch et al. [31], where they developed a dataset of MOF names and their water stability by manually reading 111 research articles. To mimic the practical data extraction scenario, in which the agent is passed many articles, many of which do not contain the desired information, we included articles with no information about water stability. Each MOF in our set is assigned to one of the three classes of "Stable", "Unstable", and "Not provided". Figure 3.a presents the distribution of the classes within this dataset.

For this case study, we have established criteria to characterize water-stable MOFs, drawing from the study by Burch et al. [31] and our own chemical intuition. A water-stable MOF should meet the following criteria:

- No alteration in properties after being subjected to moisture or steam, or when soaked or boiled in water or an aqueous solution.
- Preservation of its porous architecture in liquid (water) environments.
- Sustained crystallinity without loss of structural integrity in aqueous environments.
- Insoluble in water or aqueous solutions.
- Exhibiting a pronounced rise in its water adsorption isotherm.

These water stability guidelines are defined as rules to Eunomia within the input prompt, as well as in its equipped CoV tool.

Eunomia with CoV tool retrieve most (yield of 86.20%) of the reported MOFs and shows an excellent performance (accuracy of 0.91) in inferring their water stability. This high yield and accuracy demonstrates the capability of our approach in extracting desired knowledge from the natural text. As expected, in the absence of CoV, there is a marginal decrease in accuracy to 0.86, along with a yield reduction to 82.70%. Taking into account the confusion matrix in Figure 3.b, it is evident that our agent adopts a cautious approach in its predictions. This is reflected in the substantial number of "Not provided" predictions which, upon comparison with the actual ground-truth class, indicates a propensity of the agent to acknowledge the insufficiency of information for making a definitive prediction, rather than mistakenly categorizing samples into the incorrect "Stable" or "Unstable" classes, and contaminating the agent's resulting dataset with unwanted noise.

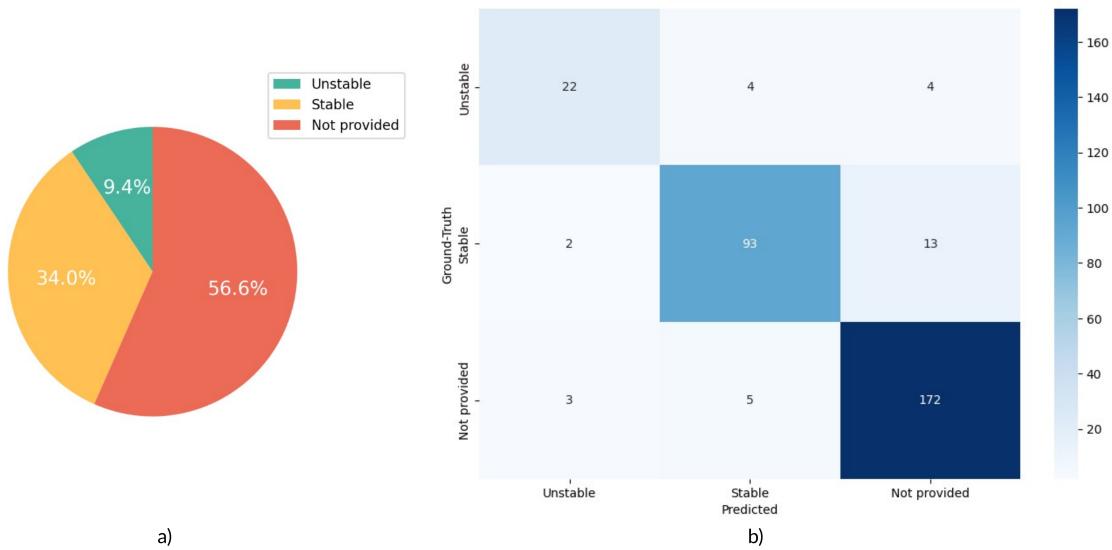


Figure 3: Performance of the AI agent in information retrieval. a) Class distribution for water stability in the hand-labeled ground-truth dataset of 371 MOFs based on 101 research articles. b) Confusion matrix for ternary classification of water stability property with CoV tool using OpenAI text-ada-002 embeddings. It is apparent that our agent exercises caution in its judgments. Specifically, the abundance of "Not provided" predictions, when matched against their actual ground-truth categories, suggests that the agent prefers to concede some uncertainty in instances where making an accurate prediction is not feasible, rather than incorrectly assigning samples to the "Stable" or "Unstable" categories. The ternary accuracy is found to be 0.91 with a yield of 86.20%.

4 Discussion

We presented a high performing and robust method for extracting domain specific information from complex, unstructured text - ranging from sentences and paragraphs to extensive research articles - using AI agents. Scientists and researchers can use our open-source application to effortlessly develop tailored datasets for their specific areas and use them for downstream predictive tasks[32]. While currently the cost of querying large datasets may become expensive, we expect the rapid advancements in LLMs will diminish this cost.

Unlike other methods that follow a pipeline-based or end-to-end approach, our agent-based method could appeal to domain experts due to its minimal demand for programming skills, NLP and machine learning knowledge. Users are not required to rigidly define an output schema or engage in the meticulous task of creating manual annotations for the purpose of fine-tuning. Rather, they can simply prompt the agent with more context and describe how their desired output should be formatted in natural language. Moreover, the agent can easily be extended and equipped with other tools (e.g. Dataset Search, CSV Generator, etc.) to be adapted to other problems. For example, we showed that, by equipping the agent with the chain-of-verification tool (CoV), we can minimize Hallucinations and improve the agent’s performance. Similarly, by including reasoning tools, we can ask the agent to explain its reasoning based on the provided context to develop more transparent workflows for the LLM-based methods, and reduce their known “black-box” nature. This, simultaneously, offers a great opportunity for human-in-the-loop oversight, especially for tasks of critical importance.

Our results reveal an important observation: while *large language models are few-shot learners* [30], AI agents with appropriate tools and instructions are capable of being zero-shot learners. This brings an excellent opportunity to boost the performance of standalone LLMs across various domain-specific tasks without having to go through labor-intensive fine-tuning processes. A future thorough and systematic analysis of prompt sensitivity can provide valuable insights into this observation.

5 Methods

5.1 Agent Toolkits

5.1.1 Doc Search

This tool allows for extracting relevant knowledge materials properties from text, ranging from a single sentence and paragraph to a scientific research paper. The research papers are obtained from various chemistry journals including Royal Society of Chemistry (RSC), American Chemical Society (ACS), Elsevier, Inorganic Chemistry, Structural Chemistry, Coordination Chemistry, Wiley, and Crystallographic Communications as a PDF or in XML format (the XML files are obtained through a legal agreement between University of Toronto and ACS). Inspired by the paper-qa Python package (<https://github.com/whitead/paper-qa>), this tool aims at obtaining the most relevant context (sentences) from the papers to a given input query. This involves embedding the paper and queries into numerical vectors and identifying top k passages within the document that either mention or can somehow imply the property of interest for a MOF. k is set to 9 in our case studies, and is dynamically adjusted depending on the length of the paper to avoid OpenAI’s token limitation error. We use OpenAI’s text-ada-002 embeddings [33] to represent texts as high dimensional vectors, which are stored as a vector database using FAISS [34]. Note that the choice of embedding is another hyperparameter that can be changed in future studies. For benchmarking purposes, we have also conducted all case studies with the newly released Cohere embed-english-v3.0 embeddings (see Supporting Information).

The semantic similarity search is ranked using Maximum Marginal Relevance (MMR) [35] based on cosine similarity, defined as,

$$\text{MMR} = \arg \max_{d_i \in R \setminus S} \left[\lambda \cdot \cos(d_i, q) - (1 - \lambda) \cdot \max_{d_j \in S} \cos(d_i, d_j) \right] \quad (1)$$

where d_i is a document from the set of retrieved documents R , S is the set of already selected documents, q is the query. λ is a parameter between 0 and 1 that balances the trade-off between relevance (to the query) and diversity (or novelty with respect to already selected documents). In this work, we use the default value of 0.5. The idea behind MMR is to retrieve or select documents that are not just relevant to the query (or topic of interest), but are also diverse among themselves, thus minimizing redundancy. This tool provides the exact in-context sentence from the paper that provides some reasoning for the agent’s choice, allowing for a more methodical evaluation of the agent’s decision-making process, and reducing the likelihood of hallucinations or fabrications with the human-in-loop verification of the resulting datasets. It is important to note that in some unsuccessful experiments, we observed that the AI agent repeatedly referred back to the document, even after pinpointing the correct answer. Although this minor issue remained unresolved, we introduced an iteration limit for the agent to avoid unnecessary model running costs.

5.1.2 Chain-of-Verification

Inspired by the Chain-of-Verification (CoV) [36] methodology, this tool entails the following steps: initially, the agent provides a preliminary reply, which is followed by iterative verification queries to authenticate the initial draft. The agent independently responds to these queries to ensure the answers remain impartial and unaffected by other responses, and finally it produces its conclusive, verified response. Our implementation of CoV stands apart from the method described in Dhuliawala et al. [36], specifically in how the verification queries are generated. While in the Dhuliawala et al. [36]’s approach, the LLM produces task-specific queries, our method allows for user customization. This adaptability not only enables broader, more tailored domain-specific fact-checking across various tasks, but also opens up opportunities for human-in-the-loop verification, enhancing the accuracy and relevance of the results. This tool substantially boosts agent efficacy and mitigates the likelihood of hallucinations, especially in the events of completing complex tasks (see Figure 2 for more details). It is important to note that for unknown reasons, we have observed that the CoV tool usage was skipped by the agent on a few occasional instances.

5.1.3 Dataset Search

This tool allows for obtaining the chemical structure of MOFs from publically available datasets, including the Materials Projects [37], Crystallography Open Database (COD) [38–45], Cambridge Structural Database (CSD) [46], and QMOF [47, 48].

5.1.4 CSV Generator

This tool stores the output of the agent into a CSV or JSON file.

5.2 Evaluation Metrics

Multiple metrics have been defined to assess agent’s performance across different case studies. Precision, recall and F1 score are defined as,

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (2)$$

$$\text{Recall} = \frac{TP}{TP + FN}, \quad (3)$$

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \quad (4)$$

where, TP represents true positives, FP stands for false positives, and FN denotes false negatives. Precision measures the accuracy of the positive predictions, recall measures the fraction of actual positives that were correctly identified, and the F1 score is the harmonic mean of precision and recall. Binary classification accuracy is defined as,

$$\text{Binary accuracy} = \frac{TP + TN}{N}, \quad (5)$$

In Case studies 1 and 2 (Sections 3.1 and 3.2), the evaluation metrics used are precisely those defined in the work of [7]. In specific, they assessed named entity relationships on a word-to-word matching basis by initially decomposing an entity E into a collection of k words separated by whitespace, denoted as $E = \{w_1, w_2, w_3, \dots, w_k\}$. For evaluating entities in Named Entity Recognition exclusively, they enumerated the words that are identical in both the true entity set E_{true} and the test entity set E_{test} as true positives ($E_{\text{true}} \cap E_{\text{test}}$), and the distinct elements in each set as false positives ($E_{\text{test}} \setminus E_{\text{true}}$) or false negatives ($E_{\text{true}} \setminus E_{\text{test}}$). For instance, if the true entity is “ Bi_2Te_3 thin film” and the predicted entity is “ Bi_2Te_3 film sample”, they noted two true positive matches (“ Bi_2Te_3 ”, “film”), one false negative (“thin”), and one false positive (“sample”). An exceptional case arises for formula-type entities critical to material identification, whereby E_{test} must encompass all w_i interpreted as stoichiometries to consider any $w_i \in E_{\text{test}}$ as correct. For example, with “ Bi_2Te_3 thin film” as E_{true} and “thin film” as E_{test} , three false negatives would be registered. For more details on the scoring metrics and the case studies, readers are encouraged to refer to Ref. [7].

For our last case study in Section 3.3 (predicting water stability of MOFs), the ternary accuracy is defined as,

$$\text{Ternary accuracy} = \frac{TP_S + TP_U + TP_{NP}}{N}, \quad (6)$$

where N shows the total number of predictions and S, U, NP denote the three classes “Stable”, “Unstable”, and “Not provided”, respectively. TP_i shows then number of instances correctly predicted as class i . Additionally, we evaluate the information recovery capabilities of the agent by defining yield as

$$\text{Yield} = \frac{N}{N_{GT}}, \quad (7)$$

where N_{GT} is the *ground-truth* number of MOFs mentioned in the research papers, regardless whether the paper discusses water stability or not.

Acknowledgements

Research reported in this work was supported by the Acceleration Consortium at the University of Toronto. SMM acknowledges the support by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grant number RGPIN-2023-04232. The authors thank Alexander Dunn and Andrew S. Rosen for their assistance on the implementation of LLM-NERRE. The authors also thank Haoning Yuan for assisting in manual curation of the water stability dataset used in Section 3.3.

Data and Code Availability

All data (including the dataset used for case study 3) and code used to produce results in this study are publically available in the following GitHub repository: <https://github.com/AI4ChemS/Eunomia>. The methodology in this work is also developed as an open-source application on <https://eunomia.streamlit.app>.

References

- [1] Seyed Mohamad Moosavi, Kevin Maik Jablonka, and Berend Smit. The role of machine learning in the understanding and design of materials. *Journal of the American Chemical Society*, 142(48):20273–20287, 2020.
- [2] Keith T Butler, Daniel W Davies, Hugh Cartwright, Olexandr Isayev, and Aron Walsh. Machine learning for molecular and materials science. *Nature*, 559(7715):547–555, 2018.
- [3] Benjamin Sanchez-Lengeling and Alán Aspuru-Guzik. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365, 2018.
- [4] Hasan M Sayeed, Wade Smallwood, Sterling G Baird, and Taylor D Sparks. Quantifying the distribution of materials data types in scientific literature across text, tables, and figures. 2023.
- [5] Leigh Weston, Vahe Tshitoyan, John Dagdelen, Olga Kononova, Amalie Trewartha, Kristin A Persson, Gerbrand Ceder, and Anubhav Jain. Named entity recognition and normalization applied to large-scale information extraction from the materials science literature. *Journal of chemical information and modeling*, 59(9):3692–3702, 2019.
- [6] Matthew C Swain and Jacqueline M Cole. Chemdataextractor: a toolkit for automated extraction of chemical information from the scientific literature. *Journal of chemical information and modeling*, 56(10):1894–1904, 2016.
- [7] Alexander Dunn, John Dagdelen, Nicholas Walker, Sanghoon Lee, Andrew S Rosen, Gerbrand Ceder, Kristin Persson, and Anubhav Jain. Structured information extraction from complex scientific text with fine-tuned large language models. *arXiv preprint arXiv:2212.05238*, 2022.
- [8] Aditya Nandy, Chenru Duan, and Heather J Kulik. Using machine learning and data mining to leverage community knowledge for the engineering of stable metal–organic frameworks. *Journal of the American Chemical Society*, 143(42):17535–17547, 2021.
- [9] Edward Kim, Kevin Huang, Adam Saunders, Andrew McCallum, Gerbrand Ceder, and Elsa Olivetti. Materials synthesis insights from scientific literature via text extraction and machine learning. *Chemistry of Materials*, 29(21):9436–9444, 2017.
- [10] Lawson T Glasby, Kristian Gubsch, Rosalee Bence, Rama Oktavian, Kesler Isoko, Seyed Mohamad Moosavi, Joan L Cordiner, Jason C Cole, and Peyman Z Moghadam. Digimof: A database of metal–organic framework synthesis information generated via text mining. *Chemistry of Materials*, 2023.
- [11] Tao Li, De-Li Chen, Jeanne E Sullivan, Mark T Kozlowski, J Karl Johnson, and Nathaniel L Rosi. Systematic modulation and enhancement of co 2: N 2 selectivity and water stability in an isoreticular series of bio-mof-11 analogues. *Chemical Science*, 4(4):1746–1755, 2013.

- [12] Maciej P Polak and Dane Morgan. Extracting accurate materials data from research papers with conversational language models and prompt engineering. *arXiv preprint arXiv:2303.05352*, 2023.
- [13] Zhiling Zheng, Oufan Zhang, Christian Borgs, Jennifer T Chayes, and Omar M Yaghi. Chatgpt chemistry assistant for text mining and prediction of mof synthesis. *arXiv preprint arXiv:2306.11296*, 2023.
- [14] Zhiling Zheng, Zhiguo He, Omar Khattab, Nakul Rampal, Matei A Zaharia, Christian Borgs, Jennifer T Chayes, and Omar M Yaghi. Image and data mining in reticular chemistry using gpt-4v. *arXiv preprint arXiv:2312.05468*, 2023.
- [15] Shu Huang and Jacqueline M Cole. Batterybert: A pretrained language model for battery database enhancement. *Journal of Chemical Information and Modeling*, 62(24):6365–6377, 2022.
- [16] Andrew D White, Glen M Hocky, Heta A Gandhi, Mehrad Ansari, Sam Cox, Geemi P Wellawatte, Subarna Sasmal, Ziyue Yang, Kangxin Liu, Yuvraj Singh, et al. Assessment of chemistry knowledge in large language models that generate code. *Digital Discovery*, 2(2):368–376, 2023.
- [17] Andres M Bran, Sam Cox, Andrew D White, and Philippe Schwaller. Chemcrow: Augmenting large-language models with chemistry tools. *arXiv preprint arXiv:2304.05376*, 2023.
- [18] Daniil A Boiko, Robert MacKnight, and Gabe Gomes. Emergent autonomous scientific research capabilities of large language models. *arXiv preprint arXiv:2304.05332*, 2023.
- [19] Jakub Lála, Odhran O’Donoghue, Aleksandar Shtedritski, Sam Cox, Samuel G Rodrigues, and Andrew D White. Paperqa: Retrieval-augmented generative agent for scientific research. *arXiv preprint arXiv:2312.07559*, 2023.
- [20] OpenAI. Gpt-4 technical report, 2023.
- [21] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.
- [22] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.
- [23] Harrison Chase. Langchain, 10 2022. URL <https://github.com/langchain-ai/langchain>.
- [24] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.
- [25] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*, 2023.
- [26] Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. Chameleon: Plug-and-play compositional reasoning with large language models. *arXiv preprint arXiv:2304.09842*, 2023.
- [27] Yujia Qin, Shengding Hu, Yankai Lin, Weize Chen, Ning Ding, Ganqu Cui, Zheni Zeng, Yufei Huang, Chaojun Xiao, Chi Han, et al. Tool learning with foundation models. *arXiv preprint arXiv:2304.08354*, 2023.
- [28] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*, 2023.
- [29] Vipula Rawte, Amit Sheth, and Amitava Das. A survey of hallucination in large foundation models. *arXiv preprint arXiv:2309.05922*, 2023.
- [30] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- [31] Nicholas C Burtsch, Himanshu Jasuja, and Krista S Walton. Water stability and adsorption in metal–organic frameworks. *Chemical reviews*, 114(20):10575–10612, 2014.
- [32] Kevin Maik Jablonka, Philippe Schwaller, Andres Ortega-Guerrero, and Berend Smit. Leveraging large language models for predictive chemistry. 2023.
- [33] Ryan Greene, Ted Sanders, Lilian Weng, and Arvind Neelakantan. New and improved embedding model, 2022.
- [34] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

- [35] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336, 1998.
- [36] Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models. *arXiv preprint arXiv:2309.11495*, 2023.
- [37] Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, et al. Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL materials*, 1(1), 2013.
- [38] Andrius Merkys, Antanas Vaitkus, Algirdas Grybauskas, Aleksandras Konovalovas, Miguel Quirós, and Saulius Gražulis. Graph isomorphism-based algorithm for cross-checking chemical and crystallographic descriptions. *Journal of cheminformatics*, 15(1):25, 2023.
- [39] Antanas Vaitkus, Andrius Merkys, and Saulius Gražulis. Validation of the crystallography open database using the crystallographic information framework. *Journal of applied crystallography*, 54(2):661–672, 2021.
- [40] Miguel Quirós, Saulius Gražulis, Saulė Girdzijauskaitė, Andrius Merkys, and Antanas Vaitkus. Using smiles strings for the description of chemical connectivity in the crystallography open database. *Journal of cheminformatics*, 10(1):1–17, 2018.
- [41] Andrius Merkys, Antanas Vaitkus, Justas Butkus, Mykolas Okulič-Kazarinas, Visvaldas Kairys, and Saulius Gražulis. Cod:: Cif:: Parser: an error-correcting cif parser for the perl language. *Journal of applied crystallography*, 49(1):292–301, 2016.
- [42] Saulius Gražulis, Andrius Merkys, Antanas Vaitkus, and Mykolas Okulič-Kazarinas. Computing stoichiometric molecular composition from crystal structures. *Journal of applied crystallography*, 48(1):85–91, 2015.
- [43] Saulius Gražulis, Adriana Daškevič, Andrius Merkys, Daniel Chateigner, Luca Lutterotti, Miguel Quiros, Nadezhda R Serebryanaya, Peter Moeck, Robert T Downs, and Armel Le Bail. Crystallography open database (cod): an open-access collection of crystal structures and platform for world-wide collaboration. *Nucleic acids research*, 40(D1):D420–D427, 2012.
- [44] Saulius Gražulis, Daniel Chateigner, Robert T Downs, AFT Yokochi, Miguel Quirós, Luca Lutterotti, Elena Manakova, Justas Butkus, Peter Moeck, and Armel Le Bail. Crystallography open database—an open-access collection of crystal structures. *Journal of applied crystallography*, 42(4):726–729, 2009.
- [45] Robert T Downs and Michelle Hall-Wallace. The american mineralogist crystal structure database. *American Mineralogist*, 88(1):247–250, 2003.
- [46] Colin R Groom, Ian J Bruno, Matthew P Lightfoot, and Suzanna C Ward. The cambridge structural database. *Acta Crystallographica Section B: Structural Science, Crystal Engineering and Materials*, 72(2):171–179, 2016.
- [47] Andrew S Rosen, Shaelyn M Iyer, Debmalya Ray, Zhenpeng Yao, Alan Aspuru-Guzik, Laura Gagliardi, Justin M Notestein, and Randall Q Snurr. Machine learning the quantum-chemical properties of metal–organic frameworks for accelerated materials discovery. *Matter*, 4(5):1578–1597, 2021.
- [48] Andrew S Rosen, Victor Fung, Patrick Huck, Cody T O’Donnell, Matthew K Horton, Donald G Truhlar, Kristin A Persson, Justin M Notestein, and Randall Q Snurr. High-throughput predictions of metal–organic framework electronic properties: theoretical challenges, graph neural networks, and data exploration. *npj Computational Materials*, 8(1):112, 2022.

SUPPORTING INFORMATION FOR AGENT-BASED LEARNING OF MATERIALS DATASETS FROM SCIENTIFIC LITERATURE

Mehrad Ansari^{1,2} and Seyed Mohamad Moosavi^{*,1,2}

¹Acceleration Consortium, University of Toronto, Toronto, Ontario M5S 3E5, Canada

² Department of Chemical Engineering & Applied Chemistry, University of Toronto, Toronto, Ontario M5S 3E5, Canada

1 Supporting Figures

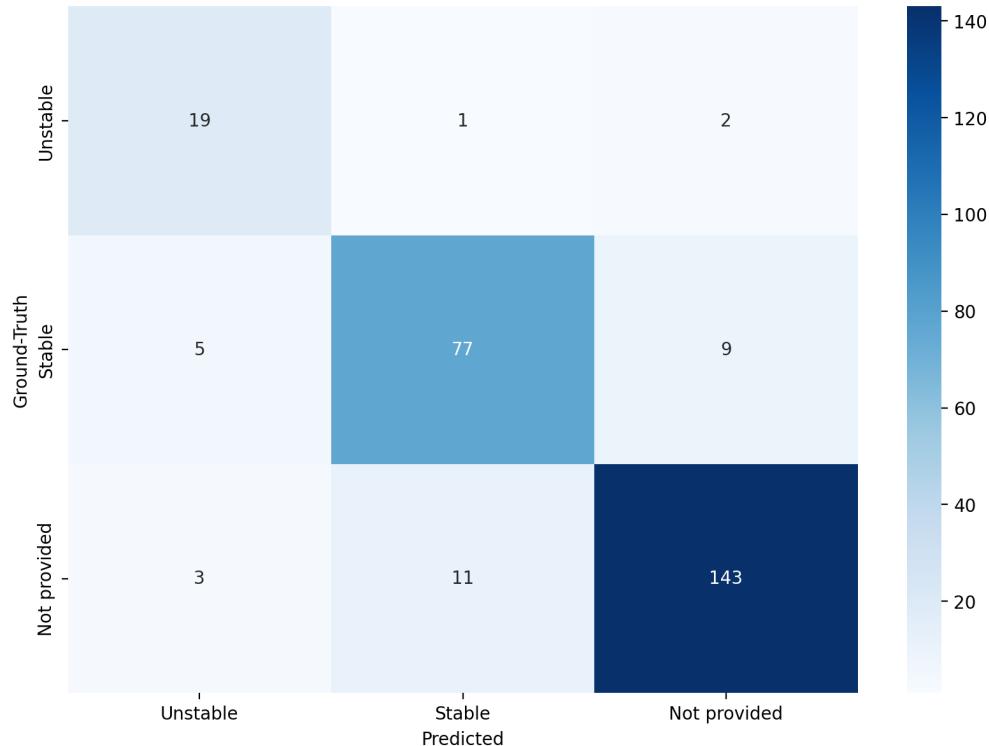


Figure S1: Confusion matrix for ternary classification of water stability property in Case Study 3 using Cohere embed-english-v3.0 embeddings. It is apparent that our agent exercises caution in its judgments. Specifically, the abundance of "Not provided" predictions, when matched against their actual ground-truth categories, suggests that the agent prefers to concede some uncertainty in instances where making an accurate prediction is not feasible, rather than incorrectly assigning samples to the "Stable" or "Unstable" categories. The ternary accuracy is found to be 0.88 with a yield of 72.96%, which is lower than performance coming from OpenAI's text-ada-002 embeddings (see Figure 4b in the main article).

* Corresponding author: mohamad.moosavi@utoronto.ca

2 Supporting Tables

Table S1: Performance comparison between LLM-NERRE, Eunomia, and Eunomia + CoV on hosts and dopants relation extraction (Case Study 1). Eunomia embeddings are generated using Cohere embed-english-v3.0.

Model	Entity	Precision (Exact Match)	Recall (Exact Match)	F1 Score (Exact Match)
LLM-NERRE	hosts	0.892	0.874	0.883
Eunomia	hosts	0.710	0.747	0.728
Eunomia + CoV	hosts	0.793	0.726	0.758

LLM-NERRE	dopants	0.831	0.812	0.821
Eunomia	dopants	0.782	0.800	0.791
Eunomia + CoV	dopants	0.863	0.812	0.836

Table S2: Performance comparison between LLM-NERRE and Eunomia on MOF formula and guest species relation extraction (Case study 2). Eunomia embeddings are generated using Cohere embed-english-v3.0.

Model	Entity	Precision (Exact Match)	Recall (Exact Match)	F1 Score (Exact Match)
LLM-NERRE	mof formula	0.409	0.455	0.424
Eunomia	mof formula	0.615	0.571	0.593

LLM-NERRE	guest species	0.588	0.665	0.606
Eunomia	guest species	0.213	0.769	0.333