Extracting Accurate Materials Data from Research Papers with Conversational Language Models and Prompt Engineering

Maciej P. Polak* and Dane Morgan†
Department of Materials Science and Engineering, University of Wisconsin-Madison, Madison, Wisconsin 53706-1595, USA

There has been a growing effort to replace hand extraction of data from research papers with automated data extraction based on natural language processing, language models, and recently, large language models (LLMs). Although these methods enable efficient extraction of data from large sets of research papers, they require a significant amount of up-front effort, expertise, and coding. In this work we propose the `ChatExtract` method that can fully automate very accurate data extraction with minimal initial effort and background, using an advanced conversational LLM. `ChatExtract` consists of a set of engineered prompts applied to a conversational LLM that both identify sentences with data, extract that data, and assure the data's correctness through a series of follow-up questions. These follow-up questions largely overcome known issues with LLMs providing factually inaccurate responses. `ChatExtract` can be applied with any conversational LLMs and yields very high quality data extraction. In tests on materials data we find precision and recall both close to 90% from the best conversational LLMs, like ChatGPT-4. We demonstrate that the exceptional performance is enabled by the information retention in a conversational model combined with purposeful redundancy and introducing uncertainty through follow-up prompts. These results suggest that approaches similar to `ChatExtract`, due to their simplicity, transferability, and accuracy are likely to become powerful tools for data extraction in the near future. Finally, databases for critical cooling rates of metallic glasses and yield strengths of high entropy alloys are developed using `ChatExtract`.

I. INTRODUCTION

Automated data extraction is increasingly used in to develop databases in materials science and other fields [1]. Many databases have been created using natural language processing (NLP) and language models (LMs) [2–24]. Recently, the emergence of large language models (LLMs) [25–29] has enabled significantly greater ability to extract complex data accurately [30, 31].

Previous automated methods require a significant amount of effort to set up, either preparing parsing rules, fine-tuning or re-training a model, or some combination of both, which specializes the method to perform a specific task. With the emergence of conversational LLMs such as `ChatGPT`, which are broadly capable and pre-trained for general tasks, there are opportunities for significantly improved information extraction methods that require almost no initial effort. These opportunities are enabled by harnessing the outstanding general language abilities of conversational LLMs, including their inherent capability to perform zero-shot (i.e., without additional training) classification, accurate word references identification, and information retention capabilities for text within a conversation. These capabilities, combined with *prompt engineering*, which is the process of designing questions and instructions (prompts) to improve the quality of results, can result in accurate data extraction without the need for fine-tuning of the model or significant knowledge about the property for which the data is to be extracted.

Prompt engineering has now become a standard practice in the field of image generation [32–34] to ensure high quality results. It has also been demonstrated that prompt engineering it an effective method in increasing the accuracy of reasoning in LLMs [35].

In this paper we demonstrate that using conversational LLMs such as `ChatGPT` in a zero-shot fashion with a well-engineered set of prompts can be a flexible, accurate and efficient method of extraction of materials properties in the form of the triplet *Material, Value, Unit*. We were able to minimize the main shortcoming of these conversational models, specifically errors in data extraction (e.g. improperly interpreted word relations) and hallucinations (i.e. responding with data not present in the provided text), and achieve 90.8% precision and 87.7% recall on a constrained test data set of bulk modulus, and 91.6% precision and 83.6% recall on a full practical database construction example of critical cooling rates for metallic glasses. These results were achieved by identifying relevant sentences, asking the model to extract details about the presented data, and then checking the extracted details by asking a series of follow-up questions that suggest uncertainty of the extracted information and introduce redundancy. This approach was first demonstrated in a preprint of this paper [36], and since then a group from Microsoft has described a similar idea, but for more general tasks than materials data extraction [37]. We work with short sentence clusters made up a a target sentence, the preceeding sentence, and the title, as we have found these almost always contain the full *Material, Value, Unit* triplet of data we seek. We also separated

* mppolak@wisc.edu
† ddmorgan@wisc.edu

cases with single and multiple data values in a sentence, thereby greatly reducing certain types of errors. In addition, by encouraging a certain structure of responses we simplified automated post-processing the text responses into a useful database. We have put these approaches together into a single method we call `ChatExtract` - a workflow for a fully automated zero-shot approach to data extraction. We provide an example `ChatExtract` implementation in a form of a python code (See. Data Availability for more details). The prompt engineering proposed here is expected to work for essentially all *Material, Value, Unit* data extraction tasks. For different types of data extraction this prompt engineering will likely need to be modified. However, we believe that the general method, which is based on simple prompts that utilized uncertainty-inducing redundant questioning applied within an information retaining conversational model, will provide an effective and efficient approach to many types of information extraction.

The `ChatExtract` method is largely independent of the conversational LLM used and is expected to improve as the LLMs improve. Therefore, the astonishing rate of LLM improvement is likely to further support the adoption of `ChatExtract` and similar approaches to data extraction. Prompt engineering has now become a standard practice in the field of image generation [32–34] to ensure high quality results. A parallel situation may soon occur for data extraction. Specifically, a workflow such as that presented here with `ChatExtract`, which includes prompt engineering utilized in a conversational set of prompts with follow-up questions, may become a method of choice to obtain high quality data extraction results from LLMs.

II. RESULTS AND DISCUSSION

Figure 1 shows a simplified illustration of the `ChatExtract` workflow. The full workflow with all of the steps is shown in Fig. 2 so here we only summarize the key ideas behind this workflow. The initial step is preparing the data and involves gathering papers, removing html/xml syntax and dividing into sentences. This task is straightforward, standard for any data extraction effort, and described in detail in other works[31].

The data extraction is done in two main stages:
(A) Initial classification with a simple relevancy prompt, which is applied to all sentences to weed out those that do not contain data.
(B) A series of prompts that control the data extraction from the sentences categorized in stage (A) as positive (i.e., as relevant to the materials data at hand). To achieve high performance in Stage (B) we have developed a series of engineered prompts and the key Features of the major Stage (B) are summarized here:

(1) Split data into single- and multi-valued, since texts containing a single entry are much more likely to be extracted properly and do not require follow up prompts, while extraction from texts containing multiple values is more prone to errors and requires further scrutinizing and verification.

(2) Include explicitly the possibility that a piece of the data may be missing from the text. This is done to discourage the model from hallucinating non-existent data to fulfill the task.

(3) Use uncertainty-inducing redundant prompts that encourage negative answers when appropriate. This lets the model reanalyze the text instead of reinforcing previous answers.

(4) Embed all the questions in a single conversation as well as representing the full data in each prompt. This simultaneously takes advantage of the conversational information retention of the chat tool while each time reinforcing the text to be analyzed.

(5) Enforce a strict Yes/No format of answers to reduce uncertainty and allow for easier automation.

Stage (A) is the first prompt given to the model. This first prompt is meant to provide information whether the sentence is relevant at all for further analysis, i.e. whether it contains the data for the property in question (value and units). This classification is crucial because, even in papers that have been extracted to be relevant by an initial keyword search, the ratio of relevant to irrelevant sentences is typically about 1:100. Therefore elimination of irrelevant sentences is a priority in the first step. Then, before starting Stage (B), we expand the text on which we are operating to a passage consisting of three sentences: the paper's *title*, the sentence *preceding* the positively classified sentence from the previous prompt, and the *positive sentence* itself. This expansion is primarily useful for making sure we include text with the material's name, which is sometimes not in the sentence targeted in Stage (A) but is almost always in the preceding sentence or title.

The relevant texts vary in their structure and we found it necessary to use different strategies for data extraction for those sentences that contain a single value and those sentences that contain multiple values (Feature (1) above). The texts containing only a single value are much simpler since the relation between material, value, and unit does not need to be analyzed. The LLMs tend to extract such data accurately and a single well-engineered prompt for each of the fields asked only once tends to perform very well. Texts containing multiple values involve a careful analysis of the relations between words to determine which values, materials, and units correspond to one another. This complexity sometimes leads to errors in extraction or hallucinated data and requires further scrutiny and prompting with follow-up questions. Thus the first prompt in Stage (2) aims at determining whether there are multiple data points included in a given sentence, and based on that answer one of two paths is taken, different for *single-valued* and *multi-valued* sentences. As a concrete example of how often this happens, our bulk modulus dataset studied below has 70% multi-valued and 30% single-valued sentences. Our follow-up question approach proved to be very successful for the

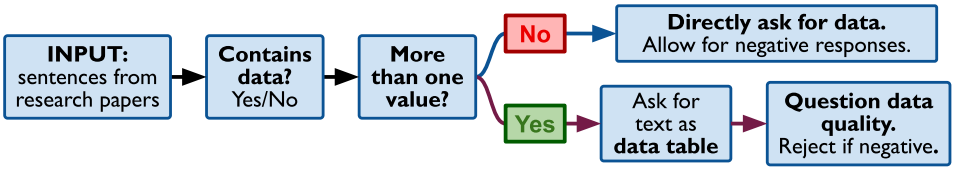

FIG. 1. A simplified flowchart describing our `ChatExtract` method of extracting structured data using a conversational large language model. Only the key ideas for each of the steps are shown, with the fully detailed workflow presented in Fig. 2

conversational `ChatGPT` models.

Next, the text is analyzed. For a single-valued text, we directly ask questions about the data in the text, asking separately for the value, its unit, and the material's name. It is important to explicitly allow for an option of a negative answer (Feature (2) above), reducing the chance that the model provides an answer even though not enough data is provided, limiting the possibility of hallucinating the data. If a negative answer is given to any of the prompt questions, the text is discarded and no data is extracted. For the case of a multi-valued sentence, instead of directly asking for data, we ask the model to provide structured data in a form of a table. This helps organize the for further processing but can produce factually incorrect data, even if explicitly allowing negative responses. Therefore, we scrutinize each field in the provided table by asking follow-up questions (this is the redundancy of Feature (3) above) whether the data and its referencing is really included in the provided text. Again, we explicitly allow for a negative answer and, importantly, plant a seed of doubt that it is possible that the extracted table may contain some inaccuracies. Similarly as before, if any of the prompt answers are negative, we discard the sentence. It is important to notice that despite the capability of the conversational model to retain information throughout the conversation, we repetitively provide the text with each prompt (Feature (4) above). This repetition helps in maintaining all of the details about the text that is being analyzed, as the model tends to pay less attention to finer details the longer the conversation is continued. The conversational aspect and information retention improves the quality of the answers and reinforces the format of short structured answers and possibility of negative responses. The importance of the information retention in a conversation is proven later in this work by repeating our analysis exercise but with a new conversation started for each prompt, in which cases both precision and recall are significantly lowered. It is also worth noticing that we enforce a strictly *Yes* or *No* format of answers for follow up questions (Feature (5) above), which enables automating of the data extraction process. Without this constraint the model tends to answer in full sentences which are challenging to automatically analyze.

The prompts described in the flowchart (Fig. 2) are engineered by optimizing the accuracy of the responses through trial and error on various properties of varying complexity. Obviously, we have not exhausted all options, and it is likely that further optimization is possible. We have, however, noticed that contrary to intuition, providing more information about the property in the prompt usually results in worse outcomes, and we believe that the prompts proposed here are a reliable and transferable set for many data extraction tasks.

We have investigated the performance of the `ChatExtract` approach on multiple property examples, including bulk modulus, metallic glass critical cooling rate, and high-entropy alloy yield stress. For bulk modulus the data is highly restricted so we can collect complete statistics on performance, and the other two cases represent applications of the method to full database generation. Our bulk modulus example data is taken from a large body of sentences extracted from hundreds of papers with bulk modulus data. To allow for an effective assessment we wanted a relatively small number of relevant sentences (containing data) of around a 100, from which we could manually extract to provide ground truth. We manually extracted data until we reached 100 relevant sentences, during which a corresponding number of 9164 irrelevant sentences (not containing data) was also labeled. We then post-processed the irrelevant sentences to remove ones that do not contain any numbers with a simple regular expression to obtain 1912 irrelevant sentences containing numbers. This preserves resources and saves time by not running the language model on sentences that obviously do not contain any data at all (since the values to be extracted are numbers), and does not impact the results of the assessment, as in our extensive tests the model never returns any datapoints from sentences that do not contain numbers. The model is is explicitly instructed not to do so in the prompts (see Fig. 2), even if it mistakenly classifies the sentence as relevant in the very first classification prompt (which is very rarely the case for sentences without numbers). In these 100 sentences with data, there were a total of 179 data points (a complete triplet of material, bulk modulus, and unit combination), which we extracted by hand to serve as a ground truth data set. We investigated the performance of multiple versions of `ChatGPT` models (see Tab. I) by following the approach as described above and in Fig. 2. For true positive sentences we divide the results into categories by type of text: single-valued and multi-valued, and provide the overall performance over the entire dataset. These results are summarized in Tab. I. Note that single- and multi-values columns represent performance on input passages that have data, which is of

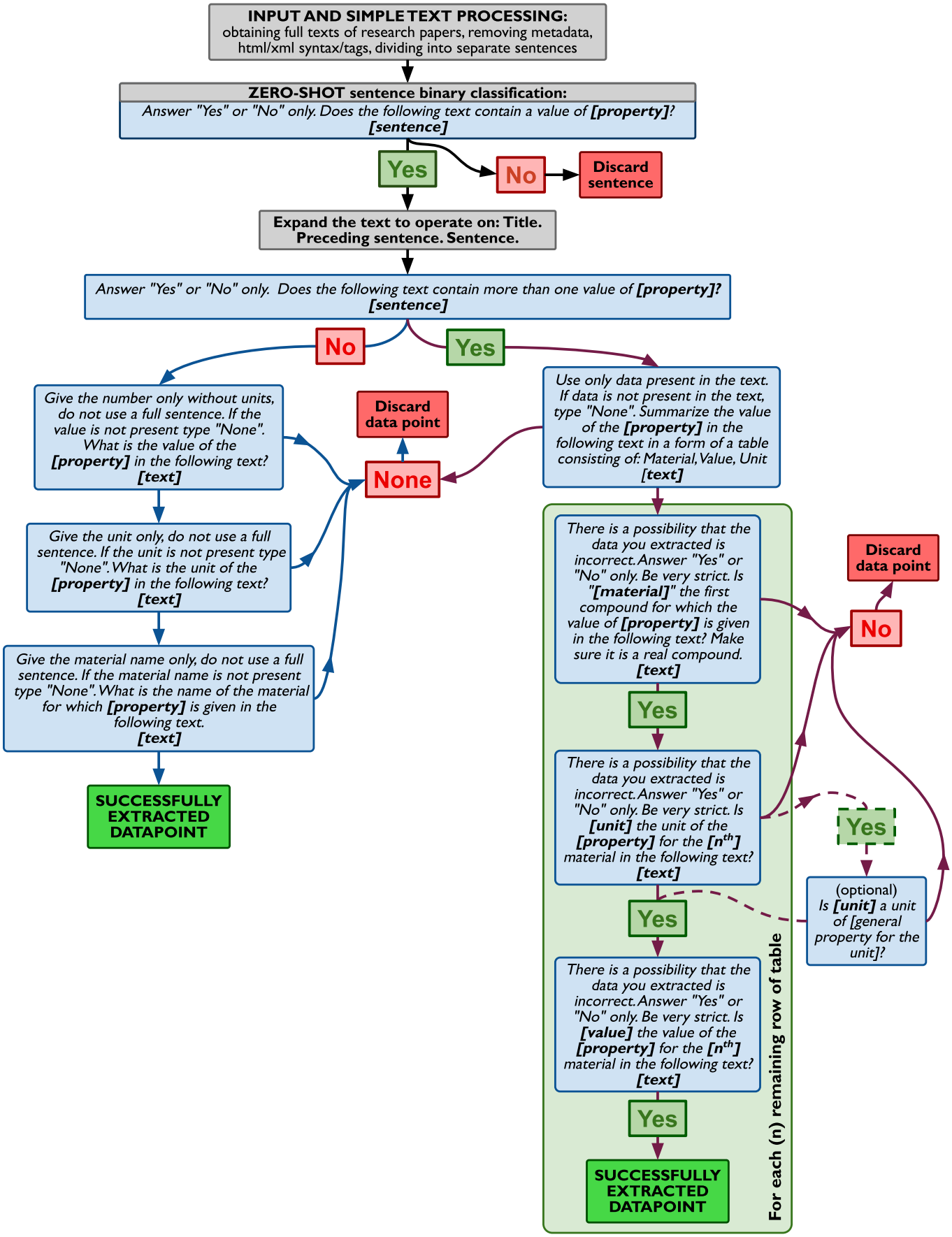

FIG. 2. A flowchart describing our `ChatExtract` method of extracting structured data using a conversational large language model. Blue boxes represent prompts given to the model, grey boxes are instructions to the user, "Yes", "No", and "None" boxes are model's responses. The text in "[]" are to be replaced with appropriate values of the named item, which includes one of sentence (the target sentence being analyzed), text (the expanded text consisting of Title, Preceding sentence, and target sentence.

interest for understanding model behavior. The statistic that best represents the model performance on real sentences is the overall column, where the input contains sentences both with and without data. We applied what we consider to be quite stringent criteria for assessing the performance against ground truth, the details of which an be found in the Methods section.

The best LLM (`ChatGPT-4`) achieved 90.8% precision at 87.7% recall, which is very impressive for a zero-shot approach that does not involve any fine-tuning. Single-valued sentences tend to be extracted with slightly higher recall (100% and 85.5% in `ChatGPT-4` and `ChatGPT-3.5`, respectively) compared to multi-valued sentences with a recall of 82.7% and 55.9% for the same models.

We believe that there are two core features of `ChatGPT` that are being used in `ChatExtract` to make this approach so successful. The first feature, and we believe the most important one, is the use of redundant prompts that introduce the possibility of uncertainty about the previously extracted data. By engineering the set of prompts and follow-up questions in this way, they substantially improve the factual correctness, and therefore precision, of the extracted information. The second feature is the conversational aspect, in which information about previous prompts and answers is retained. This allows the follow-up questions to relate to the entirety of the conversation, including the model's previous responses.

In order to demonstrate that the follow-up questions approach is essential to the good performance we repeated the exercise for both `ChatGPT` models without any follow-up questions (directly asking for structurized data only, in the same manner as before, only without asking the follow-up prompts in the multi-value branch (long light green box on right side of Fig. 2)). The results are denoted as *(no follow-up)* in Tab. I. The dominant effects of removing follow-up questions is to allow more extracted triplets to make it to the final extracted database. This generally increases recall across all cases (single-valued, multi-valued, and overall). For passages with data (single-valued, multi-valued) these additional kept triplets are very few and almost all correct, leading to just slightly lower precisions. However, for the large number of passages with no data the additional kept triplets represent many false positives, and therefore dramatically reduce precision in the overall category. Specifically, removing follow-up questions decreases the overall precision to just 42.7% and 26.5% for `ChatGPT-4` and `ChatGPT-3.5`, respectively, from the values of 90.8% and 70.1%, respectively, resulting from a full `ChatExtract` workflow. These large reductions in precision demonstrate that follow-up questions are critical, and the analysis shows that their role is primarily to avoid the model erroneously hallucinating data in passages where none was actually given.

In order to demonstrate that the information retention provided by the conversational model is important to the good performance we repeated our approach but modified it to start a new conversation for each prompt, which meant that no conversation history was available during each prompt response. The results are denoted *(no chat)* in Tab. I. This test was performed on `ChatGPT-3.5` and had little or no reduction in precision. However, there was a significant loss in recall in all categories (e.g., overall recall dropped by 10.7% to 54.7%). This loss is recall is because the multiple redundant questions tend to reject too many cases of correctly extracted triplets when the questions are asked without model knowing they are connected through a conversation. We did not perform this test on `ChatGPT-4` to reduce overall time and expense as the implications results on `ChatGPT-3.5` seemed clear.

To further demonstrate the utility of the `ChatExtract` approach we used the method to extract two materials property databases, one for critical cooling rates of bulk metallic glasses and one for yield strength of high entropy alloys. Before sharing the results of these data extractions it is useful to consider in more detail different types of desirable database of a materials property that might be extracted from text.

Different types of databases can be achieved with different levels of post-processing after automated data extraction. Here we describe three type of databases that we believe cover most database use cases. At one extreme is a database that encompasses all relevant mentions of a specific property, which is useful when initiating research in a particular field to assess the extent of data available in the literature, the span of values (including outliers), and the various material groups investigated. Entries included in such database might contain ranges, limits or approximate values, families of materials, etc. This is typically what the `ChatExtract` directly extracts, and we will refer to this as the *raw* database. At the other extreme is a strict *standardized* database, which

	Single-valued	Multi-valued	Overall
ChatGPT-4 (gpt-4-0314)	**P=100% R=100%**	**P=100% R=82.7%**	**P=90.8% R=87.7%**
ChatGPT-3.5 (gpt-3.5-turbo-0301)	**P=100% R=88.5%**	**P=97.3% R=55.9%**	**P=70.1% R=65.4%**
ChatGPT-4 (no follow-up) (gpt-4-0314)	P=100% R=100%	P=99.2% R=98.4%	P=42.7% R=98.9%
ChatGPT-3.5 (no follow-up) (gpt-3.5-turbo-0301)	P=97.9% R=88.5%	P=94.0% R=74.0%	P=26.5% R=78.2%
ChatGPT-3.5 (no chat) (gpt-3.5-turbo-0301)	P=100% R=76.9%	P=86.6% R=45.7%	P=70.0% R=54.7%

TABLE I. Precision (P) and recall (R) for different types of text passages: containing single- and multi-valued data, and overall, which includes all analyzed text passages, both containing data and not. Bold font represents final results of models used within the `ChatExtract` workflow, while the remaining demonstrate the importance of redundant follow-up questioning (no follow-up) and conversational information retention aspect (no chat).

contains uniquely defined materials with standard format, machine-readable compositions, and discrete values (i.e. not ranges or limits) with standardized units (which also helps remove the very rare occurrence where a triplet with wrong units is extracted). A *standardized* database facilitates easy interpretation and usage by computer codes and might be well-suited to, e.g., machine learning analysis. A *standardized* database can be developed from a *raw* data collection and will be a subset of that *raw* data. A third type of dataset, which is intermediate between *raw* and *standardized*, is a *cleaned* database, which removes duplicate triplets derived from within a single paper from the *raw* data, as these are almost always the exact same entry repeated multiple, e.g. in the Discussion and Conclusions sections) and are obviously undesirable. While the *cleaned* database can be done automatically, the *standardized* database may require some manual post-processing of the data extracted with the `ChatExtract` method.

In this study we provide two materials property databases: critical cooling rates of metallic glasses, and yield strength of high entropy alloys. Both databases are presented in all three of the above forms: *raw* data which is what is directly extracted by the `ChatExtract` approach, *cleaned* data from which single paper duplicate values have been removed, and *standardized* data where all materials that were uniquely identifiable are in a standard form of AXBYCZ... (where A,B,C,... are elements and X,Y,Z,... are mole fractions). The standardization required post-processing which we accomplished manually and with a combination of further prompting with an LLM, text processing with regular expression and `pymatgen` [38]. While this standardization approach may introduce some additional errors, it provides a very useful form for the data with modest amounts of human effort. While we were able to employ further prompting combined with LLMs and text analysis tools to make this conversion, the approaches necessitate substantial additional prompt engineering and coding, which was time consuming and likely not widely applicable without significant changes. Given these limitations of our present approach to generating a *standardized* database from a *raw* database we do not discuss the details of our approach or attempt to provide a guide on how to do this most effectively. Automating developing a *standardized* database from a *raw* database is an important topic for future work.

To limit the scope of critical cooling rates to just metallic glasses, and yield strengths to high entropy alloys, we first limited the source papers by providing a specified query to the publishers database to return only papers relevant to the family of systems we were after, and then we applied a simple composition-based filter to the final *standardized* database. Details about both these steps are given in the following section when discussing the respective databases.

The first database is of critical cooling rates in the context of metallic glasses. To obtain source research articles we performed a search query "bulk metallic glass"+"critical cooling rate" from Elsevier's ScienceDirect database which returned 684 papers, consisting of 110126 sentences.

A reference database (ground truth), which we will call R_{c1} , was developed using a thorough manual data extraction process based on text processing and regular expressions and aided by a previous database of critical cooling rates extracted with a more time consuming and less automated approach that involved significant human involvement [31]. This laborious process done by an experienced researcher although highly impractical, labor-intensive and time consuming, is capable of providing the most complete and accurate reference database, allowing to accurately evaluate the performance of `ChatExtract` in a real database extraction scenario, which is the most relevant assessment of the method.

To develop the critical cooling rate database with `ChatExtract` the `ChatExtract` approach was applied identically as to the bulk modulus case except that the phrase "bulk modulus" was replaced with "critical cooling rate". We call this dataset R_{c2} . In comparing R_{c2} data to R_{c1} ground truth the same rules for equivalency of triplet datapoints have been applied in the same way as the benchmark bulk modulus data (see Methods): equivalent triplets had to have identical units and values (including inequality symbols, if present), and material names had to be similar enough to allow entries to be uniquely identified as the same materials system (e.g. "$Mg_{100-x}Cu_xGd_{10}$ (x=15)" was the same as $Mg_{85}Cu_{15}Gd_{10}$, but not the same as "Mg-Cu-Gd metallic glass" and "Zr-based bulk metallic glass" was the same as "Zr-based glass forming alloy" but not the same as $Zr_{41.2}Ti_{13.8}Cu_{12.5}Ni_{10.0}Be_{22.5}$). Critical cooling rates for bulk metallic glasses proved to be quite a challenging property to extract. The analyzed papers very often (much more often than in the other properties we worked on) contained values of critical cooling rates described as ranges or limits, and the materials were often families or broad group of materials, in particular in the Introduction sections of the papers. The `ChatExtract` workflow is aimed at extracting triplets of materials, value, and units without specifying further what do these mean exactly, as will be discussed in the next paragraph. To provide the most comprehensive assessment, the human curated database contains all mentions of critical cooling rates that are accompanied by any number, no matter how vague or specific. This manually extracted, very challenging *raw* database contained 721 entries. `ChatExtract` applied on the same set of research papers resulted in 634 extracted values with 76.9% precision and 63.1% recall. The vast majority of reduction in precision and recall comes from the more ambiguous material names such as the above mentioned broad groups or families of materials or ranges and limits of values. In many cases the error in extraction was minor, such as a missing inequality sign (e.g. "<0.1" in R_{c1} but "0.1" in R_{c2}), extracting only one value from a range (e.g. "10-100"

in R_{c1} but only "10" in R_{c2}), or missing details in materials described as a group or family (e.g. "Zr-based metallic glasses" in R_{c1} but only "Metallic glasses" R_{c2}). Even though these could be regarded as minor errors, we still consider such triplets to be incorrect. The performance is slightly improved for the *cleaned* database where a precision of 78.1% and 64.3% recall is obtained with 637 and 553 entries in R_{c1} and R_{c2} , respectively. The most relevant *standardized* version of the database, when extracted with `ChatExtract` yielded a final precision of 91.9% and 84.2% recall, with 313 and 286 entries subject for comparison in R_{c1} and R_{c2} , respectively. This large reduction in the size of the *standardized* database when compared to *cleaned*, and the improvement in performance, are both due to the large amount of material groups/families and ranges/limits of values. These cases do not classify as uniquely identifiable material compositions and discrete values so they do not satisfy the requirements for the *standardized* database, and as mentioned before, they were the most problematic for `ChatExtract` to extract (as they were for the human curating R_{c1}). It is important to note that in order to provide an accurate assessment of the extraction performance, as mentioned previously, the triplets are not matched by themselves, but they also have to originate from the same text passage. Therefore both the ground truth and the `ChatExtract` extracted databases were standardized separately, and if either contained a standardized value, it was considered in the assessment, making the comparison more challenging. The performance of `ChatExtract` for the *standardized* database of critical cooling rates is close to that for bulk modulus presented in Tab. I and demonstrates the transferabilty of `ChatExtract` to different properties.

The final *standardized* database obtained with `ChatExtract` consists of 280 datapoints. Duplicate values originating from within a single paper have already been removed for the *cleaned* database, duplicate triplets originating from different papers are still present. We believe it is important to keep all values, as it allows for an accurate representation of the frequency at which different systems are studied and for accurate averaging if necessary. If the duplicates were to be removed, 192 unique triplets would be left, with the many duplicates being for an industry standard system $Zr_{41.2}Ti_{13.9}Cu_{12.5}Ni_{10}Be_{22.5}$ (Vit1). The values in the final database ranged from 10^{-3} K/s (for $Ni_{40}P_{20}Pd_{40}$) to 10^{12} (for $Mg_{25}Al_{75}$), with an average 10^2, all quite reasonable values. An additional *standardized-MG* database is given, in which all non-metallic materials have been removed. In the case of this modest-sized database, simply removing oxygen containing systems proved to be enough and 5 non-metallic oxide materials have been removed. Out of the 192 unique datapoints, there were 125 unique material compositions (some compositions had multiple values originating from different research papers) in the *standardized* database, and after removing non-metallic systems *standardized-MG* database contained 182 unique datapoints for 120 unique material compositions. This size of 120 unique compositions is significantly larger than the previous largest hand-curated database published by Afflerbach, et al. [39], which had just 77 entries. This result shows that, at least in this case, `ChatExtract` can generate more quality data with much less time than human extraction efforts.

Finally, we developed a database of yield strength of high entropy alloys (HEAs) using the `ChatExtract` approach. This database does not have any readily available ground truth for validation but represents a very different property and alloy set than either bulk modulus or critical cooling rate and therefore further demonstrates the efficacy of the `ChatExtract` approach. In the first step we searched for a combination of the phrase "yield strength" and ("high entropy alloys" or "compositionally complex alloys" or "multi-principle component alloys") in the Elsevier's ScienceDirect API. The search returned 4029 research papers consisting of 840431 sentences. 10269 *raw* data points were extracted. The *cleaned* database consisted of 8980 datapoints. Further post-processing yielded 4859 datapoints that constitute the *standardized* data, where we assumed that all compositions were given as atomic %, unless otherwise stated in the analyzed text (which was infrequent). The 4854 *standardized* datapoints contained a number of alloys that were not were not HEAs, with HEA defined as a systems containing 5 or more elements. The non-HEA systems are not an error in `ChatExtract` as the data was generally in the papers, despite their being extracted by the above initial keyword search. By restricting the database to only HEAs we obtained a final *standardized-HEA* database of 2804 values. The *standardized-HEA* database had 636 materials with unique compositions. The values ranged from 12 MPa for $Al_{0.4}Co_1Cu_{0.6}Ni_1Si_{0.2}$ to 19.16 GPa for $Fe_7Cr_{31}Ni_{23}Co_{34}Mn_5$. These values are extreme but not unphysical and we have confirmed that both these extremes are extracted correctly. The distribution of yield stress values resembles a positively skewed normal distribution with a maximum around 400 MPa, which is a physically reasonable distribution shape with a peak at a typical yield stress for strong metal alloys. A large automatically extracted database of general yield strengths, not specific to HEAs, has been developed previously [5]. Direct quantitative comparison is not straightforward, but the histogram of values obtained from the previous database exhibits a very similar shape to the data obtained here, further supporting that our data is reasonable. The database of yield strengths for HEAs developed here is significantly larger than databases developed for HEAs previously, for example, databases containing yield strengths for 169 unique HEA compositions from 2018 [40] and containing yield strength for 419 unique HEA compositions from 2020 [41]. The `ChatExtract` generated databases are available in Figshare [42] (See. Data Availability).

Now that we developed and analyzed these databases it is easier to understand the utility of `ChatExtract` .

ChatExtract was developed to be general and transferable, therefore it tackles a fundamental type of data extraction - a triplet of *Material, Value, Unit* for a specified property, without imposing any other restrictions. The lack of specificity when extracting "Material" or "Value" allows for extraction of data from texts where the materials are presented both as exact chemical compositions, or broad groups or families of systems. Similarly values may be discrete numbers, or ranges or limits. However, certain restrictions are often desired in developing a database, and we believe that these fall into two broad categories with respect to the challenges of integrating them into the present ChatExtract workflow. The first category is restrictions based on the extracted data, for example, targeting only desired compositions or ranges of a property value. Such restrictions are trivial to integrate with ChatExtract by either limiting the initial search query in the publisher's database, limiting the final *standardized* database, or both, based on the restriction. For example, in our HEA database we assured only HEAs in final data by both limiting the search query in the publisher's database and applying a composition-based rule on the final *standardized* database. The second category is where we want a property value when some other property conditions hold, for example, the initial property should be considered at a certain temperature and pressure. This situation is formally straightforward for ChatExtract as it can be captured by generalizing the problem from finding the triplet: *material, property, unit*, to finding the multiplet: *material,* $property_1$*,* $unit_1$*,* $property_2$*,* $unit_2$*, ...* The ChatExtract workflow can then be generalized to apply to these multiplets by adding more steps to both the left and right branches in Fig. 2, for example if a temperature at which the data was obtained was relevant, the left branch would contain two more boxes, the first being: *Give the number only without units, do not use a full sentence. If the value is not present type "None". What is the value of the temperature at which the value of [property] is given in the following text?*, followed by a second similar one prompting for the unit. The first prompt in the right branch would ask for a table that also included a temperature value and temperature unit, followed by two validation prompts for those two columns. This approach could be expanded into extracting non-numerical data as well, such as sample crystalinity or processing conditions. While these generalizations are formally straightforward we have made no assessment of their accuracy in this work, and some changes to ChatExtract might be needed to implement them effectively. For example, any additional constraints or information would have to be included in the text being examined by the LLM, and the more information that is required, the less likely it is that it will all be contained in the examined text passage. Thus the examined text passage may need to be expanded, or sometimes the required additional data may be missing from the paper altogether.

III. CONCLUSIONS

This paper demonstrates that conversational LLMs such as ChatGPT, with proper prompt engineering and a series of follow-up questions, such as the ChatExtract approach presented here, are capable of providing high quality materials data extracted from research texts with no additional fine-tuning, extensive code development or deep knowledge about the property for which the data is extracted. We present such a series of well-engineered prompts and follow-up questions in this paper and demonstrate its effectiveness resulting in a best performance of over 90% precision at 87.7% recall on our test set of bulk modulus data, and 91.6% precision and 83.6% on a full database of critical cooling rates. We show that the success of the ChatExtract method lies in asking follow-up questions with purposeful redundancy and introduction of uncertainty and information retention within the conversation by comparing to results when these aspects are removed. We further develop two databases using ChatExtract - a database of critical cooling rates for metallic glasses and yield strengths for high entropy alloys. The first one was modest-sized and served as a benchmark for full database development since we were able to compare it to data we extracted manually. The second one was a large database, to our knowledge the largest database of yield strength of high entropy alloys to date. The high quality of the extracted data and the simplicity of the approach suggests that approaches similar to ChatExtract offer an opportunity to replace previous, more labor intensive, methods. Since ChatExtract is largely independent of the used model, it is also likely improve by simply applying it to newer and more capable LLMs as they are developed in the future.

IV. METHODS

The main statistical quantities used to assess performance of ChatExtract were precision and recall, defined as:
$Precision = \frac{TruePositive}{TruePositive+FalsePositive}$
$Recall = \frac{TruePositive}{TruePositive+FalseNegative}$.
In our assessment we defined true positives (for precision) and false negatives (for recall) in terms of each input text passage, which we define above to consist of a target sentence, its preceding sentence, and the title. The exact approach can be confusing so we describe it concretely for every case. For a given input text passage there are zero, one or multiple unique datapoint triplets of *material, value, and unit*. We take the hand extracted triplets as the ground truth. We then process the text passage with ChatExtract to get a set of zero or more extracted triplets. If the ground truth has zero triplets and the extracted data has zero triplets, this is a true negative. Every extracted triplet from a passage with zero ground truth triplets is counted as a false positive. If

the ground truth has one triplet and the extracted data has zero triplets this is counted as a false negative. If the ground truth has one triplet and the extracted data has one equivalent triplet (we will define "equivalent" below) then this is counted as a single true positive. If the ground truth has one triplet and the extracted data has one inequivalent triplet then this is counted as a single false positive. If the ground truth has one triplet and the extracted data has multiple triplets they are each compared against the ground truth sequentially, assigning them as a single true positive if they are equivalent to the ground truth triplet and a single false positive if they are not equivalent to the ground truth triplet. However, only one match (a match is an equivalent pair of triplets) can be made of an extracted triplet to each ground truth triplet for a given sentence, i.e., we consider the ground truth triplet to be used up after one match. Therefore, any further extracted triplets that are equivalent to the ground truth triplet are still counted as each contributing a single false positive. Finally, we consider the case where ground truth has multiple triplets. In this case, if the extracted data has no triplets it is counted as a multiple false negatives. If the extracted data has one triplet and it is equivalent to any ground truth triplet that is counted as one true positive. If the extracted data has multiple triplets each one is compared to each ground truth triplet. If a given extracted triplet is equivalent to any one of the ground truth triplets that extracted triplet is counted as a true positive. However, as above, each ground truth triplet can only be matched once, and any addition matches of extracted triplets to an already matched ground truth triplet are counted as one additional false positive.

In the above we defined "equivalent" triplets in the following way. First, equivalent triplets had to have identical units and values (if uncertainty was present, it did not have to be extracted, but if it was extracted it had to be extracted properly as well). Second, equivalent triplets had to have materials names in the ground truth and extracted text that uniquely identified the same materials system (e.g., $Li_{17}Si_{(4-x)}Ge_x$ (x=2.3) and $Li_{17}Si_{1.7}Ge_{2.3}$ would be equivalent but "Zr-Ni alloy" and "$Zr_{62}Ni_{38}$" would not). These requirements for equivalent triplets are quite unforgiving. In particular, in many cases where we identified false positives the LLM extracted data that was partially right or had just small errors. This fact suggests that better precision and recall might be obtained with some human input or further processing. Overall we believe the above methods provide a rigorous and demanding assessment of the `ChatExtract` approach. For `ChatGPT API (gpt-3.5-turbo-0301)` and `(gpt-4-0314)`, default parameters were used, except for temperature frequency and presence penalties, which have been set to zero. No system prompt was used.

Acknowledgments: The research in this work was primarily supported by the National Science Foundation Cyberinfrastructure for Sustained Scientific Innovation (CSSI) Award No. 1931298.

Data Availability: The extracted databases of critical cooling rates and yield strength for HEAs, data used in the assessment of the models is available on figshare [42]: https://doi.org/10.6084/m9.figshare.22213747. There, we also provide at this link a version of the python code we used for data extraction that follows the workflow presented in Fig. 2, and involves additional simple post-processing of the `ChatGPT` responses to follow the workflow and provide a more convenient output. The post-processing included in the example code is relatively simple, and while it worked well for the properties we studied here, there may be cases when it fails in processing responses for different datasets, since occasionaly `ChatGPT` may format its response in an unexpected way. The provided code is just a simple example of how `ChatExtract` could be implemented and has a very limited error handling.

Contributions: M. P. P. conceived the study, performed the modeling, tests and prepared/analyzed the results, D. M. guided and supervised the research. Writing of the manuscript was done by M. P. P. and D. M..

[1] E. A. Olivetti, J. M. Cole, E. Kim, O. Kononova, G. Ceder, T. Y.-J. Han, and A. M. Hiszpanski, Data-driven materials research enabled by natural language processing and information extraction, Applied Physics Reviews **7**, 041317 (2020).

[2] M. C. Swain and J. M. Cole, Chemdataextractor: A toolkit for automated extraction of chemical information from the scientific literature, Journal of Chemical Information and Modeling **56**, 1894 (2016).

[3] J. Mavračić, C. J. Court, T. Isazawa, S. R. Elliott, and J. M. Cole, Chemdataextractor 2.0: Autopopulated ontologies for materials science, Journal of Chemical Information and Modeling **61**, 4280 (2021).

[4] C. Court and J. Cole, Magnetic and superconducting phase diagrams and transition temperatures predicted using text mining and machine learning, npj Comput Mater **6**, 18 (2020).

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

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

[7] J. Zhao and J. M. Cole, Reconstructing chromatic-dispersion relations and predicting refractive indices using text mining and machine learning, Journal of Chemical Information and Modeling **62**, 2670 (2022).

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

[9] E. Beard and J. Cole, Perovskite- and dye-sensitized solar-cell device databases auto-generated using chemdataextractor, Sci Data **9**, 329 (2022).

[10] Q. Dong and J. Cole, Auto-generated database of semiconductor band gaps using chemdataextractor, Sci Data **9**, 193 (2022).

[11] E. J. Beard, G. Sivaraman, A. Vazquez-Mayagoitia, *et al.*, Comparative dataset of experimental and computational attributes of uv/vis absorption spectra, Sci Data **6**, 307 (2019).

[12] Z. Wang, O. Kononova, K. Cruse, *et al.*, Dataset of solution-based inorganic materials synthesis procedures extracted from the scientific literature, Sci Data **9**, 231 (2022).

[13] H. Huo, C. J. Bartel, T. He, A. Trewartha, A. Dunn, B. Ouyang, A. Jain, and G. Ceder, Machine-learning rationalization and prediction of solid-state synthesis conditions, Chemistry of Materials **34**, 7323 (2022).

[14] J. E. Saal, A. O. Oliynyk, and B. Meredig, Machine learning in materials discovery: Confirmed predictions and their underlying approaches, Annual Review of Materials Research **50**, 49 (2020).

[15] D. Morgan and R. Jacobs, Opportunities and challenges for machine learning in materials science, Annual Review of Materials Research **50**, 71 (2020).

[16] J. Zhao and J. M. Cole, Reconstructing chromatic-dispersion relations and predicting refractive indices using text mining and machine learning, Journal of Chemical Information and Modeling **62**, 2670 (2022).

[17] Z. Wang, O. Kononova, K. Cruse, T. He, H. Huo, Y. Fei, Y. Zeng, Y. Sun, Z. Cai, W. Sun, and G. Ceder, Dataset of solution-based inorganic materials synthesis procedures extracted from the scientific literature, Scientific Data **9**, 231 (2022).

[18] C. Karpovich, Z. Jensen, V. Venugopal, and E. Olivetti, Inorganic synthesis reaction condition prediction with generative machine learning 10.48550/ARXIV.2112.09612 (2021).

[19] A. B. Georgescu, P. Ren, A. R. Toland, S. Zhang, K. D. Miller, D. W. Apley, E. A. Olivetti, N. Wagner, and J. M. Rondinelli, Database, features, and machine learning model to identify thermally driven metal–insulator transition compounds, Chemistry of Materials **33**, 5591 (2021).

[20] O. Kononova, T. He, H. Huo, A. Trewartha, E. A. Olivetti, and G. Ceder, Opportunities and challenges of text mining in materials research, iScience **24**, 102155 (2021).

[21] E. Kim, Z. Jensen, A. van Grootel, K. Huang, M. Staib, S. Mysore, H.-S. Chang, E. Strubell, A. McCallum, S. Jegelka, and E. Olivetti, Inorganic materials synthesis planning with literature-trained neural networks, Journal of Chemical Information and Modeling **60**, 1194 (2020).

[22] E. Kim, K. Huang, A. Saunders, A. McCallum, G. Ceder, and E. Olivetti, Materials synthesis insights from scientific literature via text extraction and machine learning, Chemistry of Materials **29**, 9436 (2017).

[23] Z. Jensen, E. Kim, S. Kwon, T. Z. H. Gani, Y. Román-Leshkov, M. Moliner, A. Corma, and E. Olivetti, A machine learning approach to zeolite synthesis enabled by automatic literature data extraction, ACS Central Science **5**, 892 (2019).

[24] L. P. J. Gilligan, M. Cobelli, V. Taufour, and S. Sanvito, A rule-free workflow for the automated generation of databases from scientific literature (2023).

[25] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, Language models are few-shot learners 10.48550/ARXIV.2005.14165 (2020).

[26] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe, Training language models to follow instructions with human feedback 10.48550/ARXIV.2203.02155 (2022).

[27] B. Workshop, :, T. L. Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilić, D. Hesslow, R. Castagné, A. S. Luccioni, F. Yvon, and M. Gallé et al., Bloom: A 176b-parameter open-access multilingual language model 10.48550/ARXIV.2211.05100 (2022).

[28] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, T. Mihaylov, M. Ott, S. Shleifer, K. Shuster, D. Simig, P. S. Koura, A. Sridhar, T. Wang, and L. Zettlemoyer, Opt: Open pre-trained transformer language models 10.48550/ARXIV.2205.01068 (2022).

[29] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, Llama: Open and efficient foundation language models 10.48550/arXiv.2302.13971 (2023).

[30] A. Dunn, J. Dagdelen, N. Walker, S. Lee, A. S. Rosen, G. Ceder, K. Persson, and A. Jain, Structured information extraction from complex scientific text with fine-tuned large language models 10.48550/ARXIV.2212.05238 (2022).

[31] M. P. Polak, S. Modi, A. Latosinska, J. Zhang, C.-W. Wang, S. Wang, A. D. Hazra, and D. Morgan, Flexible, model-agnostic method for materials data extraction from text using general purpose language models https://doi.org/10.48550/arXiv.2302.04914 (2023).

[32] Midjourney, `https://www.midjourney.com`, [Online; accessed 08-Feb-2023].

[33] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, Hierarchical text-conditional image generation with clip latents 10.48550/ARXIV.2204.06125 (2022).

[34] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, High-resolution image synthesis with latent diffusion models, in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022) pp. 10674–10685.

[35] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, Large language models are zero-shot reasoners https://doi.org/10.48550/arXiv.2205.11916 (2022).

[36] M. P. Polak and D. Morgan, Extracting accurate materials data from research papers with conversational language models and prompt engineering, (2023), arXiv:2303.05352.

[37] B. Li, R. Wang, J. Guo, K. Song, X. Tan, H. Hassan, A. Menezes, T. Xiao, J. Bian, and J. Zhu, Deliberate then generate: Enhanced prompting framework for text generation, (2023), arXiv:2305.19835.

[38] S. P. Ong, W. D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V. L. Chevrier, K. A. Persson, and G. Ceder, Python materials genomics (pymatgen): A robust, open-source python library for materials analysis, Computational Materials Science **68**, 314 (2013).

[39] B. T. Afflerbach, C. Francis, L. E. Schultz, J. Spethson, V. Meschke, E. Strand, L. Ward, J. H. Perepezko, D. Thoma, P. M. Voyles, I. Szlufarska, and D. Morgan, Machine learning prediction of the critical cooling rate for metallic glasses from expanded datasets and elemental features, Chemistry of Materials **34**, 2945 (2022).

[40] S. Gorsse, M. Nguyen, O. Senkov, and D. Miracle, Database on the mechanical properties of high entropy alloys and complex concentrated alloys, Data in Brief **21**, 2664 (2018).

[41] C. K. H. Borg, C. Frey, J. Moh, T. M. Pollock, S. Gorsse, D. B. Miracle, O. N. Senkov, B. Meredig, and J. E. Saal, Expanded dataset of mechanical properties and observed phases of multi-principal element alloys, Scientific Data **7**, 430 (2020).

[42] M. P. Polak and D. Morgan, Datasets and Supporting Information to the paper entitled 'Using conversational AI to automatically extract data from research papers - example of ChatGPT' 10.6084/m9.figshare.22213747 (2023).