

Cite this: *Digital Discovery*, 2024, 3, 313

MaScQA: investigating materials science knowledge of large language models†

Mohd Zaki,^{ID} ^a Jayadeva,^{ID} ^{bc} Mausam^{ID} ^{cd} and N. M. Anoop Krishnan^{ID} ^{*ac}

Information extraction and textual comprehension from materials literature are vital for developing an exhaustive knowledge base that enables accelerated materials discovery. Language models have demonstrated their capability to answer domain-specific questions and retrieve information from knowledge bases. However, there are no benchmark datasets in the materials science domain that can be used to evaluate the understanding of the key concepts by these language models. In this work, we curate a dataset of 650 challenging questions from the materials domain that require the knowledge and skills of a materials science student who has cleared their undergraduate degree. We classify these questions based on their structure and the materials science domain-based subcategories. Further, we evaluate the performance of LLaMA-2-70B, GPT-3.5, and GPT-4 models on solving these questions via zero-shot and chain of thought prompting. It is observed that GPT-4 gives the best performance (~62% accuracy) as compared to other models. Interestingly, in contrast to the general observation, no significant improvement in accuracy is observed with the chain of thought prompting. To evaluate the limitations, we performed an error analysis, which revealed conceptual errors (~72%) as the major contributor compared to computational errors (~28%) towards the reduced performance of the LLMs. We also compared GPT-4 with human performance and observed that GPT-4 is better than an average student and comes close to passing the exam. We also show applications of the best performing model (GPT-4) on composition–extraction from tables of materials science research papers and code writing tasks. While GPT-4 performs poorly on composition extraction, it outperforms all other models on the code writing task. We hope that the dataset, analysis, and applications discussed in this work will promote further research in developing better materials science domain-specific LLMs and strategies for information extraction.

Received 25th September 2023

Accepted 19th December 2023

DOI: 10.1039/d3dd00188a

rsc.li/digitaldiscovery

Introduction

Large language models (LLMs) are machine learning (ML) models based on transformer neural network architecture.¹ These models are called *large* due to their billions of inherent parameters. The increase in the number of model parameters and different training strategies have improved the performance of these models on natural language tasks such as question answering,^{2,3} text summarization,^{4,5} sentiment analysis,^{1,3} machine translation,⁶ conversational abilities,^{7–9} and code generation.¹⁰ In the materials science domain, existing

datasets are mainly related to tasks like named entity recognition (NER),^{11,12} text classification,^{13–15} synthesis process and relation classification,¹⁶ and composition extraction from tables,¹⁷ which are used by researchers to benchmark the performance of materials domain language models like MatSciBERT¹⁴ (the first materials-domain language model), MatBERT,¹⁸ MaterialsBERT,¹⁹ OpticalBERT,²⁰ and BatteryBERT.¹⁵ Recently, Song *et al.* (2023) reported better performance of materials science domain specific language models compared to BERT and SciBERT on seven materials domain datasets related to named entity recognition, relation classification, and text classification.²¹

There exist several large-sized datasets like MMLU,^{22,23} Hel-laSwag,²⁴ WinoGrande,²⁵ HumanEval,¹⁰ and DROP²⁶ to evaluate the capabilities of LLMs. However, there are limited datasets in the materials science domain for assessing their question-answering abilities. Table 1 lists datasets related to mathematics, chemistry, and materials science suitable for evaluating LLMs. In addition to these datasets, Jablonka *et al.* (2023) demonstrated the application of LLMs on 14 chemistry and materials science specific datasets.²⁷ Further, based on datasets

^aDepartment of Civil Engineering, Indian Institute of Technology Delhi, Hauz Khas, New Delhi 110016, India. E-mail: krishnan@iitd.ac.in; cez198233@iitd.ac.in

^bDepartment of Electrical Engineering, Indian Institute of Technology Delhi, Hauz Khas, New Delhi 110016, India

^cYardi School of Artificial Intelligence, Indian Institute of Technology Delhi, Hauz Khas, New Delhi 110016, India

^dDepartment of Computer Science & Engineering, Indian Institute of Technology Delhi, Hauz Khas, New Delhi 110016, India

† Electronic supplementary information (ESI) available. See DOI: <https://doi.org/10.1039/d3dd00188a>



Table 1 List of question answering datasets in mathematics, basic sciences, and materials science domain

Dataset	Description
GSM8K ³⁰	A set of 8.5 K linguistically diverse grade school mathematics word problems
AI2RC (ARC) ³¹	A set of ~7.7 K school-level science questions
ChemistryQA ³²	A dataset of 4.5 K chemistry question answers
ScienceQA ³³	~21 K multimodal multiple choice questions from natural, language, and social sciences
SciQ ³⁴	Crowdsourced dataset of ~13.7 K science questions
MatSci-Instruct ²⁹	52 K synthetically generated instructions dataset to finetune LLMs for materials science information extraction
JEEBench ³⁵	450 questions on physics, chemistry, and mathematics from JEE advanced examination of India for admission to IITs

listed in Table 1, researchers have attempted to finetune materials domain LLMs and proposed DARWIN²⁸ and HoneyBee²⁹ and compared them with the performance of LLaMA, GPT3.5 and GPT-4 on different tasks.

Although these datasets are diverse, the complexity of questions asked in examinations for testing students who have completed their undergraduate-level education, is quite different from the existing ones. Therefore, developing such a dataset is crucial to investigate the materials science domain knowledge of these LLMs so that they can be further used for addressing challenging problems related to materials discovery for areas such as manufacturing, energy, environment, and sustainability. This information is further essential to understand the lacunae of the understanding of such LLMs, which are being proposed to be used for several domains such as manufacturing, planning, material synthesis, and materials discovery.^{14,19}

To this end, we collected questions that require students to have a undergraduate-level understanding of materials science topics to solve them. These questions and answers are carefully curated from the original questions in the graduate aptitude test in engineering (GATE) exam—a national-level examination for graduate admission in India. More than 800 000 students take this exam annually, with an average of 100 000 students in major disciplines, such as mechanical or civil engineering, to enroll in master's/doctoral courses in the premier institutes in India. We classify these questions based on their (a) structure, which leads to 4 types of questions, and (b) domain knowledge required to solve them, which divides the database into 14 categories. The questions in MaScQA also have diversity in length, ranging from 9 words per question to 145 words question, with an average of 50 words (see Fig. S1†). We then evaluate the performance of state-of-the-art proprietary models, GPT-3.5 and GPT4, and an open-source model, LLaMA-2-70B, in solving these questions. The availability of MaScQA will allow the researchers to benchmark existing models and prompting strategies. Specifically, the analysis from a domain-specific perspective will enable the researchers to train better domain-specific LLMs and help them decide where these models can be used in the materials discovery pipeline. Note that, MaScQA is an open database where other researchers can also contribute questions for increasing the diversity of topics on which LLMs can be evaluated.

Finally, we evaluate LLMs on domain-specific tasks and compare their performance with the existing models suitable

for such tasks. The first task, introduced by Gupta *et al.* (2023),¹⁷ is composition extraction from tables in materials-related research papers. The second task, related to code writing, employed the dataset released by White *et al.* (2022),³⁶ which is a compilation of ~100 Python functions comprising of the docstring, return statement, and the [insert] token, which has to be replaced upon prompting the LLM. The performance of GPT-4 on these tasks further allows researchers to devise strategies for task-oriented finetuning of the LLMs. Overall, we try to answer the following questions in this paper:

1. How well do general-purpose LLMs perform in answering complex questions from the materials science domain?
2. Can we improve the performance of the LLMs by using the chain of thought prompting methods?
3. What are the factors limiting the performance of these LLMs?
4. Can LLMs be used for accelerated materials modelling and design through information extraction and code writing?

Methodology

Dataset preparation

We are motivated to investigate how LLMs will perform on questions that require an undergraduate-level understanding of materials science topics for their solution. To compile a dataset of such questions, we take question papers related to materials science and metallurgical engineering asked in the GATE examination conducted in India for admission to masters and doctorate courses. To this end, we compiled 650 questions and classified them into four types based on their structure: multiple choice questions (MCQs), match the following type questions (MATCH), numerical questions where options are given (MCQN), and numerical questions (NUM). MCQs are generally conceptual, given four options, out of which mostly one is correct and sometimes more than one option is also correct (Fig. 1(a)). In MATCH, two lists of entities are given, which are to be matched with each other. These questions are also provided with four options, out of which one has the correct set of matched entities (Fig. 1(b)). In MCQN, the question has four choices, out of which the correct one is identified after solving the numerical stated in the question (Fig. 1(c)). The NUM type questions have numerical answers, rounded to the nearest integer or floating-point number as specified in the questions (Fig. 1(d)).



<p>Floatation beneficiation is based on the principle of (A) <u>Mineral surface hydrophobicity</u> (B) Gravity difference (C) Chemical reactivity (D) Particle size difference</p>	<p>A peak in the X-ray diffraction pattern is observed at $2\theta = 78^\circ$, corresponding to {311} planes of an fcc metal, when the incident beam has a wavelength of 0.154 nm. The lattice parameter of the metal is approximately (A) 0.6 nm (B) <u>0.4 nm</u> (C) 0.3 nm (D) 0.2 nm</p>
(a) Multiple choice question (MCQ)	(c) Numerical question with multiple choices (MCQN)
<p>Match the composite in Column I with the most suitable application in Column II.</p> <p>Column I : [(P) Glass fibre reinforced plastic, (Q) SiC particle reinforced Al alloy, (R) Carbon-carbon composite, (S) Metal fibre reinforced rubber] Column II: [(1) Missile cone heads, (2) Commercial automobile chassis, (3) Airplane wheel tyres, (4) Car piston rings, (5) High performance skate boards] :</p> <p>(A) P-4, Q-5, R-1, S-2 (B) P-3, Q-5, R-2, S-4 (C) <u>P-5, Q-4, R-1, S-3</u> (D) P-4, Q-2, R-3, S-1</p>	<p>The third peak in the X-ray diffraction pattern of a face-centered cubic crystal is at 2θ value of 45°, where 2θ is the angle between the incident and reflected rays. The wavelength of the monochromatic X-ray beam is 1.54 Å. Considering first-order reflection, the lattice parameter (in Å) of the crystal is? (Round off to two decimal places) Ans. <u>5.64 to 5.73</u></p>
(b) Matching type question (MATCH)	(d) Numerical question (NUM)

Fig. 1 Sample questions from each category: (a) multiple choice question (MCQ), (b) matching type question (MATCH), (c) numerical question with multiple choices (MCQN), and (d) numerical question (NUM). The correct answers are in bold and underlined.

To understand the performance of LLMs from the materials science domain perspective, we classified the questions into 14 categories. The list of categories was prepared in consultation with domain experts who teach materials science subjects at the institute where this research is conducted. Then, two experts assign all the questions to one of the categories. The conflict in the category assignments was resolved through discussion and mutual agreement. Fig. 2 shows the number of questions in each category. The color of the bars represents the broad

category of materials science topics under which each subtopic is shown in the graphical abstract. The database can be accessed at <https://github.com/M3RG-IITD/MaScQA>.

Solutions using LLMs

In this work, we benchmark the question-answering ability of LLaMA 2-70B³⁷ chat model (will be referred to as LLaMA-70B from now onwards in the paper), GPT-3.5, and GPT-4 models

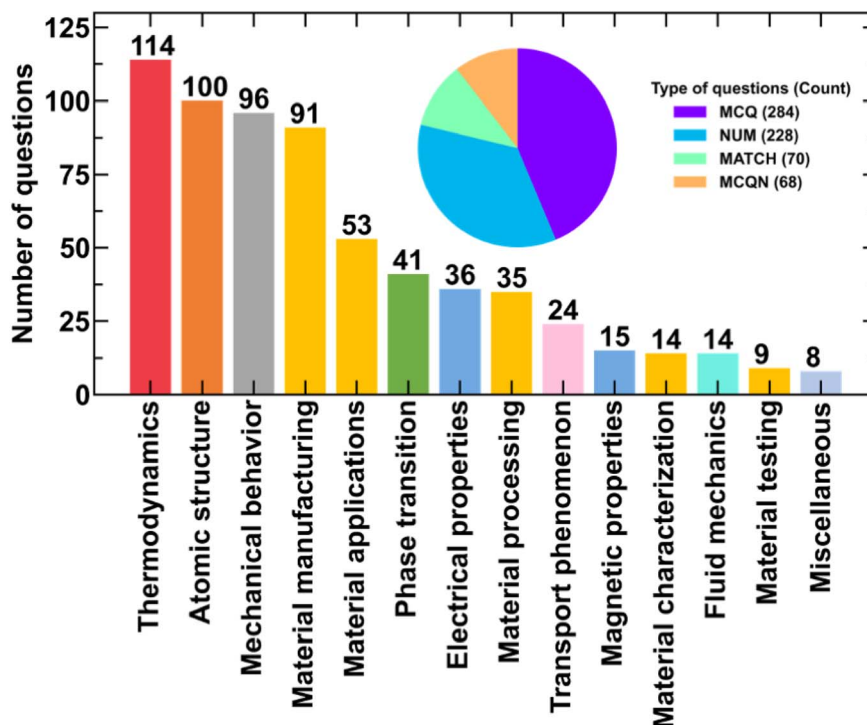


Fig. 2 The number of questions in each materials science sub-domain. The bar chart shows the number of questions in different sub-domains. The pie chart shows the number of questions classified according to question structure.



prompting,⁴⁰ self-consistency,⁴¹ and self-critique.^{42,43} However, in this work, we choose the basic prompting methodologies like zero-shot and CoT to investigate existing knowledge in LLMs and present baselines on the newly introduced dataset of MaScQA.

The solutions to all the questions obtained using two approaches for both models can be accessed at <https://github.com/M3RG-IITD/MaScQA>. The official answer keys are obtained from the official website of IIT Kharagpur, which is one of the organizing institutes of the GATE exam. https://gate.iitkgp.ac.in/old_question_papers.html. The LLMs' performance on two prompting methods is discussed in detail in the following sections.

Results

Dataset visualization

Fig. 2 shows the details of the dataset comprising a total of 650 questions in different categories. First, we categorize the

questions based on their structure. We observe that the largest category of questions (284) are MCQs, while 70 are MATCH-type questions. Further, 68 questions are MCQN, while the remaining 228 questions are NUM that do not provide any choices. We then analyze these questions from materials science domain perspective. To this extent, the questions are categorized into 14 domains: thermodynamics, atomic structure, mechanical behaviour, materials manufacturing, material applications, phase transition, electrical properties, material processing, transport phenomenon, magnetic properties, material characterization, fluid mechanics, and miscellaneous.

Fig. 2 shows the number of questions in different domain-specific categories. To visualize the frequently used words related to each domain-specific category of questions, word clouds are shown in Fig. 3 and top 10 most occurring words are shown in Fig. 4. The maximum number of questions (114) are in the *thermodynamics* category, which deals with questions related to enthalpy of formation, energy balance during

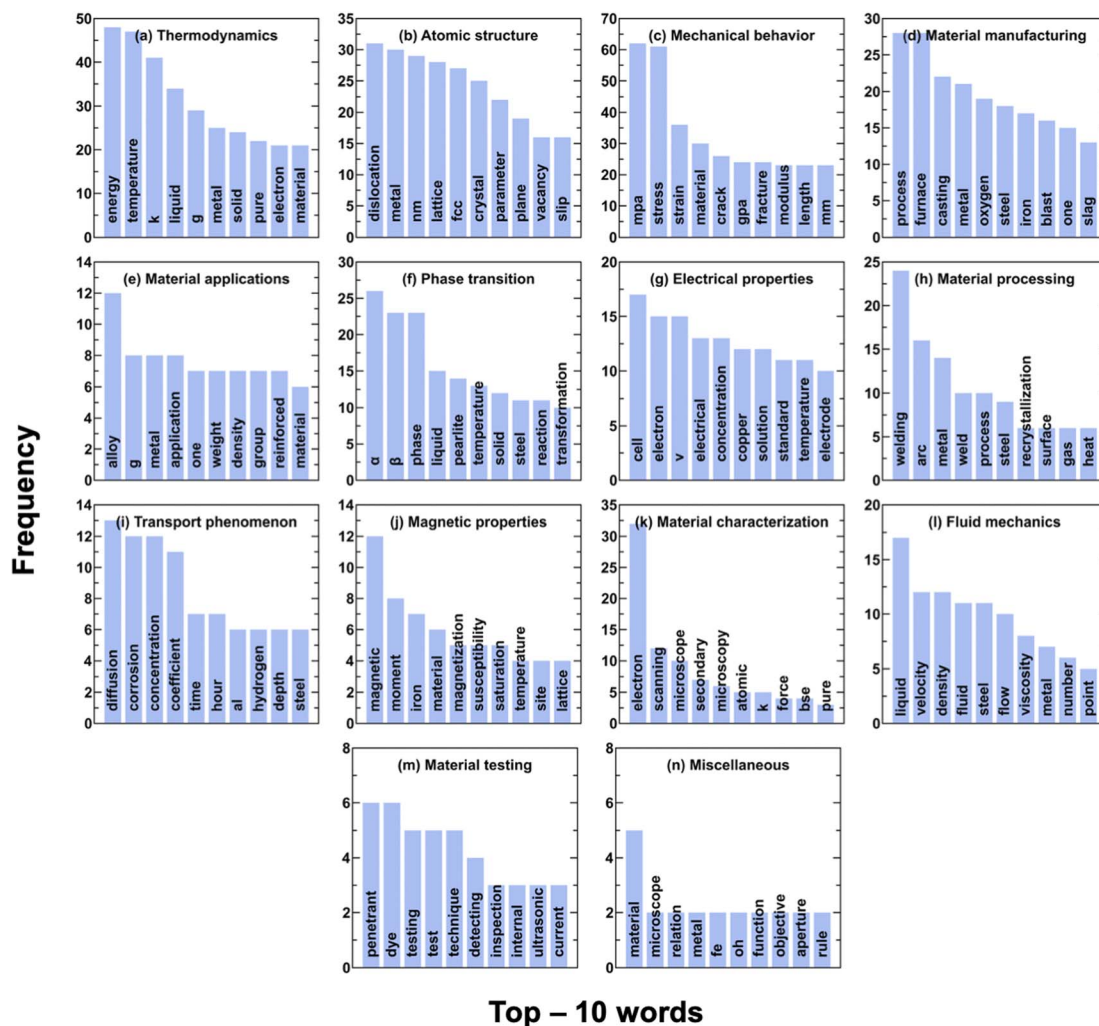


Fig. 4 Frequency of top – 10 words in each materials science subdomain present in MaScQA (a) thermodynamics, (b) atomic structure, (c) mechanical behavior, (d) material manufacturing, (e) material applications, (f) phase transition, (g) electrical properties, (h) material processing, (i) transport phenomenon, (j) magnetic properties, (k) material characterization, (l) fluid mechanics, (m) material testing, and (n) miscellaneous.



chemical reactions, transition temperatures, activation energy, and heat transfer (Fig. 3(a)) which is also reflected by most occurring words like energy, *k* (Kelvin), *j* (Joule), *g* (indicating gaseous state of chemicals in reactions and units of material properties), as shown in Fig. 4(a). The category of *atomic structure* comprises 100 questions, which are based on concepts such as dislocations, diffraction planes, and crystal structures (Fig. 3(b) and 4(b)). The *mechanical behavior* category is based on the concepts of stress-strain behavior of materials, creep, fatigue, and fracture mechanics (Fig. 3(c)). Further, the presence of words like “mpa and gpa” (Fig. 4(c)), which are units of stress and strength (MPa and GPa), indicate the correct classification of questions. In *materials manufacturing* (Fig. 3(d) and 4(d)) and *material applications* (Fig. 3(e) and 4(e)), the questions test the knowledge of extraction processes of materials from their respective ores and why a particular material, e.g. oxides, alloys are used for a specific application. Thus, these questions require logical understanding connecting multiple concepts: first, “recall” or “deduce” the properties of a material based on its composition, label, or processing conditions, and second, “identify” the properties required for a particular application and then connect these two concepts to “derive” a logical explanation to arrive at the correct answer. The questions on *phase transition* test the knowledge of how phase transition can be induced in materials, how to calculate the percentage of different phases in the materials, and the characteristics of different phases. This is also indicated by the high frequency of words related to different phases of materials (Fig. 3(f) and 4(f)). The questions on *electrical properties* include fuel cells, characteristics of materials used in batteries, and semiconductor devices (Fig. 3(g)). This is also seen in the frequency of top-10 words in this domain (Fig. 4(g)), which comprises of electron, *v* (Volt), and electrode. Then, questions are based on *material processing* such as welding, annealing, tempering, recrystallization, welding, etc. (Fig. 3(h) and 4(h)). The questions on transport phenomena test concepts related to the diffusion or transport of ions, corrosion, and duration of the phenomena (Fig. 3(i) and 4(i)). The question related to *magnetic properties* tests the knowledge about magnetization and the characteristics of different magnetic materials (Fig. 3(j) and 4(j)). The material characterization topic has questions related to methods like scanning electron microscopy, diffraction studies, and back-scattered electron microscopy (Fig. 3(k) and 4(k)). The fluid mechanics topic comprises questions on the viscosity of the fluid and the movement of particles in a viscous medium

(Fig. 3(l) and 4(l)). In the *material testing* topic, the questions are based primarily on non-destructive material testing methods (Fig. 3(m) and 4(m)). The miscellaneous category deals with questions requiring a simultaneous understanding of multiple materials science domains like optical properties, piezoelectricity, and microscopy for their solution (Fig. 3(n) and 4(n)).

Performance evaluation

Now, we evaluate the performance of LLMs on MaScQA and the effect of prompting methods on the performance, corresponding to the first two questions posed in this work. Table 2 reports the accuracy of the LLMs on the MaScQA corpus. The scores corresponding to model names GPT-3.5 and GPT-4 represent the accuracy of the models when questions are asked directly to the models representing zero-shot answering. The model names with the suffix “CoT” imply we have asked the models to provide detailed “stepwise” solutions to the given questions. In MCQs, we observe that GPT-4 significantly outperforms GPT-3.5 and LLaMA. We observed that LLaMA yields very low performance, which might be due to limited training corpora and fewer parameters than GPT models. Further, we also observe that the CoT provides only marginal improvement in the result for GPT-3.5 and GPT-4.

Here, GPT-4-CoT gives an accuracy of 77.11% on MCQ, which is a high score considering the difficulty levels of this exam. Also, the performance of GPT-4-CoT is ~20% higher than GPT-3.5-CoT for MCQ type of questions. For MATCH questions, GPT-4-CoT exhibits the maximum performance with a score of 92.86%, a very high score considering the amount of knowledge required to connect the entities. In contrast, the variants of GPT-3.5 performed poorly on MATCH questions, with a score of 40% and 38.57% for the variants without and with CoT, respectively. In this case, the GPT-4-CoT provides ~4% improvement over direct prompting. For MCQN, GPT-4 gives the best performance with a score of 58.82%, while CoT reduces the model's performance to 50.0%. The same trend of reduced performance on these questions is observed with the GPT-3.5 model. This implies that CoT prompting may not always lead to better performance. Now, we focus on the numerical questions. Among all the categories, models exhibit the worst performance in the NUM category. Here, GPT-4 and GPT-4-CoT obtain the maximum score of 37.28% and 39.04%. Interestingly, we observe that CoT yields poorer results in the case of GPT-3.5, while it yields better accuracy in the case of GPT-4.

Table 2 Performance (% accuracy) of different evaluation styles using LLaMA and GPT models on various question types. The number in parenthesis represents the total number of questions under respective categories

Evaluation method	MCQ (284)	Matching (MATCH) (70)	Numerical with MCQ (MCQN) (68)	Numerical (NUM) (228)	Overall accuracy
Baseline scores	25	25	25	0	
LLaMA-70B-CoT	41.20	22.86	20.59	3.95	24.0
GPT-3.5	56.69	40.00	35.29	15.79	38.31
GPT-3.5-CoT	57.04	38.57	33.82	14.91	37.85
GPT-4	74.65	88.57	58.82	37.28	61.38
GPT-4-CoT	77.11	92.86	50.00	39.04	62.62



Table 3 *p*-Values obtained from statistical testing of LLMs performance using paired *t*-test

LLMs	GPT-3.5	GPT-3.5-CoT	GPT-4	GPT-4-CoT
GPT-3.5-CoT	0.864			
GPT-4	3.56×10^{-17}	7.96×10^{-18}		
GPT-4-CoT	6.12×10^{-19}	1.26×10^{-19}	0.648	
LLAMA-70B-CoT	2.17×10^{-8}	5.75×10^{-8}	1.76×10^{-42}	2.17×10^{-48}

Finally, regarding overall performance, GPT-4-CoT gives the best score of 62.62%, with GPT-4 following closely at 62%. It should be noted that in MCQ, there are 13 questions where more than one option is correct, of which GPT-4 and GPT-4-CoT answered six and seven questions correctly, respectively. Interestingly, we observe that CoT does not always give improved results. In fact, for GPT-3.5, CoT gives poorer results in all the cases except MCQs and marginally better results for GPT-4 in all the cases except MCQN. Note that this observation contrasts with the general observation that the CoT prompting results in improved performance of LLMs on QA tasks. To identify whether the overall performance of LLMs-based evaluation strategies on MaScQA is statistically significant, we perform paired *t*-test by taking the performance of two evaluation strategies at a time and report the resulting *p*-values in Table 3. The null hypothesis tested is “there is no significant difference between the performance of two LLMs-based evaluation strategies in solving the questions of MaScQA”. Since the *p*-values when comparing the performance of GPT-3.5 with GPT-3.5-CoT and GPT-4 with GPT-4-CoT are quite higher than 0.05, it is

accepted that in these two cases, there is no significant difference between the performance of the two evaluation strategies. However, in all other cases, the *p*-values are lower than 0.05, implying significant difference in the performance of the LLMs-based evaluation strategies.

In addition to evaluating the performance of LLMs in answering different types of questions like MCQ, MATCH, MCQN, and NUM, which test different abilities of the students, it is also essential to analyze the performance of the models from a domain perspective. To this end, we classify all the questions of our dataset into 14 broad categories. Fig. 5 shows the accuracy of the GPT-4-CoT prompting method while answering the questions. Since the number of questions differs under each category, we report the percentage of questions answered correctly and incorrectly to show proper comparison. The number of question for each case are written with white color inside the respective bars. It is observed that questions related to materials' mechanical behavior and electrical properties have the most percentage of incorrectly answered questions (~60%). The questions on thermodynamics, atomic

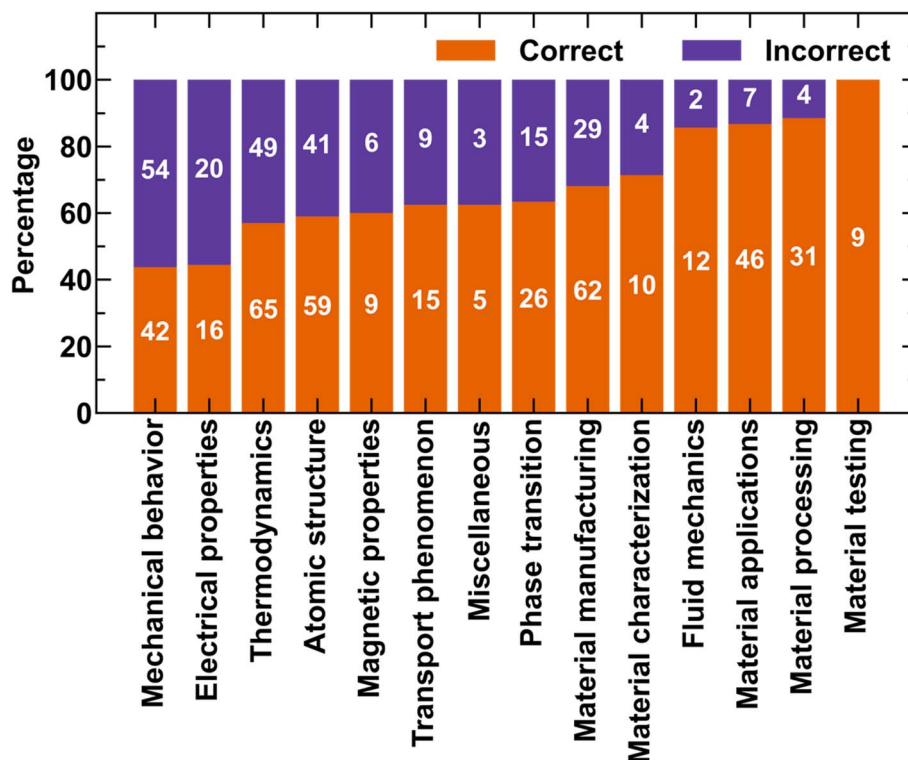


Fig. 5 Performance of GPT-4-CoT on questions classified from materials science domain perspective.



structure, magnetic properties, transport phenomena, and phase transition have ~40% of incorrectly answered questions in the respective categories. Further, ~30% of materials manufacturing and characterization questions are incorrectly answered. In the categories of fluid mechanics and materials applications, ~15% of questions are incorrectly answered with the lowest error rates for material processing and no mistakes made on material testing questions. To further gain insights into the factors limiting LLMs' performance, we will discuss them by classifying the errors into two categories, as explained in the Discussion section.

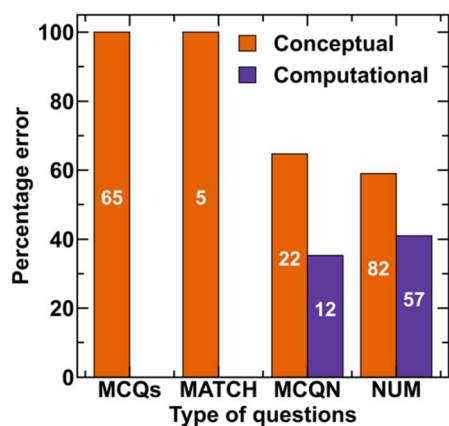


Fig. 6 Types of errors made by GPT-4-CoT on the questions classified based on the structure.

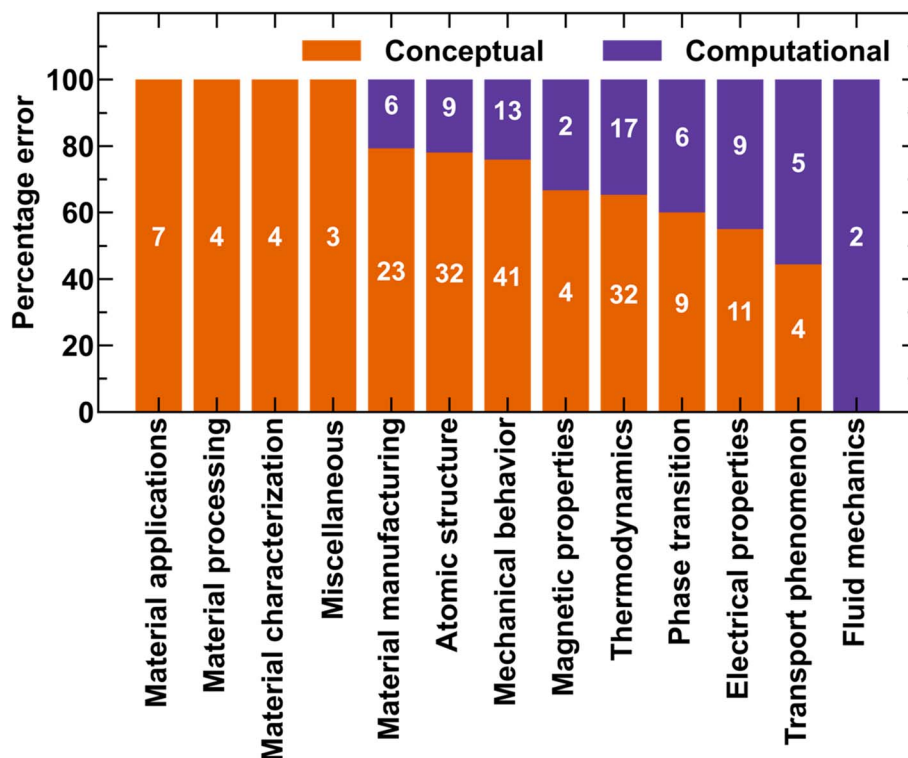


Fig. 7 Types of the error made by GPT-4-CoT on questions classified according to materials science perspective.

Discussion

Error analysis

To use LLMs effectively for materials discovery and identify areas that require further research, it is important to understand the mistakes made by the LLMs in the materials domain. Answering a question requires retrieval of correct concepts/facts, applying them to the scenarios posed in the question by appropriate substitution in the relevant formulae, and then solving it correctly by applying relevant computational steps. To understand further, we can divide these errors into three categories: namely, (i) conceptual error, where the correct concept, equation, or facts related to the problem are not retrieved, or the LLM hallucinates some facts; (ii) grounding error: where the relevant concepts are not correctly applied to the scenario or incorrect values are substituted in the equations (for example, °C to K conversion not applied) and (iii) computational error: where the numerical computation is performed incorrectly.³⁵ Note that CoT prompting enables the model to reflect upon the knowledge it already has, connect it with multiple choices, and then arrive at the answer. Thus, in general, it has been observed that CoT helps in reducing grounding errors (in our case, it virtually eliminates them).

To analyze different errors, we perform error analysis on GPT-4-CoT response because this strategy performed best on MaScQA. We take all the incorrectly answered questions by GPT-4-CoT, in which 139 are NUM, 65 are MCQs, 34 are MCQN, and 5 are matching-type questions (MATCH) (Fig. 6). The number of incorrectly answered questions across materials science sub-



domains are shown in Fig. 7. Note that there may be questions with conceptual and numerical errors, but we have considered only the conceptual error in these questions since it is the first to be found. If the retrieved concept is incorrect, we deem the computational error secondary.

Fig. 6 shows the distribution of errors made by GPT-4-CoT in different categories of question based on their structure. The text inside the bars representing conceptual and computational error shows the number of questions in respective category. The analysis of all the incorrectly answered questions reveals that majority of errors are conceptual. Further, in MCQs and MATCH type questions, the error is always conceptual because answering such questions requires retrieval of appropriate concepts and facts and then connecting them with relevant options. For MCQN and NUM, majority of the questions were answered incorrectly (~65% and ~59%) due to conceptual errors implying the need for domain-specific models or better prompting and problem-solving approaches.

As mentioned earlier, we observed that GPT-4-CoT made no grounding errors. To evaluate whether this is due to the effectiveness of CoT, we investigate questions that are incorrectly answered by GPT-4 and correctly by GPT-4-CoT. Out of 66 such questions from the entire dataset, GPT-4's solutions had ~70% conceptual errors, ~30% computational errors, and no grounding errors. Further, we also analyzed the erroneously answered questions by GPT-4-CoT and are correctly answered by GPT-4. There were 58 such questions in the complete dataset. Out of these questions, solutions of 45 questions (~78%) had conceptual errors; for one question, there was a grounding error, and the remaining 12 questions had computational errors when solved using GPT-4-CoT. Since there are little to no grounding errors in either GPT-4 or GPT-4-CoT, both models are adept in this regard. The CoT prompting is helping reduce some numerical errors.

Fig. 7 shows the domain-wise distribution of conceptual and computational errors on the all the incorrectly answered

(a) Thermodynamics	At 1000 K, the linear thermal expansion coefficients of graphite, parallel and perpendicular to the graphite layers, are $0.8 \times 10^{-6} \text{K}^{-1}$ and $29 \times 10^{-6} \text{K}^{-1}$, respectively. The percentage increase in the volume of graphite when heated from 900 K to 1100 K is? (round off to 2 decimal places) <u>(Ans. 0.6 to 0.62)</u>
(b) Atomic structure	Chromium has the bcc structure with atomic diameter of 2.494 Å. Find the first diffraction peak position (2θ) for Cu $K\alpha$ radiation with a wavelength of 1.54 Å (A) 21.76° (B) 33.05° <u>(C) 44.43°</u> (D) 66.10°
(c) Electrical properties	Identify the correct combination of the following statements: P. Hydrogen electrode is a standard used to measure redox potentials Q. Activation polarization refers to electrochemical processes controlled by reaction sequence at metal-solution interface R. Potential-pH diagrams can be used to predict corrosion rates of metals S. Cathodic protection can use sacrificial anodes such as magnesium (A) P, Q and R (B) Q, R and S <u>(C) P, Q and S</u> (D) P, R and S
(d) Magnetic properties	Calculate the saturation magnetization (M_{sat}) for bcc iron of lattice parameter 2.866 Å. (A) $0.79 \times 10^6 \text{ A/m}$ (B) $1.5 \times 10^6 \text{ A/m}$ (C) $3.15 \times 10^6 \text{ A/m}$ <u>(D) $4.73 \times 10^6 \text{ A/m}$</u>
(e) Material manufacturing	Match the reactors / refining sites in Column I with the corresponding refining processes in Column II. Column I: [(P) Blast furnace runner, (Q) AOD, (R) Torpedo car, (S) BOF] Column II: [1. De-carburization, 2. External De-sulfurization, 3. De-phosphorization, 4. External De-siliconization] Options: <u>(A) P-4, Q-1, R-2, S-3</u> (B) P-4, Q-2, R-3, S-1 (C) P-2, Q-1, R-4, S-3 (D) P-1, Q-3, R-2, S-4

Fig. 8 Visualizing some of the questions where GPT-4-CoT made conceptual errors in the solution. The correct answers to each question are marked in bold and underlined.



questions by GPT-4-CoT. The number written in white color over colored bars represent the number of question in each case. All categories have conceptual errors in more than 50% of the respective questions except for transport phenomena (~45%) and fluid mechanics. Now, we will discuss some conceptual errors in different domains. The list of all questions subjected to analysis is provided in the GitHub repository of this work.

Fig. 8(a) shows an example of the conceptual error made on a question related to thermodynamics. In this question, instead of considering the coefficient of thermal expansion same in the planar dimension, it considered the coefficient of thermal expansion in the perpendicular direction as the same in one of the planar directions. Mathematically, instead of obtaining the final coefficient using $2 \times \text{parallel} + \text{perpendicular}$ coefficients, GPT-4-CoT used $\text{parallel} + 2 \times \text{perpendicular}$, leading to an incorrect answer. While solving a question on atomic structure, as shown in Fig. 8(b), GPT-4-CoT mistook the relation between lattice parameter (a) and atomic diameter (D) as $a = \frac{\sqrt{3}}{2}D$ instead of $a = \frac{2}{\sqrt{3}}D$. In a question on the electrical properties of materials (Fig. 8(c)), the GPT-4-CoT answered that all the given statements were correct. Hence, it could not choose from the four options shown as answers. According to the materials science domain and the Wikipedia entry of Pourbaix diagrams, one of their major limitations is that these diagrams do not estimate actual corrosion rates; also, these diagrams cannot be used while studying corrosion due to chloride ions. Hence, the statement R is incorrect, making (C) the correct choice. While solving the question shown in Fig. 8(d), GPT-4-CoT did not convert the lattice parameter into the atomic diameter and considered them the same while using it in the formula required for solving the problem. For a question on materials manufacturing (Fig. 8(e)), GPT-4-CoT retrieved the functions of (P) blast furnace slag and (R) Torpedo car as opposite, thus leading to a wrong answer, C , when the correct option was A . The complete solution of GPT-4-CoT can be found in the GitHub repository of this work. Some examples of correct answers given by GPT-4-CoT on four types of question according to structure (MCQ, MATCH, MCQN, and NUM) are shown in ESI (Fig. S2–S5).†

To summarise, the CoT prompting cannot significantly improve the LLM performance as the mistakes are mainly conceptual. This makes a strong case for a domain-specific LLM for materials and potentially domain-specific alternate prompting strategies. Further, for questions where the LLMs give the incorrect response due to computational error, the solution involved unit conversions, logarithms, and exponentials and had numbers with multiplying factors (e.g., 10^{10}). There have been recent works in the literature that suggest methods for improving calculations and for improving on concept-based mistakes.⁴⁴ Introducing such heuristics while prompting can help researchers in two ways: (1) probe the existing LLMs more deeply and (2) generate datasets to train LLMs with lesser parameters, thus making the use of these

Table 4 Comparing the performance of GPT-4-CoT with human performance

Year	Maximum marks	Cut off marks	Average marks	GPT-4-CoT
2020	83	49.2	N.A.	46.46
2021	87.67	48.5	28.7	42.86
2022	77.67	46.2	27.6	38.62

models economical. Hence, this answers the third research question (limiting factors for LLMs) raised in this work.

Comparison with human performance

Based on the reports published by organising institutes of GATE, marking criteria is as follows: for NUM questions, there is no negative marking. For all other types of questions, there is a negative marking of $1/3$ times the marks of the question. The questions can carry 1 or 2 marks. Further, GATE for the materials science domain has only 25 questions, which is too few to compare with human performance. Therefore, we consider questions asked in the years 2020, 2021, and 2022 in the metallurgical engineering exam, where 65 questions are asked yearly. Out of 65 questions, 10 are of general aptitude and, hence, ignored in this work. By considering the remaining questions (119) and associated marks (185), GPT-4-CoT obtained 79 marks, translating to an average of 42.7% marks over three years. Table 4 shows the maximum marks obtained by humans in the GATE metallurgical engineering exam, the cut-off marks required to qualify, and the average of the marks obtained by students who appeared. It can be concluded that GPT-4 is better than an average student appearing in the exam and comes quite close to the cut-off required for qualification.

Additional tasks based on question-answering

In this section, we evaluate the performance of LLMs on two additional tasks that enable accelerated materials modelling and discovery, namely, composition extraction from tables in materials science articles and code-writing for materials modelling. Note that both the problems are formulated as question-answering tasks and hence evaluate the ability of LLMs to answer materials science domain questions consistent with the previous section.

Compositions extraction from tables in materials science research papers

The understanding of materials compositions, their processing, and testing conditions, structure, and properties form the basis of automated material discovery pipelines.⁴⁵ According to Gupta *et al.* (2023),¹⁷ ~85% of materials composition in existing databases are extracted from tables. Gupta *et al.* (2023) developed a graph neural network based pipeline, DiSCoMaT, which can extract materials compositions from the tables published in materials science research papers.¹⁷ In this work, we sample 100 compositions from the dataset, which is used to evaluate the



(a) System message: Can you please extract the unique material compositions from this table by considering information from the caption as well? For example, find the value of one compound from the table and, put it in the expression given in caption and solve the expression for obtaining composition.

Note: If compositions are reported as single strings, extract the value of each constituent such that the sum of the value of all constituents is 100. The missing chemical component values should be replaced with 0. The material-ids, component names, and units are strings. The values are floating point numbers. Do not extract unwanted information. The missing composition unit should be considered as mol%. The output should be in the form of comma separated list of tuples in the following format:

```
(((material id-1, chemical component -1, value, unit), (material id-1,chemical component-2, value, unit)), ((material id-2, chemical component -1, value, unit), (material id-2,chemical component-2, value, unit)),...].
```

(b) Table and its caption as prompt

Table caption: Glass compositions in weight percentage (wt%).

```
Table: [['Glass', 'Na2O', 'K2O', 'MgO', 'CaO', 'B2O3', 'P2O5', 'SiO2'],
        ['Bioactive', '6', '12', '5', '20', '-', '4', '53'],
        ['Biocompatible 1', '25.5', '-', '-', '11', '1.3', '2.5', '59.7'],
        ['Biocompatible 2', '16.6', '5.0', '10.3', '6.6', '1.0', '2.0', '58.5']]
```

Fig. 9 (a) System message (b) table as prompt to extract materials compositions from tables using GPT-4 API.

performance of DiSCoMaT and compare it with the compositions extracted by GPT-4 from the same tables. The table types and the prompts are given in the GitHub repository. Different prompts are used for different tables (see ref. 17 for different table types) to improve the results of the LLM. For example, Fig. 9 shows the system message provided in the API for extracting compositions and the table, and its caption converted into the prompt.

In the composition extraction task, the extracted compositions must be expressed as a set of tuples containing material ID (as defined in the paper), constituent elements or compounds, corresponding percentage contributions, and corresponding units. To evaluate the performance on this task, two categories

of metrics are used: *tuple level* precision, recall and F1-scores, and *material level* precision, recall, and F1-scores. Tuple level metrics imply that the individual components of a material are extracted along with its value and unit. In contrast, the material level metrics also consider extracting material id in the extracted tuple. For more details about the extraction task and metrics, the readers are requested to refer to the paper introducing this dataset and models.¹⁷ Table 5 shows the performance of GPT-4 along with the DiSCoMaT's performance on the same dataset. Since GPT-4 is not particularly trained for this task, it produces extra text like “*The extracted compositions are...*” which is incompatible with the evaluation pipeline used in DiSCoMaT. Therefore, we analyse only the relevant part of the extractions from the GPT-4 output. The lower performance of GPT-4 compared to DiSCoMaT can be attributed to the fact that GPT-4 was not pre-trained/finetuned for such tasks. The mistakes made by GPT-4 include non-extraction of material ids, not being able to normalise the component values if the sum of all components is not 100, and not being able to extract nominal compositions when both nominal and experimental compositions are reported in the table. These mistakes, thus,

Table 5 Comparing the performance of GPT-4 and DiSCoMaT on the composition extraction task

Model	Tuple level metrics			Material level metrics		
	Precision	Recall	F1-score	Precision	Recall	F1-score
GPT-4	76.39	76.0	76.2	57.45	51.92	54.55
DiSCoMaT	83.24	66.33	73.80	88.18	62.50	73.11



<pre>def claussius(HVap, T1, P1, T2): """ This function returns the phase transition pressure at temperature T2 given a heat of vaporization HVap, and and reference temperature and pressure T1 and P1 """ [insert] return P2</pre>	<pre>def claussius(HVap, T1, P1, T2): """ This function returns the phase transition pressure at temperature T2 given a heat of vaporization HVap, and and reference temperature and pressure T1 and P1 """ import math R = 8.314 # Universal gas constant in J/(mol*K) P2 = P1 * math.exp((HVap/R) * (1/T1 - 1/T2)) return P2</pre>
--	--

Fig. 10 (a) An example prompt provided to the GPT-4 model for generating the complete output (b) response of GPT-4.

Table 6 Comparing the performance of different LLMs on code generation task

Model	Performance
Openai/text-davinci-003 (ref. 36)	60.49%
Openai/code-davinci-002 (ref. 36)	53.09%
Openai/code-cushman-001 (ref. 36)	56.79%
Openai/gpt-4-0613 (GPT-4)	71.60%

constitute both computational and factual errors, as investigated in the Discussion section of this paper. The dataset of 100 compositions, prompts used for this study and response of GPT-4 are provided in the GitHub repository of this work.

Code writing

One of the important usages of LLMs for materials discovery could be in developing codes for materials simulation.⁴⁵ To evaluate this capability, we obtain the performance of GPT-4 on the code completion dataset provided by White *et al.* (2023).³⁶ Although this dataset was prepared to evaluate code-generating LLM's understanding of chemistry, the questions belonging to categories like thermodynamics, molecular simulations, spectroscopy, and atomic structure are common with that of materials science. In this work, the questions are provided to GPT-4 with the system message "Complete the following code by following the docstring and replacing [insert]." followed by the prompt which contains the skeleton of the python function with the docstring, [insert] marker and return statement. An example of the prompt is shown in Fig. 10(a), along with the solution provided by GPT-4 (Fig. 10(b)).

The performance of GPT-4, compared to the output of other models, is reported in Table 6. It was observed that most of the mistakes made by GPT-4 are in the codes related to molecular dynamics, spectroscopy, chemical informatics, and quantum mechanics, which is consistent with the performance of GPT-4-CoT on MaScQA. Another interesting observation is that GPT-4 answered all code-related thermodynamics questions (a total of ten questions), which is consistent with the observation that GPT-4 has a reasonable understanding of thermodynamics concepts, and the poor performance on MaScQA was mainly due to computational error. We have provided the output of GPT-4 on all the questions in the GitHub repository of this work.

Altogether, we observe that GPT-4 achieves state-of-the-art performance for the code writing task.

Conclusion

This work evaluated how well LLMs understand the materials science domain to determine their applications in materials discovery, synthesis, and planning pipelines. To this end, our new dataset, MaScQA, annotated questions and answers on the materials science domain, will provide a means to gain deeper insights. Evaluation of LLMs on MaScQA revealed that the LLMs make both numerical and conceptual mistakes. There are several core materials science domains where LLMs show poor performance, such as the atomic and crystal structure of materials and their electrical, magnetic, and thermodynamic behavior. Further, we evaluated the ability of LLMs for advanced tasks such as composition extraction from tables and code writing. These tasks require LLMs to have domain insights and the ability to produce output in the desired format, thus testing their conceptual, grounding, and computational capabilities. While GPT-4 performs poorly on the composition extraction task, it outperformed all other models on the code writing tasks.

Interestingly, the results suggest that domain-adaptation and task-specific prompting strategies are necessary to elicit the desired output from the LLMs. Therefore, the language models must be finetuned on a domain-specific and task-specific datasets to enable the use of LLMs in the materials discovery pipeline. Moreover, the performance of the LLMs on MaScQA can enable a deeper understanding of the lacunae of materials science knowledge in the LLMs, thereby providing new research avenues. For instance, LLMs' poor performance in NUM questions suggests that a pipeline connecting the LLM to a math calculator can potentially yield improved results. Further, the conceptual mistakes made by the LLMs indicate that the development of an LLM trained on materials literature could provide improved results. The materials science domain is a field that derives concepts from physics, chemistry, and mechanics. Therefore, a benchmark like MaScQA will allow the researchers to benchmark their domain specific models and prompting strategies against a standard dataset. Further, the correct solutions can help researchers create a new dataset for training lightweight or small language models, which are economical and, hence, can be easily deployed on low-memory industrial devices for materials discovery and their usage for educational purposes.



Data availability

The code, data, and links used in our work titled as MaScQA: a question answering dataset for investigating materials science knowledge of large language models authored by Mohd Zaki, Jayadeva, Mausam, and N. M. Anoop Krishnan can be found at <https://github.com/M3RG-IITD/MaScQA>.

Conflicts of interest

The authors declare no competing financial interest.

Acknowledgements

N. M. A. K. acknowledges the funding support received from BRNS YSRA (53/20/01/2021-BRNS), ISRO RESPOND as part of the STC at IIT Delhi, and the Google Research Scholar Award. M. Z. acknowledges the funding received from the PMRF award by the Ministry of Education, Government of India. M. acknowledges grants by Google, IBM, Microsoft, Wipro, and a Jai Gupta Chair Fellowship. The authors acknowledge the assistance of Mr Aditya Pratap Singh (B. Tech. student in the Department of Materials Science and Engineering, IIT Delhi) in compiling GATE questions from previous year papers, and Ms Devanshi Khatsuriya (B. Tech. student in the Department of Computer Science and Engineering, IIT Delhi) in evaluating the performance of GPT-4 on composition extraction task. The authors thank Microsoft Accelerate Foundation Models Research (AFMR) for access to OpenAI models. The authors thank the High Performance Computing (HPC) facility at IIT Delhi for computational and storage resources.

References

- 1 J. Devlin, M. W. Chang, K. Lee and K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in *Proceedings of NAACL*, Association for Computational Linguistics, Minneapolis, Minnesota, 2019, p. 4171–4186, available from: <https://www.aclweb.org/anthology/N19-1423>.
- 2 A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, *et al.*, PaLM: Scaling Language Modeling with Pathways, *arXiv*, 2022, preprint, arXiv:2204.02311 [cs.CL], DOI: [10.48550/arXiv.2204.02311](https://doi.org/10.48550/arXiv.2204.02311).
- 3 C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, *et al.*, Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer, *arXiv*, 2020, preprint, arXiv:1910.10683v4 [cs.LG], DOI: [10.48550/arXiv.1910.10683](https://doi.org/10.48550/arXiv.1910.10683).
- 4 A. Kedia, S. C. Chinthakindi and W. Ryu, Beyond reptile: meta-learned dot-product maximization between gradients for improved single-task regularization, in *Findings of the association for computational linguistics: EMNLP 2021*, Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021, p. 407–420, available from: <https://aclanthology.org/2021.findings-emnlp.37>.
- 5 B. Pang, E. Nijkamp, W. Kryściński, S. Savarese, Y. Zhou and C. Xiong, Long Document Summarization with Top-down and Bottom-up Inference, *arXiv*, 2022, preprint, arXiv:2203.07586v1 [cs.CL], DOI: [10.48550/arXiv.2203.07586](https://doi.org/10.48550/arXiv.2203.07586).
- 6 A. Fan, S. Bhosale, H. Schwenk, Z. Ma, A. El-Kishky, S. Goyal, *et al.*, Beyond english-centric multilingual machine translation, *Journal of Machine Learning Research*, 2021, 22, 107.
- 7 OpenAI R. Gpt-4 technical report, *arXiv*, 2023, preprint, arXiv:2303.08774v4, DOI: [10.48550/arXiv.2303.08774](https://doi.org/10.48550/arXiv.2303.08774).
- 8 H. Touvron, T. Lavril, G. Izacard, X. Martinet, M. A. Lachaux, T. Lacroix, *et al.*, LLaMA: Open and Efficient Foundation Language Models, *arXiv*, 2023, preprint, arXiv:2304.03277v1 [cs.CL], DOI: [10.48550/arXiv.2304.03277](https://doi.org/10.48550/arXiv.2304.03277).
- 9 B. Peng, C. Li, P. He, M. Galley and J. Gao, Instruction Tuning with GPT-4, *arXiv*, 2023, preprint, arXiv:2304.03277v1 [cs.CL], DOI: [10.48550/arXiv.2304.03277](https://doi.org/10.48550/arXiv.2304.03277).
- 10 M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. de Oliveira Pinto, J. Kaplan, *et al.*, Evaluating large language models trained on code, *arXiv*, 2021, preprint, arXiv:2107.03374v2 [cs.LG], DOI: [10.48550/arXiv.2107.03374](https://doi.org/10.48550/arXiv.2107.03374).
- 11 L. Weston, V. Tshitoyan, J. Dagdelen, O. Kononova, A. Trewartha, K. A. Persson, *et al.*, Named Entity Recognition and Normalization Applied to Large-Scale Information Extraction from the Materials Science Literature, *J. Chem. Inf. Model.*, 2019, 59(9), 3692–3702.
- 12 K. Cruse, A. Trewartha, S. Lee, Z. Wang, H. Huo, T. He, *et al.*, Text-mined dataset of gold nanoparticle synthesis procedures, morphologies, and size entities, *Sci. Data*, 2022, 9(1), 234.
- 13 V. Venugopal, S. Sahoo, M. Zaki, M. Agarwal, N. N. Goswami and N. M. A. Krishnan, Looking through glass: Knowledge discovery from materials science literature using natural language processing, *Patterns*, 2021, 2(7), 100290.
- 14 T. Gupta, M. Zaki, N. M. A. Krishnan and M. Mausam, MatSciBERT: a materials domain language model for text mining and information extraction, *npj Comput. Mater.*, 2022, 8(1), 102.
- 15 S. Huang and J. M. Cole, BatteryBERT: A Pretrained Language Model for Battery Database Enhancement, *J. Chem. Inf. Model.*, 2022, 62(24), 6365–6377.
- 16 S. Mysore, Z. Jensen, E. Kim, K. Huang, H. S. Chang, E. Strubell, *et al.*, The materials science procedural text corpus: annotating materials synthesis procedures with shallow semantic structures, in *Proceedings of the 13th linguistic annotation workshop*, Association for Computational Linguistics, Florence, Italy, 2019, p. 56–64, available from: <https://aclanthology.org/W19-4007>.
- 17 T. Gupta, M. Zaki, D. Khatsuriya, K. Hira, N. M. A. Krishnan and M. Mausam, DiSCoMaT: distantly supervised composition extraction from tables in materials science articles, in *Proceedings of the 61st annual meeting of the association for computational linguistics*, Association for Computational Linguistics, Toronto, Canada, 2023, vol. 1, p. 13465–13483, available from: <https://aclanthology.org/2023.acl-long.753>.



- 18 A. Trewartha, N. Walker, H. Huo, S. Lee, K. Cruse, J. Dagdelen, *et al.*, Quantifying the advantage of domain-specific pre-training on named entity recognition tasks in materials science, *Patterns*, 2022, 3(4), 100488.
- 19 P. Shetty, A. C. Rajan, C. Kuenneth, S. Gupta, L. P. Panchumarti, L. Holm, *et al.*, A general-purpose material property data extraction pipeline from large polymer corpora using natural language processing, *npj Comput. Mater.*, 2023, 9(1), 1–12.
- 20 J. Zhao, S. Huang and J. M. Cole, OpticalBERT and OpticalTable-SQA: Text- and Table-Based Language Models for the Optical-Materials Domain, *J. Chem. Inf. Model.*, 2023, 63(7), 1961–1981.
- 21 Y. Song, S. Miret and B. Liu, MatSci-NLP: evaluating scientific language models on materials science language tasks using text-to-schema modeling, in *Proceedings of the 61st annual meeting of the association for computational linguistics*, Association for Computational Linguistics, Toronto, Canada, 2023, vol. 1, p. 3621–3639, available from: <https://aclanthology.org/2023.acl-long.201>.
- 22 D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, *et al.*, Proceedings of the international conference learning, *Measuring massive multitask language understanding*, ICLR, 2021, p. 2049.
- 23 D. Hendrycks, C. Burns, S. Basart, A. Critch, J. Li, D. Song, *et al.*, Aligning AI with shared human values, *Proc. Int. Conf. Learn Represent ICLR*, 2021.
- 24 R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi and Y. Choi, HellaSwag: can a machine really finish your sentence?, in *Proceedings of the 57th annual meeting of the association for computational linguistics*, Association for Computational Linguistics, Florence, Italy, 2019, p. 4791–4800, available from: <https://aclanthology.org/P19-1472>.
- 25 K. Sakaguchi, R. Le Bras, C. Bhagavatula and Y. Choi, WinoGrande: An Adversarial Winograd Schema Challenge at Scale, *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34(5), pp. 8732–8740.
- 26 D. Dua, Y. Wang, P. Dasigi, G. Stanovsky, S. Singh and M. Gardner, DROP: a reading comprehension benchmark requiring discrete reasoning over paragraphs, in *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies*, Association for Computational Linguistics, Minneapolis, Minnesota, 2019, vol. 1, p. 2368–2378, available from: <https://aclanthology.org/N19-1246>.
- 27 K. M. Jablonka, Q. Ai, A. Al-Feghali, S. Badhwar, J. D. Bocarsly, A. M. Bran, *et al.*, 14 examples of how LLMs can transform materials science and chemistry: a reflection on a large language model hackathon, *Digital Discovery*, 2023, 2(5), 1233–1250.
- 28 T. Xie, Y. Wan, W. Huang, Y. Zhou, Y. Liu, S. Wang, *et al.*, DARWIN series: Domain specific large language models for natural science, *arXiv*, 2023, preprint, arXiv:2308.13565v1 [cs.CL], DOI: [10.48550/arXiv.2308.13565](https://doi.org/10.48550/arXiv.2308.13565).
- 29 Y. Song, S. Miret, H. Zhang and B. Liu, HoneyBee: Progressive instruction finetuning of large language models for materials science, *arXiv*, 2023, preprint, arXiv:2310.08511v1 [cs.CL], DOI: [10.48550/arXiv.2310.08511](https://doi.org/10.48550/arXiv.2310.08511).
- 30 K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, *et al.*, Training verifiers to solve math word problems, *arXiv*, 2021, preprint, arXiv:2110.14168v2 [cs.LG], DOI: [10.48550/arXiv.2110.14168](https://doi.org/10.48550/arXiv.2110.14168).
- 31 P. Clark, I. Cowhey, O. Etzioni, T. Khot, A. Sabharwal, C. Schoenick, *et al.*, Think you have solved question answering? try arc, the ai2 reasoning challenge, *arXiv*, 2018, preprint, arXiv:1803.05457v1 [cs.AI], DOI: [10.48550/arXiv.1803.05457](https://doi.org/10.48550/arXiv.1803.05457).
- 32 *ChemistryQA Data*, Microsoft, 2023, available from: <https://github.com/microsoft/chemistry-qa>.
- 33 P. Lu, S. Mishra, T. Xia, L. Qiu, K. W. Chang, S. C. Zhu, *et al.*, Learn to explain: Multimodal reasoning via thought chains for science question answering, in *The 36th conference on neural information processing systems (NeurIPS)*, 2022.
- 34 J. Welbl, N. F. Liu and M. Gardner, Crowdsourcing multiple choice science questions, *arXiv*, 2017, preprint, arXiv:1707.06209v1 [cs.HC], DOI: [10.48550/arXiv.1707.06209](https://doi.org/10.48550/arXiv.1707.06209).
- 35 D. Arora, H. Singh, M. Mausam, Proceedings of the 2023 conference on empirical methods in natural language processing, in *Have LLMs advanced enough? A challenging problem solving benchmark for large language models*, ed. Bouamor H., Pino J., Bali K., Association for Computational Linguistics, Singapore, 2023, pp. 7527–7543, Available from: <https://aclanthology.org/2023.emnlp-main.468>.
- 36 A. D. White, G. M. Hocky, H. A. Gandhi, M. Ansari, S. Cox, G. P. Wellawatte, *et al.*, Assessment of chemistry knowledge in large language models that generate code, *Digital Discovery*, 2023, 2(2), 368–376.
- 37 H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, *et al.*, Llama 2: Open Foundation and Fine-Tuned Chat Models, *arXiv*, 2023, preprint, arXiv:2307.09288v2 [cs.CL], DOI: [10.48550/arXiv.2307.09288](https://doi.org/10.48550/arXiv.2307.09288).
- 38 B. Workshop, T. L. Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilić, *et al.*, BLOOM: A 176B-Parameter Open-Access Multilingual Language Model, *arXiv*, 2023, preprint, arXiv:2211.05100v4 [cs.CL], DOI: [10.48550/arXiv.2211.05100](https://doi.org/10.48550/arXiv.2211.05100).
- 39 G. Penedo, Q. Malartic, D. Hesslow, R. Cojocar, A. Cappelli, H. Alobeidli, *et al.*, The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web Data Only, *arXiv*, 2023, preprint, arXiv:2211.05100v4 [cs.CL], DOI: [10.48550/arXiv.2306.01116](https://doi.org/10.48550/arXiv.2306.01116).
- 40 J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, *et al.*, Chain-of-thought prompting elicits reasoning in large language models, *Advances in Neural Information Processing Systems*, 2022, vol. 35, pp. 24824–24837.
- 41 X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, *et al.*, Self-Consistency Improves Chain of Thought Reasoning in Language Models, *arXiv*, 2023, preprint, arXiv:2203.11171v4 [cs.CL], DOI: [10.48550/arXiv.2203.11171](https://doi.org/10.48550/arXiv.2203.11171).
- 42 A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegrefe, *et al.*, Self-Refine: Iterative Refinement with



- Self-Feedback, *arXiv*, 2023, preprint, arXiv:2303.17651v2 [cs.CL], DOI: [10.48550/arXiv.2303.17651](https://doi.org/10.48550/arXiv.2303.17651).
- 43 N. Shinn, F. Cassano, E. Berman, A. Gopinath, K. Narasimhan and S. Yao, Reflexion: Language Agents with Verbal Reinforcement Learning, *arXiv*, 2023, preprint, arXiv:2303.11366v4 [cs.AI], DOI: [10.48550/arXiv.2303.11366](https://doi.org/10.48550/arXiv.2303.11366).
- 44 S. Gunasekar, Y. Zhang, J. Aneja, C. C. T. Mendes, A. Del Giorno, S. Gopi, *et al.*, Textbooks Are All You Need, *arXiv*, 2023, arXiv:2306.11644v2 [cs.CL], DOI: [10.48550/arXiv.2306.11644](https://doi.org/10.48550/arXiv.2306.11644).
- 45 M. Zaki, A. Jan, N. M. A. Krishnan and J. C. Mauro, Glassomics: An omics approach toward understanding glasses through modeling, simulations, and artificial intelligence, *MRS Bull.*, 2023, **48**(10), 1026–1039.

