Taylor & Francis
Taylor & Francis Group

Science and Technology of Advanced Materials: Methods

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/tstm20

Mining experimental data from materials science literature with large language models: an evaluation study

Luca Foppiano, Guillaume Lambard, Toshiyuki Amagasa & Masashi Ishii

To cite this article: Luca Foppiano, Guillaume Lambard, Toshiyuki Amagasa & Masashi Ishii (2024) Mining experimental data from materials science literature with large language models: an evaluation study, Science and Technology of Advanced Materials: Methods, 4:1, 2356506, DOI: 10.1080/27660400.2024.2356506

To link to this article: https://doi.org/10.1080/27660400.2024.2356506

SCIENCE AND TECHNOLOGY OF ADVANCED MATERIALS: METHODS
2024, VOL. 4, NO. 1, 2356506
https://doi.org/10.1080/27660400.2024.2356506

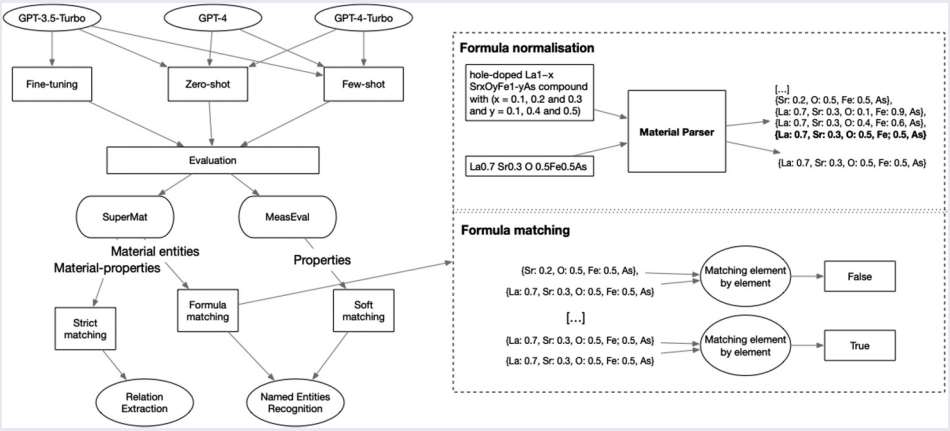
Mining experimental data from materials science literature with large language models: an evaluation study

Luca Foppiano [a,b], Guillaume Lambard [c], Toshiyuki Amagasa [b] and Masashi Ishii [c]

[a]Materials Modeling Group, Center for Basic Research on Materials, National Institute for Materials Science, Tsukuba-shi, Japan; [b]Knowledge and Data Engineering, Centre for Computational Sciences, University of Tsukuba, Tsukuba, Japan; [c]Data-driven Materials Design Group, Center for Basic Research on Materials, National Institute for Materials Science, Tsukuba-shi, Japan

ABSTRACT

This study is dedicated to assessing the capabilities of large language models (LLMs) such as GPT-3.5-Turbo, GPT-4, and GPT-4-Turbo in extracting structured information from scientific documents in materials science. To this end, we primarily focus on two critical tasks of information extraction: (i) a named entity recognition (NER) of studied materials and physical properties and (ii) a relation extraction (RE) between these entities. Due to the evident lack of datasets within Materials Informatics (MI), we evaluated using SuperMat, based on superconductor research, and MeasEval, a generic measurement evaluation corpus. The performance of LLMs in executing these tasks is benchmarked against traditional models based on the BERT architecture and rule-based approaches (baseline). We introduce a novel methodology for the comparative analysis of intricate material expressions, emphasising the standardisation of chemical formulas to tackle the complexities inherent in materials science information assessment. For NER, LLMs fail to outperform the baseline with zero-shot prompting and exhibit only limited improvement with few-shot prompting. However, a GPT-3.5-Turbo fine-tuned with the appropriate strategy for RE outperforms all models, including the baseline. Without any fine-tuning, GPT-4 and GPT-4-Turbo display remarkable reasoning and relationship extraction capabilities after being provided with merely a couple of examples, surpassing the baseline. Overall, the results suggest that although LLMs demonstrate relevant reasoning skills in connecting concepts, specialised models are currently a better choice for tasks requiring extracting complex domain-specific entities like materials. These insights provide initial guidance applicable to other materials science sub-domains in future work.



IMPACT STATEMENT

This research delves into the viability of employing Large Language Models (LLMs) for information extraction applied to the field of materials science. Through a comprehensive assessment of prominent models like GPT-3.5-Turbo, GPT-4, and GPT-4-Turbo, we aim to provide a preliminary assessment of their capabilities and constraints

ARTICLE HISTORY

Received 19 January 2024
Revised 10 April 2024
Accepted 6 May 2024

KEYWORDS

Large language models; benchmark; NER; TDM; evaluation; materials science

CONTACT Luca Foppiano luca@foppiano.org Materials Modeling Group, Center for Basic Research on Materials, National Institute for Materials Science, 1-1 Namiki, Tsukuba-shi, 305-0044, Japan

Supplemental data for this article can be accessed online at https://doi.org/10.1080/27660400.2024.2356506

1. Introduction

Mining experimental data from literature has become increasingly popular in materials science due to the vast amount of information available and the need to accelerate materials discovery using data-driven techniques. Data for machine learning in materials science is often sourced from published papers, material databases, laboratory experiments, or first-principles calculations [1]. The introduction of big data in materials research has shifted from traditional random techniques to more efficient, data-driven methods. Data mining of computational screening libraries has been shown to identify different classes of strong CO_2-binding sites, enabling materials to exhibit specific properties even in wet flue gases [2]. Machine learning techniques have been employed for high-entropy alloy discovery, focusing on probabilistic models and artificial neural networks [3]. However, the use of advanced machine learning algorithms in experimental materials science is limited by the lack of sufficiently large and diverse datasets amenable to data mining [4]. A present central tenet of data-driven materials discovery is that with a sufficiently large volume of accumulated data and suitable data-driven techniques, designing a new material could be more efficient and rational [5]. The materials science field is moving away from traditional manual, serial, and human-intensive work towards automated, parallel, and iterative processes driven by artificial intelligence, simulation, and experimental automation [6,7]. But, materials science literature is a vast source of knowledge that remains relatively unexplored with data mining techniques [8], especially for the reason that materials science data come in diverse forms such as unstructured textual content and structured tables and graphs, adding complexity to the extraction process. As a result, nowadays, many projects still depend on manual data extraction. While extensive structured databases contain accumulated experimental data [9], they remain limited in number and highly costly due to the amount of human labour involved [10].

Additionally, addressing issues related to the quality and meaning of materials science data often demands a curation step assisted by a sub-domain knowledge frequently specific to the approached subfield of materials science, e.g. polymers, metal-organic frameworks, high-entropy alloys, etc., with their own physical and chemical phenomena, methods and protocols, terminology and jargon. For instance, the classification of superconductors can be complex and sometimes arbitrary, blending compound-based classes like cuprates [11] and iron-based [12] materials with unconventional classes like heavy fermions [13]. The classification of superconductors can also be based on phenomena such as the Meissner effect, which describes how superconductors expel magnetic fields [14]. Superconductors can be divided into two classes according to how this breakdown occurs, the so-called type-I and type-II superconductors. As these classifications are not mutually exclusive certain materials could potentially fall into multiple categories, for example, a material can be both a cuprate and a type-II superconductor, the classification of superconductors is a complex task that demands an extensive knowledge of the developments and current state-of-the-arts. Moreover, substantial confusion may occur due to the cross-domain polysemy of used words, terms and symbols. In different sub-domains, the same term can take on specific nuances or meanings unique to a given sub-domain. This phenomenon is common in language and can lead to misunderstandings if the context of the sub-domain is unclear. For instance, the acronym 'TC' or 'TC' will be employed for denoting a"Temperature Curie"or a 'superconducting critical temperature', respectively. These sub-domain-specific conventions pose a significant challenge when attempting to create structured datasets across various sub-domains effectively.

Meanwhile, the advent of large language models (LLMs) has inaugurated a new technological era marked by extraordinary potential. These models not only excel in linking diverse concepts but also in engaging in sophisticated conversational reasoning [15–18]. In comparison, rule-based approaches are simpler and faster (tokens per second); however, they are time-intensive to fine-tune and have weak generalisation capabilities, as new rules should be clarified on a case-by-case basis. Small Language Models (SLMs), e.g. BERT-based models, are more specific to the task on which they are pre-trained. The data size used for pre-training LLMs is usually high enough to contain a high diversity of contexts, thus necessitating fewer examples at the fine-tuning stage than SLMs models.

LLMs offer the possibility of integrating large corpus of textual data at training, with often the ability to ingest large textual inputs at inference with a context window ranging from 4,096 to 128,000 tokens for GPT-3.5-Turbo and GPT-4-Turbo [19], respectively (at the time of this writing). Differently, BERT-based encoders are limited to only 512 tokens, whereas 1,000 tokens are about 750 English words. The size of BERT models does not allow them to sustain their contextual memory after fine-tuning. LLMs possess the capacity to be fine-tuned and retain the contextual knowledge from pre-training, which gives them an advantage in terms of generalisation to other datasets. Finally, the interaction with LLMs via prompts, i.e. tailored instructions, changes the construction paradigm of programmatic solutions, making them more accessible, flexible, and suitable to human operators. Nevertheless, the actual capabilities of LLMs in reasoning, understanding, and recognition are still constantly evolving and being evaluated.

Previous studies in Information Extraction (IE) have shown evidence of LLMs proficiency in general tasks, presenting a valuable opportunity to develop more flexible Text and Data Mining (TDM) processes. Still, they fall short in areas where specific knowledge is required [20]. In particular, LLMs are on par with SLMs in most of the discriminative tasks such as named entity recognition (NER), relation extraction (RE) and event detection (ED) in general domains [21], in history [22], and biology [23]. Other works testing chemistry capabilities found that GPT-4 understands various aspects of chemistry, including chemical compounds [24]; however, its knowledge is general and lacks methods for learning through retrieving recent literature [25].

Therefore, this study assesses LLMs' ability to comprehend, manipulate, and reason with complex information that demands substantial background knowledge, as in materials science.

The objectives of this work can be summarised with the two following questions:

- **Q1**: How effectively can LLMs extract materials science-related information?
- **Q2**: *To what extent can LLMs use reasoning to relate complex concepts?*

We first classify the fundamental components of the materials science knowledge directed towards designing novel materials with functional properties into two main entity classes: material and property expressions. Properties, e.g. a critical temperature of 4 K, are expressed using measurements of physical quantities. They exhibit a structured format, including modifiers (e.g".between"",less than"",approximately", or symbols such as">" or"∼"), values, and units, with a wide range of potential values. In contrast, material definitions are conceptually loose and often depend on the specific domain. They may require a substantial amount of accompanying text for a comprehensive description, encompassing details, e.g. compositional ratios, doping agent and ratio, synthesis protocol, process, and additional adjunct information. From a fundamental compositional standpoint, materials are defined by their chemical formula. However, in practice, authors in literature may frequently employ substantives such as commercial names, well-known terms, or crafted designations to describe samples, all of which streamline information in their research papers. Nonetheless, conveying such definitions can unambiguously be challenging.

To address Q1, we evaluate the LLM's performance on NER tasks related to materials and properties extraction. For each task, we choose a pertinent dataset and analyse the performance of each LLM.

Named Entity Recognition (NER) [26], alternatively referred to as named entity identification or entity extraction, stands as a pivotal component within information extraction. Its primary objective is to pinpoint and categorise named entities within unstructured text, assigning them to predefined categories such as individual names, organisations, geographic locations, medical codes, temporal expressions, quantities, monetary values, percentages, and more. The process of identifying entities aligns closely with sequence labelling tasks, wherein a string of text undergoes analysis, and each token within it (basic unit of text processing, typically a word or a sequence of characters that is treated as a single unit) is designated to one of the pre-established categories. For instance, these categories may include material, doping, condition, or property, among others.

We address Q2 by assessing the capability to establish connections between a predefined set of entities and extract relationships within a given context. Extracting relations between entities is a foundational undertaking in NLP. It entails discerning connections or associations among entities referenced within textual data. For instance, in biomedical research, relationship extraction might involve identifying the association between specific genes and diseases mentioned in scientific literature.

In both cases, we compare the outcomes against a baseline determined by scores (Precision, Recall and F1-score) achieved on the same datasets by either a BERT-based encoder or a rule-based algorithm we have developed in a previous work [27,28]. Our requirement for the models to be capable of generating output in a valid JSON (JavaScript Object Notation) format is part of our efforts to extract structured databases (Section 2.1.1).

The evaluation of generative models brings an additional complexity. Traditional SLM implementations for solving NER tasks are based on sequence labelling algorithms. They classify each token in the input stream with a limited number of labels, returning a sequence that fits the original input (same number of tokens and structure). Evaluating their performance against expected datasets involves a straightforward comparison of values. Soft-matching techniques can be employed to overlook minor discrepancies. However, with generative models, the output tokens may be structured in ways that significantly differ from the original input sequence. In more general scenarios, semantic models that compare the vectorised representations of two sequences can be utilised [29]. Nevertheless, when dealing with concepts like material expressions, a specialised approach is needed. As an illustration, the terms"solar cell" and"solar cells" represent identical concepts. Yet, the materials denoted by"Ca" (Calcium) and"Cr" (Chromium) are entirely distinct, highlighting a difference of just one letter between the two

examples. For this reason, we introduce a novel evaluation method for material names, which involves normalising materials to their chemical formulas before conducting a pairwise comparison of each element. This approach provides a more meaningful and context-aware assessment of the model's performance.

We summarise our contributions as follows:

- We designed and ran a benchmark for LLMs on information extraction, particularly NER of materials and properties. This contribution addresses Q1.
- We evaluated LLMs on RE on entities in the context of materials science to address Q2.
- We propose a novel approach for evaluating Information Extraction tasks applied to materials entities which leverage"formula matching" via pairwise element comparison.

2. Method

We chose three OpenAI LLM models reported with their specific names for performing API calls: GPT-3.5-Turbo (gpt-3.5-turbo-0611), GPT-4 (gpt-4), and GPT-4-Turbo (gpt-4–0611-preview). The consideration of open-source LLMs has been deferred to future work due to their limited capability to generate output in a valid JSON format (Section 2.1.1, necessitating a more in-depth investigation.

Our evaluation uses different strategies: zero-shot prompting, few-shot prompting, and fine-tuning (or instruction-learning). Few-shot prompting refers to the model's ability to adapt and perform a new task with minimal examples or prompts. In contrast, zero-shot prompting denotes the model's capability to generalise to tasks it has not been explicitly trained on, emphasising transfer learning within the language domain. Finally, fine-tuning involves adjusting the parameters of a pre-trained model on a specific task or domain using a smaller, task-specific dataset to enhance its performance for that particular application.

We selected two datasets for evaluation: MeasEval [30], a SemEval 2021 task of extracting counts, measurements, and related context from scientific documents and SuperMat, an annotated and linked dataset of research papers on superconductors [31]. SuperMat contains both materials and properties and, for copyright reasons, is not publicly distributed. This reduces the risk that its annotations had been used during the pre-training of any of the LLMs.

Baseline scores were established using a SciBERT-based [32] encoder and RE rule-based algorithm [27] for material-related extractions. Grobid-quantities [28] served as the baseline for NER on properties extraction evaluated against MeasEval.

Evaluation scores, encompassing Precision, $TP/(TP + FP)$, Recall, $TP/(TP + FN)$), and F1-score, $2\,\text{Precision} \times \text{Recall}/(\text{Precision} + \text{Recall})$, were derived from pairwise comparisons between predicted and expected entities, where TP, FP and FN are the true positive, false positive and false negative instances, respectively. Precision gauges accuracy, recall assesses information capture, and F1-Score is their harmonic mean.

The evaluations condense average F1 scores and their standard deviation over three extraction runs. The raw tables with all detailed scores are provided in Appendix A.

2.1. Named entities recognition

The NER task consists of identifying relevant entities: materials, expressed through a multitude of expressions [31], or properties, expressed as measurements of physical quantities [28].

We calculated the evaluation scores using four different matching approaches. However, we will present only the most relevant to the task (leaving the complete tables[1] in Appendix A):

- **strict**: Exact matching
- **soft** Matching using Ratcliff/Obershelp [33] with a threshold at 0.9[2]
- **Sentence BERT** Comparison using semantic similarity of sequences using Sentence BERT with a cross-encoder [29], applying a threshold set at 0.9[2]
- **formula matching** Our novel method compares material expressions via formula normalisation and element-by-element exact matching.

Prompts for interacting with LLMs are defined by two components: system and user prompts. The system prompt is the initial instruction guiding the model's output generation, defining the task or information sought. In contrast, the user prompt is the user's input, specifying their request and shaping the model's response.

The system prompt below was fixed across all tasks. It was specifically crafted to prevent the creation of non-existing facts and favour standardised answers (e.g".I don't know","None", etc.) in case of inability to respond.

The users' prompts for NER with zero-shot prompting were described including the definitions and examples from the SuperMat[3] and MeasEval[4] annotations guidelines, respectively.

Below are the user prompt templates used for both materials and properties extraction:

Then, we applied a few-shot prompting technique by incorporating in the users' prompt template above a set of suggestions extracted from the text (see Listing 3 below) using the respective SLMs based on the fine-tuned SciBERT-encoder for materials and properties,

Listing 1 Generic system prompt common to all requests.

```
Use the following pieces of context to answer the user's question.
If you don't know the answer, just say that you don't know, don't try to
    ↪ make up an answer.
----------------
{text}
```

i.e. grobid-superconductors [27] and grobid-quantities [28], respectively. Also, as these suggestions originate from another model, they may not be entirely accurate; hence, we emphasised in the prompts that they only serve as examples or hints that the LLMs may ignore.

2.1.1. Output format

For all tasks, we required the output to be formatted using a valid JSON document. We justify this decision for three main reasons: a) The responses need to be machine-readable so that the de-serialisation from JSON to objects in many programming languages becomes a trivial operation (e.g. Python, JavaScript). b) The JSON schema can be defined through a documented format regardless of the programming language or platform. Finally, c) the JSON format is an open standard that can be used by anyone and does not require reinventing the wheel by re-implementing any transformation steps from scratch.

Listing 2 User prompt designed for extracting materials and properties. The entity descriptions are separated by dashed lines ("——").

```
What are the superconductor materials mentioned in the text?
Only provide the mention of the materials. Avoid repetition.

The material can be expressed as follows:
- chemical formula with variables not substituted, like La(1-x)Fe(x),
- chemical formula with substitution variables like Zr 5 X 3 (X = Sb, Pb
    ↪ , Sn, Ge, Si and Al)
- with complete or partial abbreviations like (TMTSF) 2 PF 6,
- doping rates are represented as variables (x, y or other letters)
    ↪ appearing in the material names. These values can be used to
    ↪ complement the material variables (e.g. LaFexO1-x).
- doping rates as percentages, like 4% Hdoped sample or 14% Cu doped
    ↪ sample
- material chemical form with no variables e.g. LaFe03NaCl2 where the
    ↪ doping rates are included in the name
- chemical substitution or replacements, like (A is a random variable,
    ↪ can be any symbol): A = Ni, Cu, A = Ni, Ni substituted (which
    ↪ means A = Ni)
- chemical substitution with doping ratio, like (A is a random variable,
    ↪  can be any symbol): A = Ni and x = 0.2

If you don't know the answer, just say you don't know, don't try to make
    ↪  up an answer.

-----

Quantity is either a Count, consisting of a value, or a Measurement,
consisting of a value and usually a unit. A Quantity can additionally
    ↪ include optional Modifiers like tolerances.
Include relevant text that indicates the application of a modifier, such
    ↪  as "between", "less than", "approximately",
or symbols such as ">" or "~" if they are contiguous with the span.
    ↪ Ignore them if they are separated by additional text.

Example: "The soda can's volume was 355 ml", the quantity is "355 ml".

Extract all the Quantities in the text.
```

Listing 3 Few-shot prompting modified prompt template.

```
[...]
Here are some examples appearing in the text: {hints}
[...]
```

The JSON output was obtained by adding formatting instructions in the user's prompt based on the expected output data model, for which different concepts were described differently (e.g. properties are described as a value and an optional unit). We used the implementation provided by the LangChain library[5] of which one example is illustrated as follows.

Listing 4 Example of formatting instruction to a valid JSON format.

```
The output should be formatted as a JSON instance that conforms to the
    ↪ JSON schema below.

As an example, for the schema {"properties": {"foo": {"title": "Foo", "
    ↪ description": "a list of strings", "type": "array", "items": {"
    ↪ type": "string"}}}, "required": ["foo"]}
the object {"foo": ["bar", "baz"]} is a well-formatted instance of the
    ↪ schema. The object {"properties": {"foo": ["bar", "baz"]}} is not
    ↪  well-formatted.

Here is the output schema:
```
{"properties": {"material": {"title": "Material", "description": "
    ↪ Material or sample name, chemical formula, acronym. Include
    ↪ everything that describes the material.", "type": "string"}, "
    ↪ material_extra_info": {"title": "Material Extra Info", "
    ↪ description": "Additional information about the material", "type
    ↪ ": "string"}}, "required": ["material"]}
```
```

2.1.2. Formula matching

Matching materials poses challenges with generative models, while encoder and sequence labelling models maintain unchanged the output from the input sequences. Therefore, evaluating generative models can be complex due to potentially divergent yet semantically equivalent output sequences. Previous works [34] resort to manual evaluation due to these challenges. Notably, as of the time of writing, no specialised approach tailored for material expressions existed. Utilising Sentence BERT, trained on general text, does not ensure accurate material embeddings, raising concerns about the meaningfulness of final matches. To address issues arising from variable sets and to enhance evaluation precision, we propose a novel method named *formula_matching*, involving element-by-element pairwise comparisons on normalised formulas for extracted material denominations.

This approach extends strict matching and is activated only when the two input strings differ. In such instances, as depicted in Figure 1, the material expressions slated for comparison undergo normalisation to their formulas using a material parser developed in our prior work [27] (Figure 1 top). The material parser is adept at handling noisy material expressions and strives to parse them effectively. The anticipated output includes a structured representation with the chemical formula presented as a raw string and a dictionary detailing elements and their respective amounts. Subsequently, these structures are compared element by element, as depicted in Figure 1 bottom. The summarised evaluation scores described in Section 3.4 are calculated using the formula matching.

Evaluation and discussion of this method are detailed in Section 3.2.

2.2. Relation extraction

The baseline is established by a rule-based algorithm from our previous work [27], which was evaluated with SuperMat and for which we report the aggregated result in Section 2.2.

The prompts are designed by providing a list of entities and requesting the LLM to group them based on their relation. Unlike the NER task, the LLM is expected to reuse information passed in the prompt to compose the response: non-matching information is considered incorrect. The summarised scores in Section 3.5 are obtained with strict matching.

The previous considerations remain relevant for both system and user prompts, with the task description reiterated in each prompt.

We add specific rules to avoid creating invalid groups of relations and to ignore responses containing entities not supplied in the user prompt or empty relation blocks.

The prompt for few-shot prompting was assembled by injecting three examples listed between the dashed lines ("———") in the zero-shot prompt:

2.2.1. Shuffled vs non-shuffled evaluation

The list of entities supplied to the Language Model (LLM) might be derived based on their order of

Listing 5 System prompt for RE modified by emphasising the tasks.

```
You are a useful assistant, who knows about materials science, physics,
    ↪ chemistry and engineering.
You will be asked to compute relation extraction given a text and lists
    ↪ of entities.
If you are not sure, don't try to make up your answer, just answer "None
    ↪ ".
```

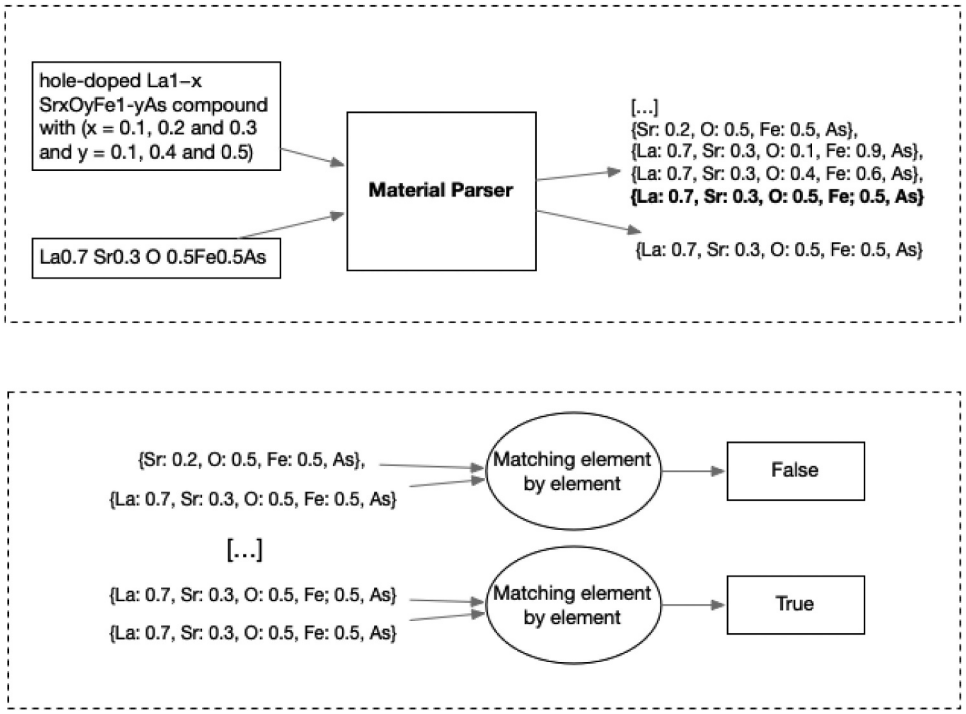


Figure 1. Two materials that appear to have a very different composition are, in reality, overlapping. (Top) Summary of the Material Parser. More information is available in [27]. (Bottom) the pairwise comparison of each chemical formula is performed element-by-element.

appearance, creating a scenario where a model generating relations sequentially may achieve an inflated score that does not accurately reflect its relational inference capabilities. To address this, we evaluate each model for RE using two strategies: a *non-shuffled evaluation*, where entities are presented in the order they appear in the original document, and a *shuffled evaluation*, where entities are randomly rearranged before being introduced to the prompt.

2.3. Consideration about the fine-tuning

We fine-tuned the GPT-3.5-Turbo model using the OpenAI platform, which ingested training and testing data and generated a new model in a few hours. At the time of writing this article, the fine-tuning of GPT-4 and GPT-4-Turbo is not available. All fine-tuned models were trained using the default parameters selected by the OpenAI platform.

Table 1 illustrates the dimension of each dataset. The fine-tuned model for properties extraction was trained using the"grobid-quantities dataset" [28] because MeasEval did not contain enough examples for a consistent and unbiased evaluation.

The primary challenge encountered when employing a fine-tuned model was to achieve a valid, machine-readable JSON format. Therefore, we formatted the training data with an expected output in valid JSON format. However, the obtained fine-tuned model struggled to produce valid JSON in its output, leading us to hypothesise that this limitation might be attributed to a shortage of training examples. To address this, we modified our training data expected output from JSON to a pseudo format structured with spaces and break-lines, facilitating simpler handling by the model. The subsequent example illustrates the expected output for a RE task:

We followed the same approach for fine-tuning the model for the NER task:

Using this technique, we could fine-tune a model and shape its behaviour to answer

Listing 6 caption=Few-shot prompting for extracting relations from lists of entities].

```
Given a text between triple quotes and a list of entities, find the
    ↪ relations between entities of different classes:
"""
{text}
"""

{entities}

Use the following examples separated by "--------" to learn the task:
--------
text 1: The researchers of Mg have discovered that MgB2 and MgB3 are
    ↪ superconducting at 29-31 K at ambient pressure.

entities 1:
 materials: MgB2, Mg, MgB3
 tcs: 29-31 K
 pressure: ambient pressure

Result 1:
 material: MgB2,
 tc: 29-31K,
 pressure: ambient pressure:

 material: MgB3,
 tc: 29-31K,
 pressure: ambient pressure:

--------
Text 2: We are studying the material La 3 A 2 Ge 2 (A = Ir, Rh). The
    ↪ critical temperature T C = 4.7 K discovered for La 3 Ir 2 Ge 2 in
    ↪  this work is by about 1.2 K higher than that found for La 3 Rh 2
    ↪  Ge 2.

entities 2:
 materials: La 3 A 2 Ge 2 (A = Ir, Rh), La 3 Ir 2 Ge 2, La 3 Rh 2 Ge 2
 tcs: 4.7 K, 1.2 K

Result 2:
 material: La 3 Ir 2 Ge 2
 tc: 4.7 K

--------
Text 3: The experimental discovery of the high-temperature
    ↪ superconducting state in the compressed hydrogen and sulfur
    ↪ systems H2S (TC = 150 K for p = 150 GPa) and H3S (TC = 203 K for
    ↪ p = 150 GPa)

entities 3:
 materials: H2S, H3S
 tcs: 150 K, 203 K
 pressures: 150 GPa, 150 GPa

Result 3:
 material: H2S,
 tc: 4.7 K,
 pressure: 150 GPa

 material: H3S,
 tc: 150 K,
 pressure: 150 GPa
--------

Apply strictly the following rules:
    - if material is not specified, ignore the relation block,
    - if tc is not specified in absolute values, ignore the relation
        ↪ block
```

Table 1. Datasets and support information for fine-tuning GPT-3.5-Turbo. For each task, the data was divided into 70/30 partitions for training and testing, respectively. The testing dataset is different from the evaluation dataset.

Task	Preparation strategy	Dataset	# Training	# Test
NER	N/A	SuperMat	1639	703
NER	N/A	grobid-quantities dataset	485	208
RE	FT.base/FT.document	SuperMat	344	148
RE	FT.augmented	SuperMat	695	299

Listing 7 Example format of the expected answer for the RE task.

```
material: mat1, tc: 22K,
material: mat2, tc: 24K, pressure: 2GPa
```

Listing 8 Example format of the expected answer for the NER task.

```
materials:
 - material1
 - material2
 - material3
```

conversationally. Then, we used the GPT-3.5-Turbo base model to transform the response into JSON format.

To fine-tune the model for the RE task, we introduced the sorting variability in the entity lists provided in the prompt (Section 2.2). This approach does not modify the size of the data set and reduces the possibility that the model learns to aggregate entities in the order they appear in the document. This is the default approach we define as"FT.base" compared to others. In Section 3.5.1, we discuss the impact of two additional strategies for preparing the fine-tuning data. First",FT.document_order" keeps the lists of entities as they appear in the document. For example, the made-up sentence *"The two materials MgB2 and MgB3 showed Tc of 39K and 40K, respectively"* will lead to two lists of entities"MgB2, MgB3" and"39K and 40K" which could be assigned in order (MgB2, 39K) and (MgB3, 40K). Intuitively, this leads to poor performance, as we see when evaluating with shuffling conditions (Section 3.5). The second strategy",FT.augmented", is to augment the size of the dataset, generating multiple training records with a further shuffled entity list for each example in"FT.base". The data used with this strategy is roughly double that of"FT.base" (Table 1). We expect this strategy to obtain similar or better results than"FT.base".

3. Results and discussions

In this section, we present and discuss the formula matching and the aggregated results of our evaluations for the LLMs. The completed raw results are available in the Appendix A.

3.1. Limitation of this study

In this paper, we aim to estimate how well LLMs work in tasks related to materials science. Due to the lack of clean datasets covering the entire materials science domain, we used a dataset that focuses on superconductor material. While our goal is to propose a methodology, we are aware that our results need to be verified empirically in other materials science sub-domains in future works. The following intuitions support our hypothesis: for material NER, we expect that the forms on which materials are presented in other domains would have similar expressions to the ones used in superconductor research, considering that chemical formulas, sample names, and commercial names would unlikely be very different between domains. Furthermore, the properties, expressed as measurement and physical quantities, are common to all domains; although the statistical distribution could be different, we don't expect dramatic differences within materials science. On the other hand, RE tasks surely require more datasets that focus both on different domains and different flavours of the same task. As an example, the MatSCIRe [35] dataset, which covers battery-related research, proposes a structure that challenges the relation extraction only between two entities (binary extraction) with the addition of the type of relation which could be inferred by the properties being extracted. In conclusion, we will remand the generalisation for further work.

3.2. Formula matching

We evaluated the formula matching to measure two main pieces of information: the gain in the F1-score, and the correctness, as the number of invalid new matches, of the gain. We compared the formula matching with the strict matching because a) it is simple to reproduce and understand visually, and b) the formula matching is built on top of strict matching. We would have more difficulties explaining matches provided by soft matching or SentenceBERT.

We examined the GPT-3.5-Turbo NER extraction (discussed in Section 3.5). 107 out of the 1402 expected records matched correctly using strict matching (P: 22.5%, R: 13.64%, F1: 17.01%). Applying formula

matching on the mismatching records, we obtained an additional 176 matches (P: 61.12%, R: 36.00%, F1: 45.31%), for a total gain in F1-score of 28.3 (+266%). For the new 176 records that the formula matching was identifying, we manually examined each pair finding 5 incorrect matches, which corresponds to an error rate of 2.5%.

Most of the mismatches in the strict matching caught up by the formula matching were due to missing adjoined information. The LLMs were not able to include information about doping or shape in the response (e.g. hole-doped La 2-x Sr x CuO 4 was not matching with La 2-x Sr x CuO 4). In other cases, the formula was different by formatting, like: Nd 2-x Ce x CuO 4 and La 2-x Sr x CuO 4. However, the more interesting cases were provided by element or amount substitutions such as: electron-doped infinite-layer superconductors Sr 0.9 La 0.1 Cu 1-x R x O 2 where R = Zn and Ni which was matched Sr0.9La0.1Cu1-xNixO2, or Eu 1-x K x Fe 2 As 2 samples with x = 0.35, 0.45 and 0.5 and Eu 0.5 K 0.5 Fe 2 As 2'. These two cases were particularly complicated to match because they required a deeper understanding of the formula structure.

Among the errors of the formula matching, all of them were provided by the formula which was not correctly parsed, for example in one complicated case with the substrate information: (1-x/2)La 2 O 3/xSrCO 3/CuO in molar ratio with x = 0.063, 0.07, 0.09, 0.10, 0.111 and 0.125 which was incorrectly matched with the general La2O3.

3.3. NER on properties extraction

The property extraction assessment was performed using the MeasEval dataset, with the baseline established by Grobid volumes, achieving an approximately 85% score using a holdout dataset created in conjunction with the application. At the time of writing, the evaluation of grobid-quantities [28] (version 0.7.3[6]) against MeasEval yielded a score of around 59% F1-score. This disparity was anticipated, given the slightly divergent annotation strategies employed by the MeasEval developers compared to those used in developing grobid-quantities (e.g. considerations such as approximate values and other proximity expressions were not considered).

Unexpectedly, none of the models outperformed grobid-quantities in zero-shot prompting, as depicted in Figure 2. This outcome is surprising considering that a) the expression of properties lacks a specific domain constraint (aside from potential variations in frequency distribution), and b) measurements of physical quantities are likely prevalent in the extensive text corpus used to pre-train the OpenAI models.

In the realm of few-shot prompting (Figure 2), a marginal improvement was observed only for GPT-4 and GPT-4-Turbo, resulting in an F1-score gain ranging around 2%. However, this improvement is not significant. We theorise that the hints provided to the LLMs may introduce bias. When these hints are incorrect or incomplete, the LLMs struggle to guide the generation effectively, impacting the quality of the output results. Significantly, the fine-tuned model (Figure 2) shows a slight enhancement compared to zero-shot, few-shot, and the baseline. Interestingly, in this specific instance where both the baseline and fine-tuned models are trained and evaluated on the same data, the LLM demonstrates an approximate 3% increase in the F1-score.

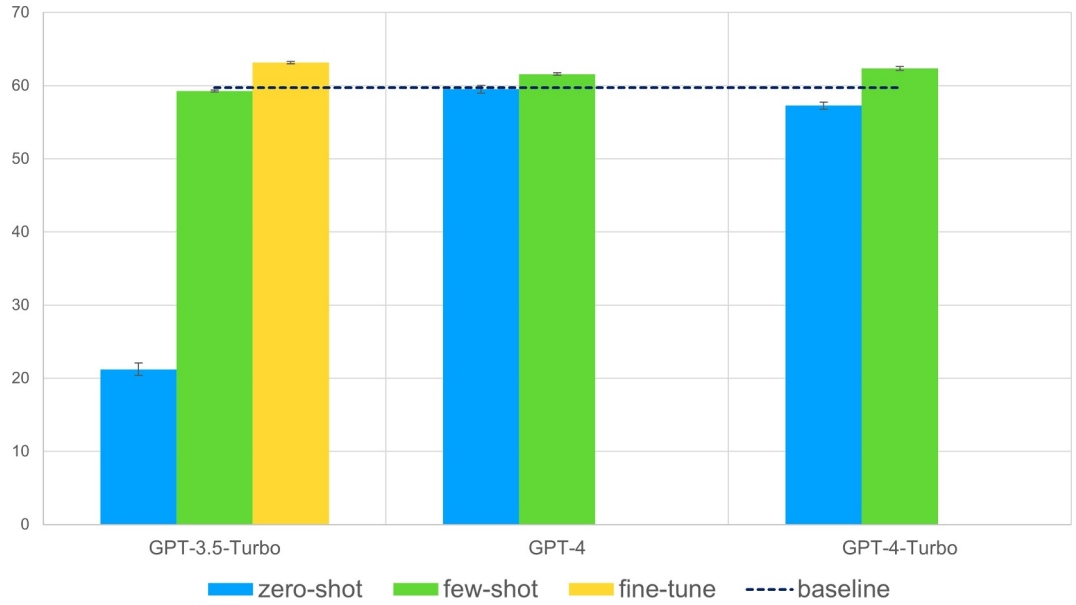


Figure 2. Comparison scores for properties extraction using NER. The scores are the aggregations of the micro average F1 scores and are calculated using soft matching with a threshold of 0.9 similarity. The error bars are calculated over the standard deviation of three independent runs.

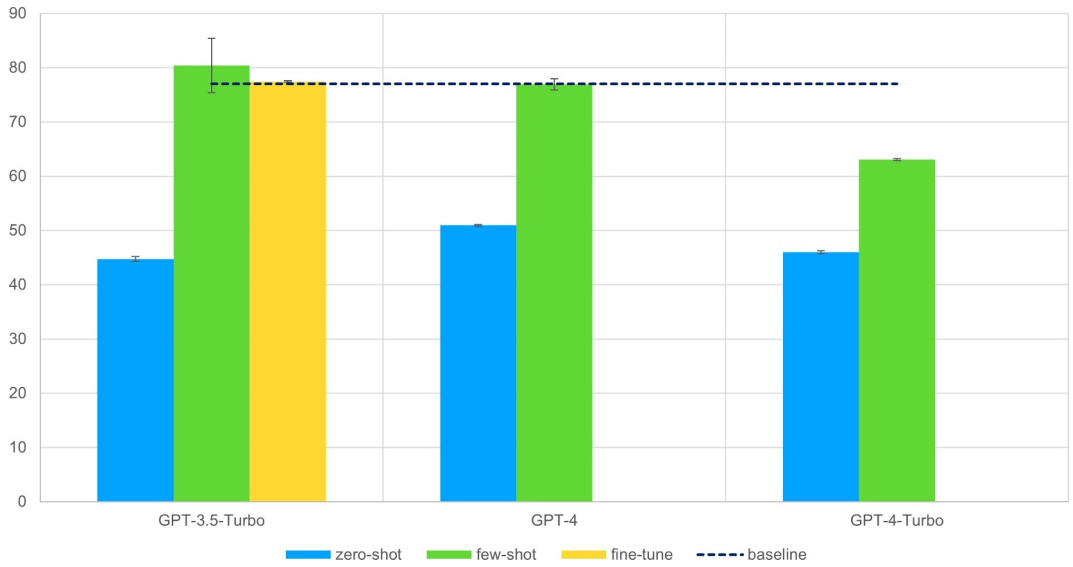


Figure 3. Comparison scores for material extraction using NER. The metrics are the aggregations of the micro average F1-scores, calculated using formula matching. The error bars are calculated over the standard deviation of three independent runs.

3.4. NER on materials expressions extraction

The evaluation of material expressions extraction was performed using the partition of the SuperMat [31] dataset dedicated to validation, consisting of 32 articles.

In zero-shot prompting (Figure 3), both GPT-4 and GPT-4-Turbo achieved comparable F1-scores, hovering around 50%. Notably, all LLMs scored at least 10% lower than the baseline [27]. This disparity is expected, given that material expressions may involve extensive sequences and encompass multiple pieces of information not easily conveyed in the prompt. Few-shot prompting (Figure 3) yielded improved results, with GPT-3.5-Turbo and GPT-4 slightly surpassing the baseline. The introduction of hints in the prompt indeed enhances performance, but, as previously discussed, it appears to strongly influence the LLMs, not able to mitigate the impact of invalid hints that may be provided. Equally unexpected, fine-tuning did not outperform few-shot prompting. This outcome suggests that the additional training did not significantly enhance the LLMs' ability to handle material expressions.

3.5. Relation extraction

The evaluation of RE utilised the complete SuperMat dataset, with the results illustrated in Figure 4, comparing the effects of shuffling across different models.

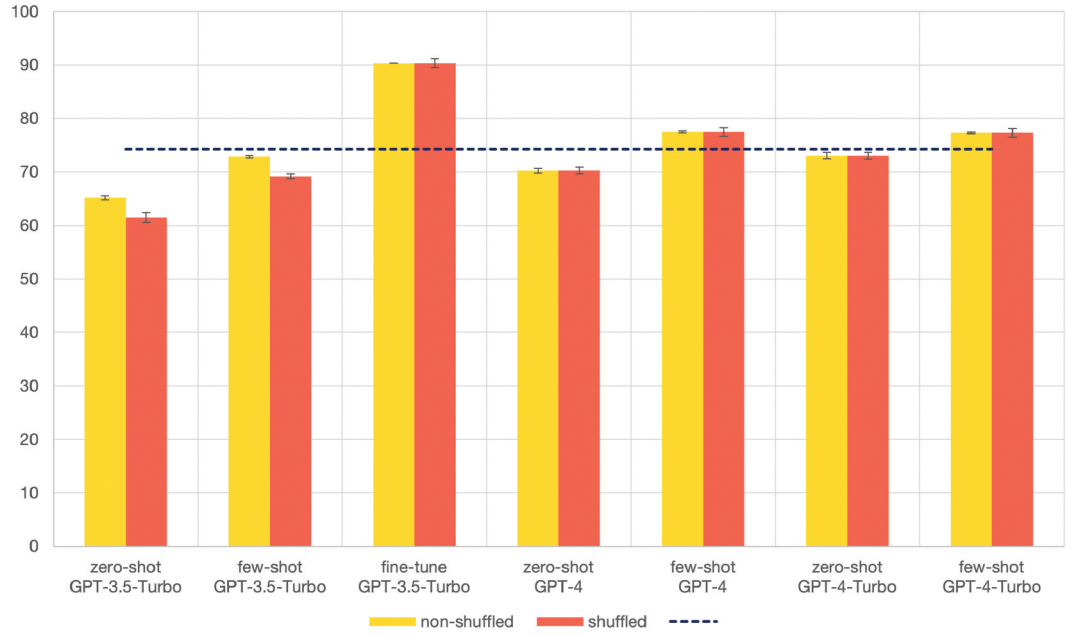


Figure 4. Comparison of the scores of the shuffled extraction using zero-shot prompting, few-shot prompting and the fine-tuned model for RE on materials and properties. The metrics are the aggregated micro average F1-scores calculated using strict matching. The error bars are calculated over the standard deviation of three independent runs.

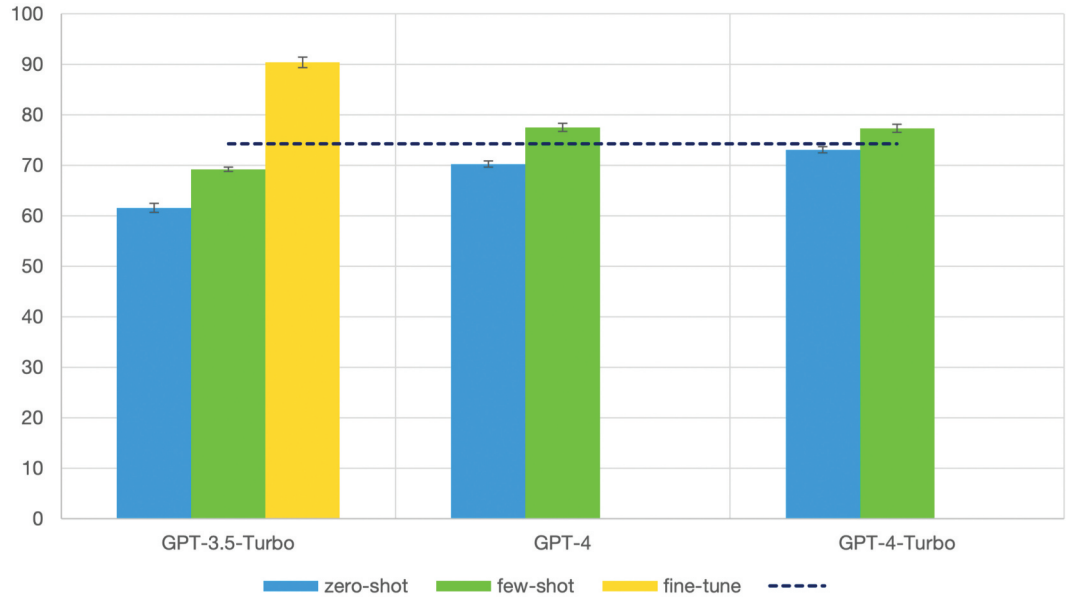


Figure 5. Overview evaluation on shuffling the provided entities in RE on materials and properties. The metrics are the aggregated micro average F1-scores calculated using strict matching. The error bars are calculated over the standard deviation of three independent runs.

GPT-3.5-Turbo zero-shot and few-shot prompting demonstrate a significant difference between shuffled and non-shuffled evaluation (Section 2.2.1), suggesting a sequential connection of entities without specific contextual reasoning. Notably, the fine-tuned GPT-3.5-Turbo model outperforms the baseline by approximately 15% F1-score and does not show relevant differences when the evaluation is performed under shuffling conditions.

Figure 5 specifically highlights the shuffled version of each model and extraction type. Except for GPT-3.5-Turbo, few-shot prompting shows an improvement compared to zero-shot prompting, achieved by incorporating additional examples in each prompt. GPT-4 and GPT-4-Turbo also exhibit stable results under shuffling conditions, achieving an F1-score of around 15–18% lower than fine-tuned GPT-3.5-Turbo.

3.5.1. Data variability for fine-tuning

In Section 2.3, we describe two additional ways to prepare the data for fine-tuning. As illustrated in Figure 6, the GPT-3.5-Turbo model fine-tuned with the strategy"FT.document_order" showed an inability to generalise when evaluated under shuffling conditions, where the model loses around 30% in F1-score.

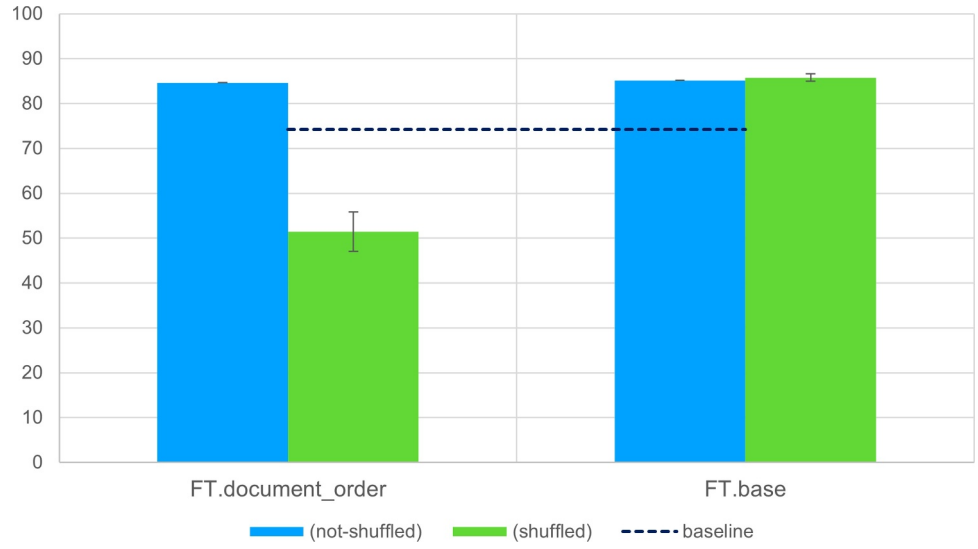


Figure 6. Evaluation of the impact of data variability in fine-tuning GPT-3.5-Turbo. The metrics are the aggregated micro average F1-scores calculated using strict matching. The model"FT.Document_order" was fine-tuned with the original data, where entities were taken of appearance. In"FT.Base", our default strategy, the entities provided to the prompt were scrambled. The error bars are calculated over the standard deviation of three independent runs.

This suggests that adding entropy (for example, by shuffling the data) should be performed as a best practice, which could result in models with larger reasoning capabilities.

When we increased the size of the dataset used in fine-tuning to almost double (Table 1), the resulting model did not improve compared to the FT.base. These results confirm that in fine-tuning, size does not matter, while data variability and quality do.

4. Code and data availability

This work is available at https://github.com/lfoppiano/MatSci-LumEn. The repository contains the scripts and the data used for extraction and evaluation. The code of the material parser used in the formula matching is available at https://github.com/lfoppiano/material-parsers, and the service API is accessible at https://lfoppiano-material-parsers.hf.space.

5. Conclusion

In this study, we have proposed an evaluation framework for estimating how well LLMs perform compared with SLMs and rule-based tasks related to materials science by focusing on sub-domains such as superconductor research. The findings obtained from our work provide initial guidance applicable to other materials science sub-domains in future research.

To evaluate material extraction comparison, we proposed a novel method to parse and match formula elements by elements through an aggregated parser for materials. This new method provides a more realistic F1 score. Compared with strict matching, we obtained a gain in F1-score from 17% to 45% for GPT3.5-Turbo NER at the price of a minimal error rate (2%).

We then evaluated LLMs on two tasks: NER for materials and properties and RE for linking them. LLMs underperform significantly on NER tasks than SLMs in material and property extraction (Q1). This finding is particularly surprising considering properties since these expressions are not confined to a specific domain.

In material extraction, GPT-3.5-Turbo with fine-tuning failed to outperform the baseline, and the same holds for any model with few-shot prompting. For property extraction, GPT-4 and GPT-4-Turbo with zero-shot prompting perform on par with the baseline. GPT-3.5-Turbo with few-shot and fine-tuning, on the other hand, outperforms the baseline by a marginal increase in points. Our results suggest that, for material expressions, small specialised models remain the most accurate choice.

The scenario improves for RE (Q2). With two examples, few-shot prompting demonstrates a significant improvement over the baseline. GPT-4-Turbo exhibits enhanced reasoning capabilities compared to GPT-4 and GPT-3.5-Turbo. GPT-3.5-Turbo performs poorly in both zero-shot and few-shot prompting, showing a substantial score decrease when entities are shuffled, which aligns with previous observations. Nevertheless, fine-tuning yields scores superior to the baseline and other models, showing stability when comparing shuffled and unshuffled evaluations.

In conclusion, to answer Q2, GPT-4 and GPT-4-Turbo showcase effective reasoning capabilities for accurately relating concepts and extracting relations without fine-tuning. However, fine-tuning GPT-3.5-Turbo out yields the best results with a relatively small dataset. GPT-4-Turbo, which costs one-third of GPT-4, remains a robust choice given its reasoning capabilities. However, for Q1, for extracting complex entities such as materials, we find that training small specialised models remains a more effective approach.

Notes

1. The calculation of micro average provides a measure independent of the distribution of the extracted entities over the different documents.
2. The threshold is fixed to a value yielding more than 90% similarity.
3. https://supermat.readthedocs.io
4. https://github.com/harperco/MeasEval/tree/main/annotationGuidelines#basic-annotation-set
5. https://github.com/langchain-ai/langchain
6. https://github.com/lfoppiano/grobid-quantities/releases/tag/v0.7.3

Acknowledgements

Our warmest thanks to Patrice Lopez for his continuous support and inspiration with ideas, suggestions, and fruitful discussions.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was partially supported by the MEXT Programme: Data Creation and Utilisation-Type Material Research and Development Project (Digital Transformation Initiative Centre for Magnetic Materials) Grant Number [JPMXP1122715503].

Author contributions

LF developed the scripts for extraction and evaluation and wrote the manuscript. GL supported the LLM evaluation and implementation and financed access to the OpenAI API. GL, TA, and MI reviewed the article. MI supervised the process and provided the budget.

ORCID

Luca Foppiano http://orcid.org/0000-0002-6114-6164
Guillaume Lambard http://orcid.org/0000-0003-0275-4079
Masashi Ishii http://orcid.org/0000-0003-0357-2832

References

[1] Xu P, Ji X, Li M, et al. Small data machine learning in materials science. Npj Comput Mater. 2023 Mar;9 (1):42. doi: 10.1038/s41524-023-01000-z
[2] Boyd PG, Chidambaram A, Garca-Dez E, et al. Data-driven design of metal–organic frameworks for wet flue gas co2 capture. Nature. 2019;576 (7786):253–256. doi: 10.1038/s41586-019-1798-7
[3] Rao Z, Tung P-Y, Xie R, et al. Machine learning–enabled high-entropy alloy discovery. Science. 2022;378(6615):78–85. doi: 10.1126/science.abo4940
[4] Zakutayev A, Wunder N, Schwarting M, et al. An open experimental database for exploring inorganic materials. Sci Data. 2018;5(1):1–12. doi: 10.1038/sdata.2018.53
[5] Doan Huan T, Mannodi-Kanakkithodi A, Kim C, et al. A polymer dataset for accelerated property prediction and design. Sci Data. 2016;3(1):1–10. doi: 10.1038/sdata.2016.12
[6] Pyzer-Knapp EO, Pitera JW, Staar PW, et al. Accelerating materials discovery using artificial intelligence, high performance computing and robotics. Npj Comput Mater. 2022;8(1):84. doi: 10.1038/s41524-022-00765-z
[7] Huber N, Kalidindi SR, Klusemann B, et al. Machine learning and data mining in materials science. Front Mater. 2020;7:51. doi: 10.3389/fmats.2020.00051
[8] Park G, Pouchard L. Advances in scientific literature mining for interpreting materials characterization. Mach Learn. 2021;2(4):045007. doi: 10.1088/2632-2153/abf751
[9] Chittam S, Gokaraju B, Xu Z, et al. Big data mining and classification of intelligent material science data using machine learning. Appl Sci. 2021;11(18):2021. doi: 10.3390/app11188596
[10] Ma B, Wei X, Liu C, et al. Data augmentation in microscopic images for material data mining. Npj Comput Mater. 2020;6(1):125. doi: 10.1038/s41524-020-00392-6
[11] Parinov IA. Microstructure and properties of high-temperature superconductors. Springer Science & Business Media; 2013.
[12] Hosono H, Tanabe K, Takayama-Muromachi E, et al. Exploration of new superconductors and functional materials, and fabrication of superconducting tapes and wires of iron pnictides. Sci Technol Adv Mater. 2015;16(3):033503. doi: 10.1088/1468-6996/16/3/033503
[13] Mydeen K, Jesche A, Meier-Kirchner K, et al. Electron doping of the iron-arsenide superconductor cefeaso controlled by hydrostatic pressure. Phys Rev Lett. 2020;125(20):207001. doi: 10.1103/PhysRevLett.125.207001
[14] Bardeen J, Cooper LN, Robert Schrieffer J. Theory of superconductivity. Phys Rev. 1957;108(5):1175. doi: 10.1103/PhysRev.108.1175
[15] Zhang C, Zhang C, Li C, et al. One small step for generative ai, one giant leap for agi: a complete survey on chatgpt in aigc era. arXiv preprint arXiv:2304.06488. 2023.
[16] Yao S, Yu D, Zhao J, et al. Tree of thoughts: Deliberate problem solving with large language models. arXiv preprint arXiv:2305.10601. 2023.
[17] Valmeekam K, Marquez M, Sreedharan S, et al. On the planning abilities of large language models–a critical investigation. arXiv preprint arXiv:2305.15771. 2023.
[18] Sun S, Liu Y, Wang S, et al. Pearl: Prompting large language models to plan and execute actions over long documents. arXiv preprint arXiv:2305.14564. 2023.
[19] OpenAI. Models. 2024 [cited 2024 Jan 4]. Available from: https://platform.openai.com/docs/models
[20] Kocoń J, Cichecki I, Kaszyca O, et al. ChatGPT: Jack of all trades, master of none. Inf Fusion. 2023 Nov;99:101861. doi: 10.1016/j.inffus.2023.101861
[21] Ma Y, Cao Y, Hong Y, et al. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! arXiv preprint arXiv:2303.08559. 2023.
[22] González-Gallardo C-E, Boros E, Girdhar N, et al. Yes but. can chatgpt identify entities in historical documents? arXiv preprint arXiv:2303.17322. 2023.
[23] Moradi M, Blagec K, Haberl F, et al. Gpt-3 models are poor few-shot learners in the biomedical domain. arXiv preprint arXiv:2109.02555. 2021.
[24] Hatakeyama-Sato K, Yamane N, Igarashi Y, et al. Prompt engineering of gpt-4 for chemical research: what can/cannot be done? Sci Technol Adv Mater. 2023;3(1):2260300. doi: 10.1080/27660400.2023.2260300
[25] Hatakeyama-Sato K, Watanabe S, Yamane N, et al. Using gpt-4 in parameter selection of polymer informatics: improving predictive accuracy amidst data scarcity and 'ugly duckling'dilemma. Digital Discov. 2023;2(5):1548–1557. doi: 10.1039/D3DD00138E
[26] Nadeau D, Sekine S. A survey of named entity recognition and classification. Lingvisticae Investigationes. 2007;30(1):3–26. doi: 10.1075/li.30.1.03nad
[27] Foppiano L, Castro P, Suarez P, et al. Automatic extraction of materials and properties from superconductors scientific literature. Sci Technol Adv Mater. 2023;3(1). doi: 10.1080/27660400.2022.2153633
[28] Foppiano L, Romary L, Ishii M, et al. Automatic identification and normalisation of physical measurements in scientific literature. In: Proceedings of the ACM Symposium on Document Engineering 2019, DocEng '19. New York, NY, USA: Association for Computing Machinery; 2019.
[29] Reimers N, Gurevych I. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In: Inui K, Jiang J, Ng V Wan X, editors. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th

International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong,China: Association for Computational Linguistics; 2019 Nov. p. 3982–3992.

[30] Harper C, Cox J, Kohler C, et al. SemEval-2021 task 8: MeasEval – extracting counts and measurements and their related contexts. In: Palmer A, Schneider N, Schluter N, Emerson G, Herbelot A Zhu X, editors. Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021). Online: Association for Computational Linguistics; 2021 Aug. p. 306–316.

[31] Foppiano L, Dieb T, Suzuki A, et al. Supermat: construction of a linked annotated dataset from superconductors-related publications. Sci Technol Adv Mater. 2021;1(1):34–44. doi: 10.1080/27660400.2021.1918396

[32] Beltagy I, Lo K, and Cohan A. SciBERT: A pretrained language model for scientific text. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics; 2019 Nov. p. 3615–3620.

[33] Ratcliff John W. Pattern matching: the gestalt approach. 1988 [cited 2024 Jan 4]. Available from: https://www.drdobbs.com/database/pattern-matching-the-gestalt-approach/184407970?pgno=5

[34] Taylor R, Kardas M, Cucurull G, et al. Galactica: A large language model for science. arXiv preprint arXiv:2211.09085. 2022.

[35] Mullick A, Akash Ghosh GSC, Ghui S, et al. Matscire: Leveraging pointer networks to automate entity and relation extraction for material science knowledge-base construction. Comput Mater Sci. 2024;233:112659. doi: 10.1016/j.commatsci.2023.112659