Digital Discovery

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal | View Issue

Cite this: *Digital Discovery*, 2026, **5**, 698

Optimizing data extraction from materials science literature: a study of tools using large language models

Wenkai Ning, [a] Musen Li,*[abc] Jeffrey R. Reimers *[ac] and Rika Kobayashi *[d]

Large Language Models (LLMs) are increasingly utilized for large-scale extraction and organization of unstructured data owing to their exceptional Natural Language Processing (NLP) capabilities. Empowering materials design, vast amounts of data from experiments and simulations are scattered across numerous scientific publications, but high-quality experimental databases are scarce. This study considers the effectiveness and practicality of five representative AI tools (ChemDataExtractor, BERT-PSIE, ChatExtract, LangChain, and Kimi) to extract bandgaps from 200 randomly selected materials science publications in two presentations (arXiv and publisher versions), comparing the results to those obtained by human processing. Although the integrity of data extraction has not met expectations, encouraging results have been achieved in terms of precision and the ability to eliminate irrelevant papers from human consideration. Our analysis highlights both the strengths and limitations of these tools, offering insights into improving future data extraction techniques for enhanced scientific discovery and innovation. In conjunction with recent research, we provide guidance on feasible improvements for future data extraction methodologies, helping to bridge the gap between unstructured scientific data and structured, actionable databases.

Received 30th October 2025
Accepted 9th December 2025

DOI: 10.1039/d5dd00482a

rsc.li/digitaldiscovery

1 Introduction

In recent years, the role of Large Language Models (LLMs) in scientific research has gained increasing attention, with a growing number of studies focusing on Artificial-Intelligence (AI)-driven methods for extracting data from scientific literature.[1,2] LLMs have demonstrated exceptional capabilities in processing unstructured data, allowing researchers to efficiently extract and organize large volumes of information.[3–8] This capability is especially important in fields such as materials science, where experimental data are often scattered throughout vast bodies of literature and presented in complex, inconsistent formats. Unlike computational data, which originate from computers and are typically structured and easy to manage,[9] experimental data require manual identification and integration—an effort that is both time consuming and labour intensive. As a result, there is often a shortage of high-quality structured experimental data. Leveraging the power of LLMs, it should be possible to overcome these challenges and construct comprehensive databases that can significantly enhance data-driven discovery and accelerate scientific progress.[3,10–12]

Currently, despite the promising applications of LLMs, there remains a significant challenge: the lack of a standardized, universally effective method for literature data extraction. Existing projects utilizing LLMs often employ varying approaches,[3,10–18] each with distinct methodologies, making it difficult for researchers to determine the most effective strategies for extracting relevant and accurate data from unstructured sources.

The information extraction is a sub-task of Natural Language Processing (NLP). Traditional extraction ways, which do not necessarily involve LLMs, follow the NLP methods, for example, Named Entity Recognition (NER), Relation Classification (RC, or Relation Extraction), Event Extraction (EE), *etc.*[19–22] NER can identify various parts of the text and find valuable elements (*i.e.* entities) in sentences (*e.g.*, compounds, descriptions of material properties, numerical values, and units), making it easier to extract the desired data. It is a crucial part of extraction and can be seen as a pre-task for RC, which identifies relationships between entities. The primary goal of RC is to accurately pair entities, especially when dealing with multiple similar types. Thus, an effective method involves creating a pipeline that integrates both NER and RC.[23,24]

[a]*Department of Physics, International Centre of Quantum and Molecular Structures, Shanghai University, Shanghai 200444, China. E-mail: musenli@shu.edu.cn; Jeffrey.reimers@uts.edu.au*

[b]*School of Materials Science and Engineering, Materials Genome Institute, Shanghai University, Shanghai 200444, China*

[c]*School of Mathematical and Physical Sciences, University of Technology Sydney, Ultimo, New South Wales 2007, Australia*

[d]*Supercomputer Facility, Australian National University, Canberra, ACT 2601, Australia. E-mail: rika.kobayashi@anu.edu.au*

After the rise of LLMs as a powerful NLP tool, researchers began to use them for data extraction tasks. For example, Carlini *et al.* used GPT-2 to extract hundreds of verbatim sequences from training data.[25] Dunn *et al.* showcase fine-tuned GPT-3's ability to perform joint NER and RC for hierarchical information in materials science.[3] Foppiano *et al.* evaluated three LLMs (GPT-3.5-Turbo, GPT-4, and GPT-4-Turbo) for information extraction tasks in materials science, benchmarking them against BERT-based models and rule-based approaches. Their results revealed that LLMs excel at reasoning and relation extraction.[10] Dagdelen *et al.* fine-tuned LLMs (GPT-3 and Llama-2) for NER and relation extraction in materials science, showing that LLMs can accurately produce structured, hierarchical scientific knowledge from unstructured text.[3]

Using deep-learning architectures, LLMs are implemented using a transformer architecture that features an attention mechanism to enable better learning of semantic relationships.[26] There are two primary types of transformers, Bidirectional Encoder Representations from Transformers (BERT)[27] and Generative Pre-trained Transformer (GPT).[28] In terms of usage, BERT is typically used for classification tasks while GPT is used for generation tasks. Both approaches provide extraction options based on deep learning (Deep Learning-based NLP), corresponding to extraction based on NER (NER-based NLP).

Many extraction techniques have been developed, and here we discuss four main techniques through five of the most representative tools: ChemDataExtractor,[29,30] BERT-PSIE,[13] ChatExtract,[14] LangChain,[31] and Kimi.[32] Key properties of these tools, and the relationships between them, are highlighted in Fig. 1.

ChemDataExtractor[22] and its improved version, ChemDataExtractor 2.0,[23] are widely used tools in the field. They have some built-in modules for parsing sentences, and users need to define rules for the target property value. They can extract nested data (different data connected to each other) simultaneously if nested rules are defined, *e.g.* a bandgap value at a specific temperature. They perform NLP without using LLMs.

BERT-PSIE[13] (BERT Precise Scientific Information Extractor) consists of three different models for the three main parts of the workflow. First, it filters out the sentences that contain data from papers. Then NER is performed and words are labelled. Finally, RC finds the right match and outputs the structured data. This tool provides a typical NER plus RC pipeline combined with BERT.

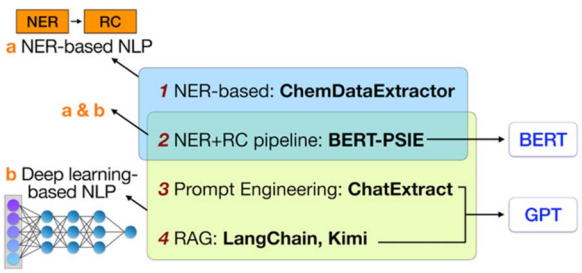

Fig. 1 Relationship between the representative techniques and tools. ChemDataExtractor and BERT-PSIE, grouped within the blue rectangle, are based on NER. BERT-PSIE, ChatExtract, LangChain, and Kimi, grouped within the (overlapping) green rectangle, are based on deep learning. The second tool, BERT-PSIE, is a combination of both main NLP ways. In terms of transformer types, BERT-PSIE uses BERT models, while ChatExtract, LangChain, and Kimi use GPT models.

ChatExtract[14] implements a specific set of prompts designed to extract data from papers, which is known as Prompt Engineering, a process that involves refining and crafting instructions for LLMs to ensure they generate accurate and relevant outputs. The key to this approach, as opposed to directly asking for information step by step, is that it introduces additional input (informing the model that it may have made mistakes during the extraction process), allowing it to reconsider and correct previously extracted information, which improves accuracy.

LangChain[31] is a framework that employs Retrieval-Augmented Generation (RAG)[33] to reduce hallucinations in LLMs—a phenomenon where models generate information that is non-existent. Hallucinations provide misinformation that undermines reliability. To enhance the accuracy and trustworthiness of LLMs in information retrieval systems and NLP tasks, researchers are actively exploring methods to detect and mitigate hallucinations.[34–40] The RAG technique is formed with three main parts: retrieval (R), augmentation (A), and generation (G), prioritizing information provided by the user. RAG first splits the original document into many paragraphs (chunks) using appropriate strategies, a step known as "chunking". Then, using *Embedding Models*, it organises the chunks into vector representations in a vector database, selecting the most similar chunks matching the inputted retrieval queries. In the augmentation step, these chunks are utilized as context information, being concatenated with the pre-defined extraction prompt. Finally, using the reduced text compared to one without RAG, the LLMs specified as *Inference Models* generate more accurate results.

In addition to LangChain, another RAG-based tool, Kimi,[32] is also used. It is an online Chatbot implementation based upon a closed-source LLM.

In summary, this study explores the use of tools based on four representative NLP techniques (NER-based, NER + RC pipeline, Prompt Engineering, and RAG) to extract data from randomly selected unstructured scientific publications. We evaluate these tools for performance with respect to human-extracted data for the extraction of bandgap data from scientific publications in materials science. In addition, we compare the results obtained using LLMs (the four Deep Learning-based tools) with those from a traditional tool used for literature data extraction, ChemDataExtractor. A crucial element of the analysis is the use of consistent evaluation criteria, enabling an objective comparison of the effectiveness of the different tools.

By doing so, we provide insights that can help researchers select the most suitable tool for their specific projects. Furthermore, we identify areas for improvement in existing methodologies, with the long-term goal of developing a more generalizable approach that can serve the needs of materials science and other fields reliant on structured data extraction from scientific literature. This work presents a comprehensive

overview of the data extraction methods, serving as a guide for future research.

2 Methods

The source code, inputs and outputs, and user instructions needed to reproduce our data extractions are provided in full on GitHub.[41] Except where explicitly noted, all results should be reproducible from this information. New applications of these tools may readily be developed using this resource.

2.1 Collection of the dataset

To make unbiased datasets, we collected metadata of arXiv papers using the dataset from Kaggle.[42] Then we selected papers in the materials science field (in which the "categories" field contains "mtrl-sci", the identifier for materials science in arXiv). To balance the quality and quantity of the dataset (minimizing scanned PDF files as much as possible while maximizing the number of articles), the publication time span was restricted from January 1, 2000, to November 1, 2024. This process selected 93 391 papers. From these, 200 papers were selected randomly to form the evaluation set of this work, and both the arXiv pre-print and publisher's versions were downloaded to make two, naively equivalent, datasets.

In total, 37 of the 200 published papers and 39 (ref. 43–45 and 46–81) of their arXiv variants were identified by human means as containing bandgap data. Including papers identified by one or more AI methods as also containing such data, these counts increase to 149 of the publisher papers and 142 of their arXiv variants. Hence the issue of the bandgap is seen to be very relevant in materials science, enabling our two diverse and unbiased datasets to expose manifold aspects of the data extraction process.

2.2 Classification of data presentation

Drawing upon our experience in reading scientific literature and manually extracting data, we identified diverse ways in which material data were presented in articles. This analysis led us to categorize these data presentation formats from various perspectives. To illustrate this, Table 1 shows four examples of bandgap property information found by human analysis, listing the identifiers: *Position Class*, *Value Class*, *Material*, *Value*, and *Source*. The two classes differentiate data presentations for subsequent analysis. The *Material* variable includes the materials' name, along with any categorical property required to uniquely identify the reported bandgap, *Value* depicts the bandgap value in eV, and *Source* refers to the original sentence(s) from which the data were taken, which facilitates subsequent checking of the AI-extracted results.

Based on the location of the data, we divide the data into the five *Position Classes* listed in Table 2: "Single Mention", "Multiple Mention", "Context", "Table", and "Figure". Single Mention indicates that both two main pieces of information for a material, *Material* and *Value*, appear in the same sentence just once, as in the examples from ref. 43 listed in Table 1. Alternatively, Multiple Mention indicates that either *Materials* and/or *Values* are mentioned more than once, *e.g.*, the two extracted records from ref. 44 listed in Table 1.

Context represents instances where *Material* and *Value* appear across different sentences, or where the name of the

Table 2 *Position Classes* describing the textual relationships between the *Material* description and bandgap *Value,* and the number of human-extracted records obtained from the 200 publisher-version[a] papers considered

Pos. Class	*Description*	*Count*[b]
1	Single Mention: a *Material* and its property *Value* are mentioned once and in the same sentence	43
2	Multiple Mention: more than one *Material* and property *Value* are mentioned in the same sentence	96
3	Context: features identifying/categorising *Material* and *Value* are in different sentences. This may overlap with other categories	70
4	Table: data appear in the tables of the paper	82
5	Figure: data appear in the figures of the paper	N/A[c]

[a] See "comparison" spreadsheets in the SI for details of the results for both the publisher and arXiv versions. [b] Up to two relationships have been ascribed to each data record, see the SI. [c] Data in figures are not considered in this work.

Table 1 Data structure showing four examples of *Source* information and how these are represented in terms of *Material* name and measurement conditions, bandgap *Value*, *Position Class* (see Table 2) and *Value Class* (see Table 3). For the complete table, please refer to the Excel file on manual data extraction in the SI

Paper	*Position Class*	*Value Class*	*Material*	*Value* (eV)	*Source*
Ref. 43	1	1	$SrFBiS_2$	0.8	… $SrFBiS_2$ is a semiconductor with a direct bandgap of **0.8 eV** …
Ref. 44	2	2	H-$MoSe_2$	∼1.13	H-$MoSe_2$ … with a direct bandgap of **about 1.13 eV** while T-$MoSe_2$, ZT-$MoSe_2$ and SO-$MoSe_2$ are zero bandgap materials
Ref. 44	2, 4	1	T-$MoSe_2$	0	H-$MoSe_2$ … with a direct bandgap of about 1.13 eV while T-$MoSe_2$, ZT-$MoSe_2$ and SO-$MoSe_2$ are **zero bandgap** materials; also in Table 1 of the original paper
Ref. 45	3	2	SmN (AFM phase, optical absorption measurements)	∼0.7	… were limited to the AFM phase of SmN. …(***sentences***)… Optical absorption measurements indicate the existence of a gap of **about 0.7 eV**

material requires contextual information to be uniquely represented. This is quite common as materials are often replaced by pronouns, requiring analysis of the preceding text, which explains what that pronoun refers to. This type of data challenges a tool's ability to read and summarize context.

The last two categories, Table and Figure, refer to cases where material property data are in tables or figures of papers. For the processing of tables, since the tools we use involve first converting them into text and then extracting information from that text, the extraction effect varies depending on the tool's ability to understand the processed text. Indeed, only LangChain and Kimi can extract data from tables and, in this work, data in figures are not considered. In principle, there are tools that would facilitate extracting data from figures, including those that recover the original data behind charts[82,83] and those that "understand" images by aligning image pixels and text descriptions.[84,85] Nevertheless, to date, such tools have not achieved the robustness needed to handle arbitrary image layouts, making quantitative analysis premature.

In addition, according to the type of numerical values present, we categorise the data into five *Value Classes* as listed in Table 3: "Fixed", "Estimated", "Range", "Bounded", and "Change". The ideal situation is an unambiguous single Fixed value, for example in Table 1 from ref. 43, "$SrFBiS_2$ is a semiconductor with a direct bandgap of 0.8 eV". However, sometimes the data are not a Fixed value and appears as in Table 1 from ref. 45 as "a gap of about 0.7 eV", which falls into the second category, Estimated value. A common alternate form of this is "$\sim$0.7 eV". In addition, sentences such as "with a bandgap more than 5 eV" are categorized as Bounded. We classify sentences similar to "the bandgap decreases from 2.07 eV by 0.15 eV" as Change, which adds considerable complexity to the data.

2.3 Data processing

In this work, we used papers in PDF format as input because all the papers within the selected time span contain PDF versions. PDFs without a fixed structure are more difficult to parse than formats such as HTML/XML, so subsequent extension to include other formats should be straightforward.

Herein, we uniformly used PyMuPDF[86] to parse PDF files and converted them into plain text, processing them into one sentence per line and saving them to TXT files. The subsequent processing of the five tools varied. For ChemDataExtractor and ChatExtract, the TXT file obtained from the previous step was directly inputted, whereas ChatExtract required the sentences presented line by line to meet the requirement for sentence-by-sentence processing. For BERT-PSIE, we further processed the TXT file into JSON format[87] (a lightweight data interchange format easy for humans to read and for machines to parse) to meet its input requirements. LangChain required unprocessed PDF files to fit the mechanism of the original code, but PyMuPDF[86] was still used for parsing. For Kimi, we directly imported the original PDF file without doing any data processing.

After extraction, post-processing was performed. First, we unified the outputs in different formats, which yielded the initial extracted data. Then, we conducted preliminary cleaning, removing entries that are, for example, duplicated or clearly not our target data, resulting in cleaned data. Finally, we normalized the data, unifying the units to eV and formats of representation, for final evaluation and comparison. Of note, optical bandgaps reported as wavelengths were not included in any of the analyses presented.

Table 3 *Value Classes* describing the type of bandgap *Value*, and the number of human-extracted records obtained from the 200 publisher-version[a] papers considered

Value Class	*Description*	*Count*[b]
1	Fixed (*e.g.*, = 1.07 eV)	144
2	Estimated (*e.g.*, $\sim$1.07 eV)	28
3	Range (*e.g.*, $\pm$0.02 eV)	47
4	Bounded (*e.g.*, <1.07 eV)	3
5	Change (*e.g.*, increased by 0.3 eV)	3

[a] See "comparison" spreadsheets in the SI for details of the results for both the publisher and arXiv versions. [b] Up to two classes have been ascribed to each data record, see the SI.

2.4 Methodological implementations

Our application of ChemDataExtractor followed the method developed by Dong *et al.*[15] for extracting bandgap and temperature values, using their Snowball model applied to the unified evaluation dataset we prepared. Owing to compatibility issues with the package, this tool was implemented indirectly through use of Docker,[88] which unfortunately was computationally very inefficient.

The original BERT-PSIE project provided models trained to extract bandgap values and Curie temperature values.[13] We modified the code to apply its three fine-tuned models to only extract bandgap values.

One of the focuses of ChatExtract's original work[14] was the design of a group of extraction prompts, providing different prompts based on the feedback from the *Inference Model* used. We further conducted Prompt Engineering based on the original work's prompt group to adapt to our unified post-processing; detailed prompts provided in the SI. In addition, we modified the online GPT-4 model that they chose, which is related to our Kimi implementation (see SI Section S2), to one of three widely used offline open-source *Inference Models*: Llama2:13b,[89] Llama3.1:70b,[90] and Qwen2.5:14b,[91] see Table 4. All three models were encapsulated in GGUF format to balance model size and performance.[92] GGUF is a binary format that is designed for fast loading and saving of models, as well as ease of reading.

The three locally deployed *Inference Models* for ChatExtract were also utilized in LangChain's generation phase (G, the third phase in RAG). Before that, two *Embedding Models* were implemented in the first phase, retrieval (R), to process the plain text data produced from the PDF files: Nomic-embed-text[93] and Bge-m3.[94]

Table 4 *Embedding Models* and *Inference Models* used by different variants of ChatExtract (CE) and LangChain (LC)

Name	*Embedding Model*	*Inference Model*
CE_1	N/A	Llama2:13b
CE_2	N/A	Llama3.1:70b
CE_3	N/A	Qwen2.5:14b
LC_{11}	Nomic-embed-text	Llama2:13b
LC_{12}	Nomic-embed-text	Llama3.1:70b
LC_{13}	Nomic-embed-text	Qwen2.5:14b
LC_{21}	Bge-m3	Llama2:13b
LC_{22}	Bge-m3	Llama3.1:70b
LC_{23}	Bge-m3	Qwen2.5:14b

As an alternative to LangChain, we also tested a different RAG tool, Kimi-k1.5,[32] which is a closed-source online model known for its ability to read long texts (see SI Section S2 for details on how it implements RAG). We extracted data by manually uploading PDF files one by one to its official webpage and using the prompt consistent with LangChain.

Some tools involve internal hyperparameters that can be adjusted to enhance performance. We used the hyperparameters specified in the original studies if provided, and for LangChain we conducted optimisation tests on the hyperparameters (see the Results section).

Regarding the deployment details of RAG, we used a separate vector database for each paper, rather than incorporating all PDF files into a single vector database for retrieval. Specifically, for LangChain, we processed one paper at a time, then cleaned the vector database and established a new vector database for the next paper; for Kimi, we opened a new chat window each time.

2.5 Performance criteria

Each extracted record from a paper, as well as the absence of extracted records, can be judged according to the following criteria. Firstly, "P" and "N" are used to represent "Positive" and "Negative" results, respectively, judging whether a record is extracted. Secondly, "T" and "F" are used to represent "True" and "False" results, respectively, judging whether the extraction situation is correct. Therefore, TP, which stands for "True Positive", indicates the desired outcome of correctly extracted records, whereas TN, which stands for "True Negative", indicates the other desired outcome that no data are extracted from papers that do not contain relevant data. Alternatively, FN indicates missed records that were not extracted (the label F here means that the unextracted situation is not ideal). Finally, FP stands for "False Positive" and indicates that wrong records were extracted, either for non-existent data (hallucinations) or else just incorrectly extracted (the label F here means that the extracted situation is not ideal). The relationships between TP, FP, FN, and TN are illustrated in Fig. 2; the numbers of extracted records of each type are denoted as *TP*, *FP*, *FN*, and *TN*, respectively.

The actual records contained within a paper were determined by human data extraction. This is a time-consuming and error-prone process and was conducted by two researchers in parallel. Differences between these manually extracted data and data from AI extraction were examined in detail, occasionally resulting in revisions of the human-extracted data.

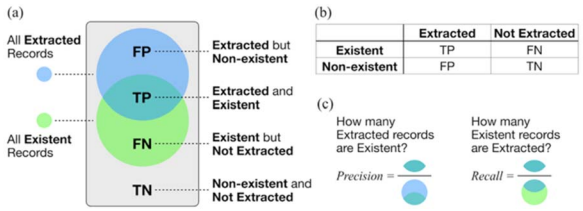

Fig. 2 Illustration of record classification based on extraction and existence status. (a) Venn diagram showing the relationship between all extracted records (blue circle) and all existing records (green circle). (b) Contingency table summarizing the classification outcomes for records based on their existence and extraction status. (c) Schematic illustration of the calculations of *Precision* and *Recall*.

2.6 Classification of error types

To help understand the factors that lead to FP generation, seven *Error Classes* were identified and are listed in Table 5. Taking the most critical entity set—*Material*, *Value*, and *Unit*—as an example, if there is a complete record including these three items, and the automatically extracted data correctly capture *Value* and *Unit*, but miss or incorrectly extract *Material*, the result is classified as FP. If the data exist but the extracted *Value* is incorrect, then this is also described as FP. Hallucinations are examples of FP results in which data are extracted whereas no data were actually present. For more details of the strict matching mechanisms used and case analyses, please refer to SI Section S3. Subtle details pertaining to the error class analysis are also provided in SI Section S4.

2.7 Performance metrics

Using the *TP*, *TN*, *FP*, and *FP* counts from our 200 examined papers in each dataset, we calculate the established *Precision*, *Recall*, and *F-score* metrics:[95–97]

$$Precision = \frac{TP}{TP + FP} = \frac{\text{correctly extracted}}{\text{all extracted}} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} = \frac{\text{correctly extracted}}{\text{all existent}} \quad (2)$$

$$F-score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (3)$$

and introduce a new metric:

$$Null\text{-}Precision = \frac{TN}{null\ papers} = \frac{\text{correctly eliminated papers}}{\text{all null-result papers}} \quad (4)$$

For all four metrics, higher scores indicate better performance.

Precision is the fraction of correctly extracted records amongst all extracted ones, whereas *Recall* is the fraction of correctly extracted records from amongst all the actual records.

Table 5 *Error Classes* contributing to FP results, and the number of materials for which each class was identified from the 200 publisher-version[a] papers considered

Error Class	*Description*	*Count*[b]
1	Hallucination: extracted *Material* or *Value* absent in the paper	265
2	Insufficient *Material*: the description of *Material* is inadequate, missing necessary categorical information	294
3	Wrong *Material*: *Material* exists but no related bandgap in the paper	394
4	Inaccurate *Value*: wrong *Value Class*, *e.g.*, missing "∼" in "∼1.45 eV"	114
5	Wrong *Value*: *Value* exists but is not a bandgap	461
6	Wrong *Unit*: error extracting *Unit*, *e.g.*, "100 meV" *vs.* "100 eV"	40
7	Wrong pairing: two (or more) *Materials* have their *Values* interchanged	14

[a] See "comparison" spreadsheets in the SI for details of the results for both the publisher and arXiv versions. [b] Up to two classes have been ascribed to each data record, see SI Section S4.

Simply speaking, *Precision* focuses on the accuracy of the model, whereas *Recall* focuses on the integrity (completeness) of the generated results database. If a model has high *Precision* but low *Recall*, it means the database is accurate but incomplete. Conversely, a model with high *Recall* but low *Precision* is guessing—it is complete, but its contents are of low accuracy. The *F-score* is the harmonic mean of *Precision* and *Recall*, balancing the two.

Null-Precision is the ability of the tool to exclude papers that are of no value, thereby simplifying subsequent (human) processing of extracted data. Unlike the data entries for the other three metrics, *Null-Precision* is calculated at the article level. It utilizes TN, which is not considered in the other metrics. If the *Null-Precision* is low, then many papers containing only FPs will be brought to attention, a highly undesirable result. If *Null-Precision* is high but *Precision* is low, then the FPs are concentrated in certain papers and absent from others, an improved result; ideally both *Null-Precision* and *Precision* should be high. Taken together, these metrics provide a comprehensive perspective for analysis of the extraction results.

Many groups have considered how to interpret analysis metrics.[10,13,14,29–31,98–101] There are, however, no generally accepted values for what constitutes "good" and "bad" results, and not all metrics always need to be high; details depend on the purpose of the study. For an exploratory study seeking information on a new topic, perhaps simply a high *Null-Precision* (>90%) would suffice. Alternatively, a study could desire high *Precision* (>90%), when simply a set of examples with accurate data is required. Then again, one may wish to know all possibilities, for which high *Recall* (>90%) is required. To build a large database that is too extensive for human curation, a high *F-score* (>90%) is required.

2.8 Statistical analysis

The 200 papers were processed as a whole to determine the performance metrics. To estimate the likely errors arising from the small sample size, the papers were divided into four sets of 50, and the metric standard deviations were used to estimate error bars:

$$\text{Error bar} = \frac{\text{standard deviation}}{\sqrt{N}} \quad (5)$$

where $N = 4$ is the number of datasets.

3 Results

The records initially extracted, cleaned, and then normalised by the 12 tool variants are provided on GitHub, with the human extracted data provided in Excel spreadsheets in the SI for both the publisher-version and arXiv-version datasets. The normalised data are then combined into final "comparison" Excel files available in the SI and statistically analysed (results that appear in the main text Tables, Figures, and Text are highlighted in red therein). Of note, except for the optimisation of the explicitly described hyperparameters, no data from the comparison spreadsheets were used to influence the tools or Python code that initially processed the 200 papers.

3.1 Unexpected differences found between the arXiv and publisher datasets

Naively, it was expected that few differences would arise between the results obtained from analysing the arXiv and publisher versions of the papers, except in the uncommon circumstance that pertinent changes were made during the publication process. Instead, significant differences were found, with Fig. 3 highlighting the changes in the *F-scores* obtained from extractions performed by each of the AI tools.

The *F-scores* from the publisher PDFs were found to be slightly higher than those from arXiv PDFs for most tools, with differences ranging from marginal (*e.g.*, CDE, 15% *vs.* 15%) to more pronounced (*e.g.*, CE_3, 27% *vs.* 18%). A notable exception

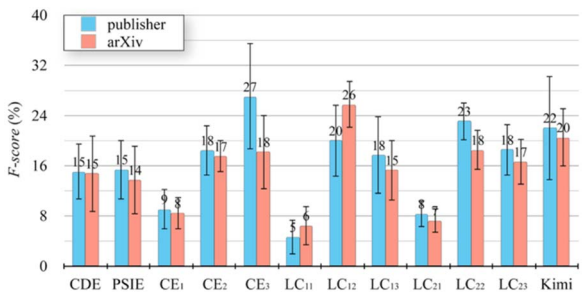

Fig. 3 Bar graph showing the *F-scores*, with error bars, obtained by the 12 AI tools from 200 arXiv and publisher version papers.

is LC_{12}, for which the arXiv version yielded a higher *F-score* (26%) compared to the publisher version (20%).

Most publisher versions embody complex and variable typesetting structures and include additional information outside the main text, whereas the structures of the arXiv versions are typically simpler and more uniform; see the SI for details on the differences in the presentation of paper content. This could account for the slightly greater error bars obtained for most tools.

Of particular interest are the results from CE_3, LC_{12}, and LC_{22}, for which the error bars obtained by dividing each dataset into four subsets are similar to the substantial differences found between the arXiv and publisher results (see Fig. 3). For example, for CE_3, the error bars are $\pm$6% for arXiv and $\pm$8% for publisher versions, with the difference between sources being 9%. In this case, including the arXiv and publisher versions together, as if they were independent publications, to make four subsets of 100 papers yields an error bar remaining at $\pm$6%. This shows that the randomness found between different data subsets can match that found when analysing two versions of the same paper. Therefore, these two different effects could have the same cause.

Human comparison of the text generated after parsing the example PDF file shows that the text processed by each tool between the arXiv and the publisher version is mostly small local changes. This was found for both the arXiv and publisher versions (see the SI). Hence the differences do indeed arise from the presentation of the local text in different global contexts, be that arising from author-generated presentation variations between papers or else from different presentations of the same paper.

3.2 Understanding RAG context analysis

To better understand the sensitivity of the observed extracted results to data presentation, we considered the mechanism of RAG in the information extraction process. We took one example paper[49] and fed different sized fractions of its content to the RAG analysis, with the size ranging from the bare minimum containing just the single feature of interest to the whole text.

Starting from the 179th sentence, which contains the content "ZnO band gap (3.4 eV)", we began with 10 sentences centred on it, expanding five sentences before and after each time, ultimately expanding to the entire article. This generated a total of 37 sets of text with gradually increasing volume for further processing. We gave them one by one to all six LangChain variants for complete processing and additionally recorded the output of each step of RAG, with the analysis summarized in the SI.

During processing, the text is divided into chunks, which are then fed into the RAG retrieval phase. In the retrieval phase, the *Embedding Model* calculates the similarity between each chunk and the retrieval query (see the Introduction section on RAG), and then ranks them. The results presented in SI Section S6 show that the retrieval query has a significant impact on the extraction results. As the amount of text in one of the 37 processed sets increases, the resulting changes in chunk segmentation alter the structure of the vector database. Even if the data-containing chunks are identical, the *Embedding Model* used in the retrieval phase can give different vector similarity scores, owing to the different global contexts in which the chunks are located. Consequently, newly added irrelevant text may obtain higher similarity scores than the target chunk, potentially causing the loss of target data and their subsequent recovery as more text is incorporated. This is one of the reasons for the difference in scores between arXiv and publisher versions of the same article and indicates intrinsic weaknesses in the RAG methods. As a means of optimising and controlling data extraction, RAG provides hyperparameters that could, in principle, alleviate these observed effects.

3.3 RAG hyperparameter selection

The most significant hyperparameters in the RAG approach are believed to be *Temperature*,[102] *Chunk-size*, *Chunk-overlap*, and *Top-k*.[103] Hyperparameters are adjustable parameters used to define any configurable part of the learning and inference processes of LLMs. Unlike the fixed parameters of LLMs after training, the hyperparameter settings used during inference can affect the output. We sought to improve the results by optimizing these hyperparameters.

3.3.1 Optimising the LangChain *Temperature*. *Temperature* is one of the most important hyperparameters in LLMs, applied during the inference stage to rescale the model's *logits* (*i.e.*, the raw output scores before softmax normalization), controlling the randomness of the predicted token distribution. In LLMs, a *token* represents the smallest unit of text the model can interpret and produce, typically corresponding to a word, subword, or symbol. It serves as the fundamental processing element on which the model operates during both training and generation.

When *Temperature* = 0, the output is deterministic and reproducible, but otherwise a random element appears so that replicated extractions from the same paper yield different results. According to Li *et al.*,[102] non-zero *Temperatures* may improve results, but there is no single *Temperature* established that can adapt to all LLMs across all types of tasks. We accordingly investigated the performance of multiple *Temperature* values in the extraction task.

We conducted extraction experiments on the 200 publisher versions using different *Temperatures* for all six variants of LangChain, and the resulting *F-scores* are shown in Fig. 4. The *Temperature* values used were 0, 0.25, and 0.8, with the set at 0.8 repeated a second time to judge the reproducibility of extractions with non-zero temperature. The figure shows that adjusting *Temperature* does not have a significant effect. Hence, as the deterministic output at a *Temperature* value of 0 is beneficial for reproducibility and does not affecting the quality of the extraction results, we choose this hyperparameter for subsequent extraction work. This hyperparameter is also applied to the LLMs of ChatExtract.

3.3.2 Optimising *Chunk-size* and *Top-k* of LangChain. The hyperparameters *Chunk-size* and *Chunk-overlap* control how text

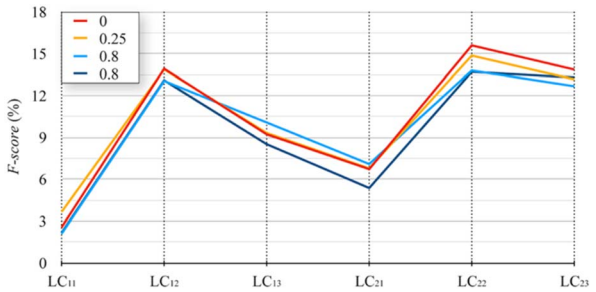

Fig. 4 The effect of varying *Temperature* on the *F-score* achieved by the six LangChain variants, setting values of 0, 0.25, 0.8, and then a repeated extraction at 0.8.

is segmented and overlapped before being embedded into the vector database. The former determines the maximum number of tokens allowed when splitting the original text into chunks, whereas the latter controls how many tokens can overlap between consecutive chunks. This acts to prevent useful information from being inadvertently split, separating *Material* and *Value* identifiers. In this study, *Chunk-overlap* is set to 1/5 of *Chunk-size*, leaving only one adjustable hyperparameter.

The final hyperparameter, *Top-k*, determines the number of priority chunks retrieved during the retrieval step of RAG. *Chunk-size* and *Top-k* operate in the first two phases of RAG, collectively determining the text supplied to the LLMs for prediction in the third phase. We employ recursive chunking to hierarchically divide text into smaller segments until each meets the token limit, so *Chunk-size* is slightly larger than the actual number of tokens in each chunk. Therefore, the amount of text seen by the LLMs can be roughly estimated as the product of *Chunk-size* and *Top-k*.

We conducted four sets of experiments by varying *Chunk-size* and *Top-k*; detailed descriptions are provided in the SI. Fig. 5 compares the effects of *Top-k* and *Chunk-size*, with data presented as *F-scores*. The comparison shows that *Chunk-size* = 1000 generally achieves higher scores than *Chunk-size* = 500. When *Chunk-size* = 500, increasing *Top-k* consistently improves results, seemingly suggesting that more contextual information is better. However, when *Chunk-size* = 1000, increasing *Top-k* actually decreases *F-scores* for LC_{21}, LC_{22}, and LC_{23}, indicating that this assumption is not valid, the optimal context amount is between the two states, and we choose the overall better *Chunk-size* = 1000 and *Top-k* = 10 for subsequent extractions. Overall, *Chunk-size* plays the dominant role, with *Top-k* providing opportunities for fine-tuning in specific-use cases.

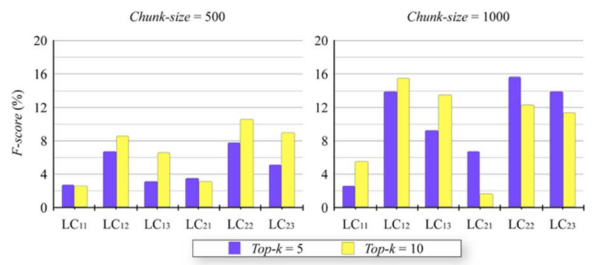

Fig. 5 Bar graph presenting a comprehensive *F-score* analysis of how the *Top-k* and *Chunk-size* hyperparameters independently influence performance for the six LangChain methods.

3.4 Performance of the 12 AI tools

Henceforth, analysis is performed only for the optimised choices of hyperparameters and for the publisher versions of the 200 papers. Human analysis revealed *TP* + *FN* = 220 as the number of *existent* records. This number is used to determine *Recall* (see eqn (2)). Notably, these 220 data records come from 37 of the 200 papers, which means that the *number of null papers* = 163; this value is used to determine *Null-Precision* (see eqn (4)).

Table 6 summarizes the performance of the 12 AI tool variants. It lists the total number of AI-extracted results (*TP* + *FP*), as well as *TP*, *Precision*, *Recall*, *F-score* and *Null-Precision*, obtained using tools based on ChemDataExtractor,[29,30] BERT-PSIE,[13] ChatExtract,[14] LangChain,[31] and Kimi.[32] The results for *Precision*, *Recall*, *F-score*, and *Null-Precision* are also presented graphically in Fig. 6(a), along with estimated error bars (see comparison spreadsheets in the SI for full details). The error bars are sufficiently small to indicate the robustness of the major qualitative features identified, but large enough to require caution in making detailed quantitative comparisons.

Considering all of the results, the best performance was found for the Prompt Engineering-based ChatExtract tool CE_3. Compared to the 220 human-extracted records, it extracted *TP* = 43 correct records, achieving a *Recall* of 20% (eqn (2)), as well as *FP* = 56 incorrect records, achieving a *Precision* of 43% (eqn (1)) and therefore an *F-score* of 27% (eqn (3)). Its *Null-Precision* is 97% (eqn (4)), making errors only five times when considering the 163 papers that did not actually contain bandgap data.

We can clearly see from Table 6 that the three tool variants CE_1, LC_{11}, and LC_{21} (which all use *Inference Model* Llama2:13b)

Table 6 Evaluation of extracted results (eqn (1)–(4)) from ChemDataExtractor[29,30] (CDE), BERT-PSIE[13] (PSIE), ChatExtract[14] (CE), LangChain[31] (LC), and Kimi-k1.5 (ref. 32) (Kimi), including the relative computational time cost

Tool	*TP* + *FP*[a]	*TP*	*Precision* (%)	*Recall* (%)	*F-score* (%)	*Null-Precision* (%)	*Rel. Cost*
CDE	47	20	43	9	15	100	185[b]
PSIE	54	21	39	10	15	99	1
CE_1	424	29	7	13	9	58	102
CE_2	52	25	48	11	18	98	252
CE_3	99	43	43	20	27	97	65
LC_{11}	174	9	5	4	5	67	4
LC_{12}	51	27	53	12	20	98	17
LC_{13}	64	25	39	11	18	96	4
LC_{21}	168	16	10	7	8	64	5
LC_{22}	75	34	45	15	23	94	18
LC_{23}	83	28	34	13	18	94	5
Kimi	153	41	27	19	22	99	N/A

[a] The sum *TP* + *FP* is the total count of data records extracted by each tool variant (after necessary post-processing cleaning). [b] Software translation is required due to device compatibility, resulting in the high cost; CDE is normally considered to be a cheap tool.

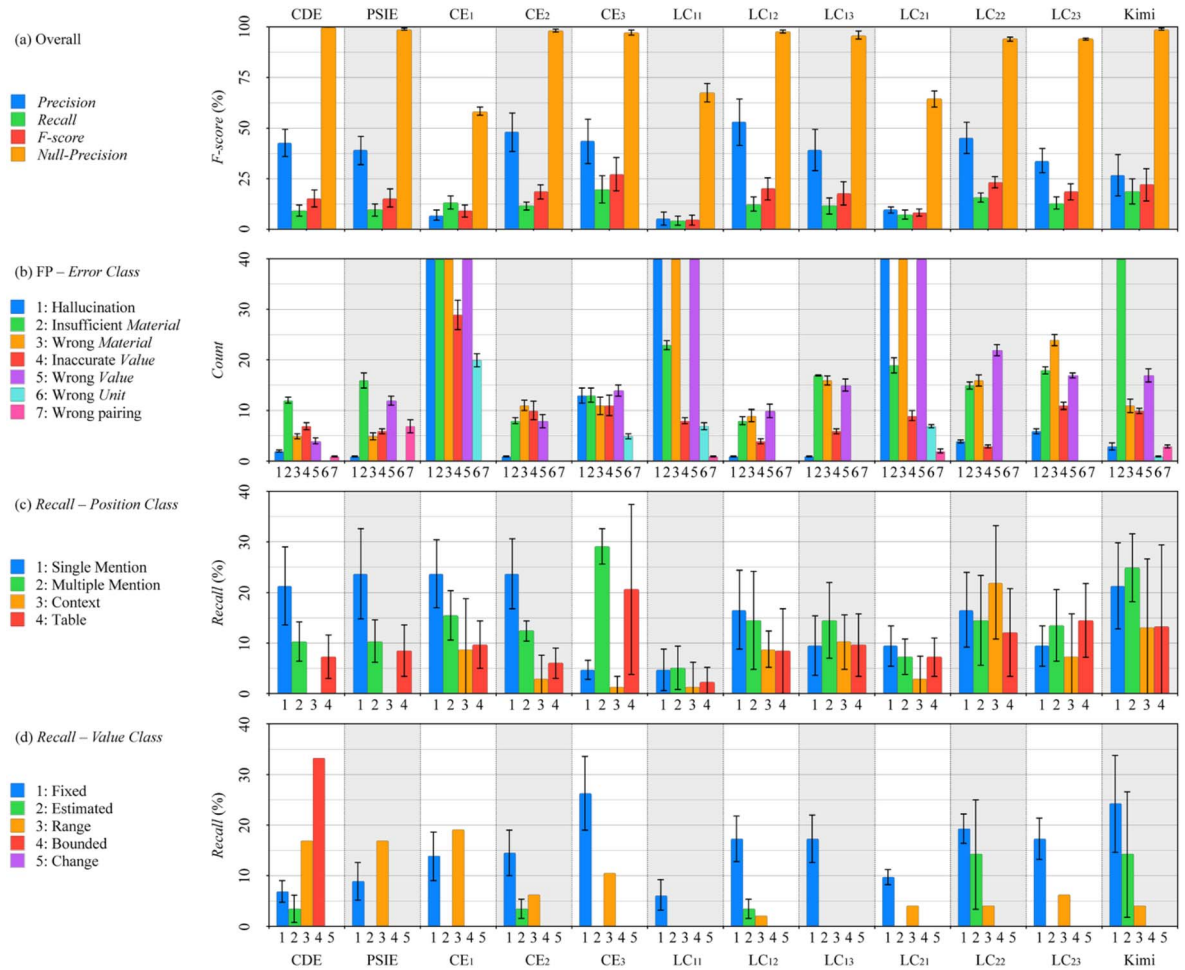

Fig. 6 Multi-perspective analysis of the 12 tool variants using the publisher-version dataset (see the SI for data for the arXiv version). Error bars come from the analysis of four datasets of 50 papers each. The four subplots have a consistent arrangement of tools on the *x*-axis, *i.e.*, the top and bottom of a block correspond to the same tool. (a) Bar graph illustrating *Precision*, *Recall*, *F-score*, and *Null-Precision* metrics (see Table 6 for detailed data). (b) *Count* of *FP* categorized by *Error Class* (refer to Table 5); for *Counts* between 40 and 196, see the SI. (c) Bar graph depicting the relationship between *Recall* and *Position Class* (see Table 2); Class 5 not included. (d) Bar graph depicting the relationship between *Recall* and *Position Class*.

performed poorly. They generated significantly lower *Precision* and *Null-Precision*, as well as excessive FPs (Fig. 6(b)) compared to the other tools. Therefore Llama2:13b is ineffective for bandgap extraction and is neglected in parts of the subsequent analysis.

With the exception of those using Llama2:13b, all tools showed *Null-Precisions* of at least 94%, a generally useful result. For Kimi and LC_{23}, *Precision* was intermediary at 27% and 34%, respectively. The remaining tools delivered higher *Precision* in the range of 39–53%, with error bars between $\pm$ 6% and $\pm$ 12%, and so these tools are not strongly differentiated using our sample set containing only 200 publications. Nevertheless, for *Precision*, the best results were obtained by LC_{12} (53 $\pm$ 12%) and CE_2 (48 $\pm$ 10%). The results for *Recall* were generally poor, with the best results being for CE_3 at 20 $\pm$ 7% and Kimi at 19 $\pm$ 7%. This effect led to poor *F-score* results, with the apparent best results being CE_3 at 27 $\pm$ 8%, LC_{22} at 23 $\pm$ 3%, and Kimi at 22 $\pm$ 8%.

Taking data pertaining to the models as a whole, a comparison between ChatExtract and LangChain variants with different end number shows that the *Inference Model* Qwen2.5:14b has better alignment with the questioning style of ChatExtract (CE_3 scores higher than CE_2 overall, and the *F-score* is much higher). We find that the extraction performance of LC_{x2} is better than that for LC_{x3}, indicating that Llama3.1:70b suits LangChain better. Regarding the *Embedding Models*, a comparison between LC_{1x} and LC_{2x} shows that Bge-m3 results are generally better than those for Nomic-embed-text.

It is difficult to discriminate between the tool types (Fig. 1) owing to the magnitude of the calculated error bars, but the results for the NER-based tools, CDE and PSIE with *F-scores* of 15% each, excluding Llama2:13b, do appear to be less than those involving deep-learning NLP: CE (18–27%), LC (18–23%) and Kimi 22%. It is *Recall* rather than *Precision* that most differentiates performance amongst these tool categories.

3.5 Correlation of false positive results with *Error Class*

As a guide to the mechanisms of failure of the 12 AI tool variants, Fig. 6(b) shows the number of extractions for each *Error Class* (see Table 5, which also lists the total number of errors in each class). The calculated error bars are shown in the figure and are relatively small compared to the magnitude of the data variations.

The *Inference Model* found previously to work very poorly, Llama2:13b, consistently generated hallucinations and fails to identify either the *Material* or *Value* correctly. For the other tools, failure to extract the categorical variables needed for the unique identification of the *Material* is a recurring issue, but errors in *Units* and in pairing *Materials* with *Values* are uncommon. It appears that the tools just struggle to get all the details of the *Material* and its *Value* simultaneously correct. The diverse nature of the identified errors indicates that not one single NLP issue controls performance; instead, wide-ranging tool improvements are currently needed. From amongst the overall better-performing tools, Kimi makes most FPs (112), failing mostly to adequately characterise the *Material*. The tools with the least number of FP results are LC_{12} (24) and CE_2 (27) (plus also CDE (27)), so a strong *Inference Model* (in this case Llama3.1:70b) helps to reduce errors.

3.6 Correlation of *Recall* with *Position Class*

The assignment of the *Position Class* (Table 2) is done by human means on the extracted records and therefore is only available for TP and FN results, leading to *Recall* (eqn (2)). Fig. 6(c) graphically depicts *Recall* for each *Position Class*, visualizing the capabilities of each tool variant. Class 5 has been excluded because data extraction from images is not considered herein. As the subset of data considered is small, the error bars shown in the figure are sizable, yet small enough to identify key features.

For Class 3 (Context), Fig. 6(c) indicates *zero Recall* for CDE and PSIE, indicating that these tools completely lack contextual capability. Low *Recall* for contextual information is also found for CE, which uses sentence-by-sentence questioning. Contextual recognition is better facilitated by the RAG tools, with LC and Kimi delivering significantly better results than the other tool variants.

It is noteworthy that for even the seemingly simplest *Position Class* 1, in which the properties to be extracted appear alone in a single sentence, none of the tools achieved a *Recall* above 25%. After examining the *Error Classes* of the extracted data entries related to *Position Class* 1 (43 entries; see Table 2), it was found that typically the *Error Class* was 2 (Wrong *Material*), 3 (Inaccurate *Material*), or 5 (Wrong *Value*), indicating that the data were found but not accurately extracted for a variety of reasons.

Also, the best tool for extracting data present as *Position Class* 2 (Multiple Mention) and 4 (Table) was found to be CE_3. This feature contributes to it being the tool with the highest overall *Recall*.

Note that tools such as CE_1 that do not parse tables achieved non-zero scores for *Recall* from Tables as some of the data in the tables are also mentioned in the text.

3.7 Correlation of *Recall* with *Value Class*

Like *Position Class*, *Value Class* is only available for TP and FN results and hence is a descriptor of *Recall* only (eqn (2)). Table 3 lists the number of records extracted from the publisher versions in each *Value Class*, with 144 having Class 1 (Fixed), 28 having Class 2 (Estimated) and 47 having Class 3 (Range). Fig. 6(d) graphically depicts *Recall* for each tool, with the small amount of available data sometimes preventing error bars from being estimated or else resulting in large error ranges. *Recall* is highest for Fixed, as would be anticipated. Only LC_{22} and Kimi produced comparable results for Estimated. The best results for Range were produced by CDE, PSIE, and CE_1. Only CDE correctly extracted results for Bounded, and no tool correctly extracted results for Change.

4 Discussion

4.1 Comprehensive evaluation

A summary of the advantages and disadvantages of ChemDataExtractor, BERT-PSIE, ChatExtract, LangChain, and Kimi, considering aspects of both tool implementation and results reliability, is provided in Table 7. This is presented under headings indicating performance, training, and ease of use, and combines aspects from the Results section with practical aspects from the Method implementations.

From the performance perspective, we primarily consider features that govern the model's accuracy (*Precision*), extraction integrity (*Recall*), computational efficiency (see Table 6), and suitability as a starting point for further refinement. From the training perspective, we compare whether the model requires additional training or is ready to use, and whether it needs a large amount of human resources. Finally, we also consider user friendliness, which includes aspects such as operational complexity and ease of getting started.

In summary, ChemDataExtractor is favoured by not needing training and its ease of use, but its performance in this work is not ideal. Additionally, owing to the lack of maintenance of the software package, there are compatibility issues in deployment on some devices, and the indirect approach taken herein to its implementation led to large unnecessary computational costs. BERT-PSIE similarly has not demonstrated the expected level here, with both BERT-PSIE and ChemDataExtractor being too cautious, extracting only very small amounts of data. ChatExtract does not need training, but it is very time-consuming owing to the large number of input and output sentences. Of note, the tool CE_3 delivered the best *F-score* of all tools considered. The overall effect of LangChain is acceptable, and it does not need training, but the *Inference Model*, *Embedding Model* and associated hyperparameters need to be optimised. Most, but not all, LangChain variants delivered on the promise of avoiding hallucinations, with some ChatExtract tools showing

Table 7 Evaluation of the practicality of the tools considered in terms of positive (+) and negative (−) features

Tool	Performance accuracy, integrity, and efficiency	Training difficulty and manual work	Ease of use
CDE	+ Includes table parser + Handles various expressions (pronouns) + Extracts nested properties + Excellent irrelevant content filtering + Can extract data ranges + Moderate *Precision* − Undesirable *Recall* − Cannot extract data in context − Cannot extract from figures	+ No training required + Single input, no additional operation required for extraction − Requires parser training for optimization − Needs repeated rule modifications for nested rules	+ Built-in modules + Simple post-processing (structured output) + Extracts with sentence source for verification − New rule definitions required for new properties − Domain knowledge-dependent
PSIE	+ Stable output + Can extract data ranges + Moderate *Precision* − Undesirable *Recall* − Cannot extract nested properties − Poor context sensitivity (sentence-level) − Cannot extract data in context − Cannot extract from tables/figures	+ Low training data requirement for new properties (fine-tuning) + Single input, no additional operation required for extraction − Requires manual labelling of training data − Challenging RC linear layer design	+ No intricate grammar rules + Extracts with sentence source for verification − Needs new models for new properties
CE	+ Includes result confirmation + Extracts nested properties + Can extract data ranges + Best at extracting data from tables + Can show moderate *Precision* − Poor context sensitivity (sentence-level) − Undesirable *Recall* − Cannot extract from figures − Slow processing	+ No training required + Single input, no additional operation required for extraction − Needs testing for best LLMs	+ Prompt modification suffices for new tasks + Extracts with sentence source for verification − May require closed-source LLMs for optimal results − Needs prompt adjustment for stability
LC	+ Document-level processing + Extracts from tables/figures + Extracts nested properties + Context awareness (low effectiveness) + Can extract estimated values + Good match between cost effectiveness and performance + Can show moderate *Precision* − Undesirable *Recall*	+ No training required + Single input, no additional operation required for extraction − Requires *Embedding Model* training (for better performance) − Needs testing for best LLMs − Contains hyperparameters that should be optimised	+ Prompt modification suffices for new tasks + Deployment flexibility and scalability − Extracting sentence sources simultaneously may result in a loss of accuracy
Kimi	+ Document-level processing + Extracts from tables/figures + Extracts nested properties + Context awareness (low effectiveness) + Moderate *Precision* and *Recall* − Accuracy depends on Online LLMs	+ No training required − Uncontrollable performance − Model hyperparameters cannot be customized − Requires per-article web interaction manually	+ Prompt modification suffices for new tasks + Ready-to-use − Extracting sentence sources simultaneously may result in a loss of accuracy − Network-dependent

similar performance. Kimi's performance is comparable with the best other tools considered and is easy to use, but the LLM's effects depend on the vendor, and the interactions depend on the network, limiting reproducibility.

For all models, parsing context information was a challenge. This could involve the context of the *Material* and *Value* entities, as used in the *Position Class*, but also generating the *Material* name from contextual information including the deciphering of pronouns. An example of this is provided from ref. 45 in Table 1. ChemDataExtractor and LangChain have made attempts at solving contextual issues, but the performance for the 200 papers considered is poor. ChemDataExtractor has a Forward-looking Dependency Resolution mechanism to deal with variations in naming. For example, white phosphorus is referred to as "white-P" throughout an article,[53] and the appropriate name translation must be made in the output records. In addition, it will also consolidate all data related to the same *Material* at the end. LangChain does this by accepting the entire paper into a vector database, only considering relevant paragraphs. Alternatively, both BERT-PSIE and ChatExtract process papers only at the sentence level and hence lack the ability to integrate contextual information.

Also, beyond the scope of this work, it is likely that the user will need to change the target of the information extraction tool from bandgap to other material properties. For example, in rule-based tools such as ChemDataExtractor, it is necessary to re-define the extraction rules, primarily by modifying the parsing expressions. To achieve comprehensive extraction, these parsing expressions must account for all potential variations, which require both domain knowledge and iterative testing for refinement. In contrast, BERT-PSIE does not rely on predefined rules but instead requires training new models for each new property. This model training process demands manually labelled data, which can be time-consuming. However, it may not be necessary to retrain the model for each new property if the underlying BERT model is sufficiently robust. For ChatExtract and LangChain, the process is simplified as they only require specifying the name of the new property to the LLM, making migration straightforward.

4.2 Further prospects

Each tool has its own challenges. Currently, there is no tool that can achieve a level suitable for large-scale automated use. One possible recommendation for researchers regarding tool selection is that it is still necessary to use a combination of various tools and integrate the results. For example, we believe that Prompt Engineering (ChatExtract) can be used in conjunction with RAG technology, leveraging the ChatExtract's ability to reconsider and correct extracted records, sacrificing more resources to further improve the extraction accuracy and completeness of RAG.

Based on the data extraction results shown in Table 6, the traditional NLP counting tool, ChemDataExtractor, appears to be gradually replaced by modern LLM tools. However, it is not necessary to completely abandon ChemDataExtractor, as it can also be used in other ways.[104]

For all tools, an improvement would be to add a "de-contextualization" pre-processing step, which replaces pronouns and other elements in the sentences with their original content in advance through semantic analysis by LLMs, thereby reducing the processing burden on the model during the extraction phase.

For RAG models, improved *Embedding Models* and *Inference Models*, as well as hyperparameter optimisation, can be envisaged. Also, we can further fine-tune LLMs[105,106] (whether it's the BERT model in BERT-PSIE or the GPT model in ChatExtract or LangChain) using data from the field of materials science to enhance the LLM's understanding of domain knowledge.

For RAG implementation, semantic chunking[107–109] or adaptive retrieval queries[110,111] may also be a possible way. In addition to adding re-ranking to optimize retrieval,[112] there are also many improved RAG frameworks that can further aid in extraction.[113–115]

We also found that LLMs have a limited ability to follow instructions ("lack of focus"), *e.g.*, if the LLM is simultaneously asked to extract data and output them in a given structure, it is likely to do neither task well. Our analysis of the *Error Class* in Section 3.4 also illustrates this as tools struggle to get all the details correct simultaneously. The best approach to this is to provide detailed execution steps and simplify the single task.

It is also possible to combine the information extraction with the latest advances in reinforcement learning to mimic reasoning, such as Chain of Thought (CoT).[116–118] This can be easily achieved by directly replacing existing LLMs with LLMs that have reinforcement learning CoT, or by autonomously building workflows with reasoning nodes.

The extraction of data from tables and figures presents a significant challenge, particularly when dealing with complex visual elements. While the focus here is not on figure extraction, advancements in technology offer promising solutions. For example, the use of models equipped with image analysis capabilities, *e.g.*, visual language models (VLMs)[85] or multimodal large language models (MLLMs),[84] to calculate the *Null-Precision* of the figures (assess whether a figure contains the target data), then humans could manually extract information from this narrowed set of relevant figures.

This work also has some limitations. For example, the number of examined articles (200) is too small to support a more diverse comparison, as the sometimes large obtained error bars indicate. Also, the data extraction is only at the numerical level, without including categorical properties (*e.g.*, data type, bandgap type, and crystal structure) that are not required to uniquely identify the *Material* within the source's context. The number of tested LLMs is also insufficient to determine the impact of parameter quantity or quantification accuracy on extraction effectiveness.

We will conduct further research, exploring not only the best ways to use the models but also attempting to more specifically adjust the model architecture, continuously trying the latest technologies to obtain a practically usable data extraction tool, and using it to extract a large-scale experimental materials science database.

5 Conclusions

The use of Natural Language Processing to extract data from publications could greatly benefit the scientific community. Nevertheless, the language and contextual issues associated with the presentation of scientific (and other) data are complex, leading to significant challenges.

Herein, five representative AI tools were applied to extract bandgap information from 200 randomly selected papers. Except for the optimisation of explicit hyperparameters, the models, and the way they were implemented, were not modified to meet the challenges presented by the individual papers from within our unbiased datasets, and so the results obtained may fall below what could be achievable in the future.

In particular, the results obtained for *Precision* were encouraging, with many tools achieving scores of 40–50%, but *Recall* is identified as a key current issue, with the best tools only achieving 20%. Tools based on either ChatExtract or LangChain were found to deliver the best results. A feature that has not been previously widely recognised is that fine details of the data presentation, *e.g.*, document origins from arXiv *versus* publisher versions, can have a significant influence on the quality of the data extraction. From the perspective of users wanting to apply these tools to extract data, it would be common for highlighted papers to be manually overviewed and key results verified. Central to this is the *Null-Precision* of the tools as high *Null-Precision* minimises such intense manual effort. It is encouraging that most tools showed *Null-Precision* values exceeding 94%.

Nevertheless, the issue of extracting elements located in context challenges all current tools. The development of a comprehensive database of bandgap values would be a major contribution to the advancement of materials science, but this must be done with high *Precision*, *Recall*, and *Null-Precision*.

Author contributions

W. N. designed the research, performed all data extractions, conducted Python programming to establish the combined spreadsheet, performed manual data extractions and data analysis, and wrote the manuscript. M. L. contributed to the research design and implementation. J. R. R. supervised data analysis design and implementation, as well as edited the manuscript. R. K. conceived and supervised the project.

Conflicts of interest

There are no conflicts to declare.

Data availability

All developed source codes, inputs and outputs, plus Excel files depicting results from each stage of the data extraction process, are available at GitHub (https://github.com/wenkaining/Bandgap-Extraction-Comparison), https://doi.org/10.5281/zenodo.17785333.

All mathematical analyses of the data are provided in the supplementary information (SI). Supplementary information: supplementary text plus Excel spreadsheets containing the data analyses. See DOI: https://doi.org/10.1039/d5dd00482a.

Acknowledgements

We thank the Australian Research Council for funding this research under Grant CE230100021, the Young Scientists Fund of the National Natural Science Foundation of China (Grant No. 12404276), the Special Funds of the National Natural Science Foundation of China (Grant No. 12347164), the China Postdoctoral Science Foundation (Grant No. 2024T170541 and GZC20231535), and the University of Technology Sydney for the provision of computing resources at the National Computational Infrastructure, Australia.

References

1 D. Xu, W. Chen, W. Peng, C. Zhang, T. Xu, X. Zhao, X. Wu, Y. Zheng, Y. Wang and E. Chen, Large Language Models for Generative Information Extraction: A Survey, *Front. Comput. Sci.*, 2024, **18**, 186357.

2 M. C. Ramos, C. J. Collison and A. D. White, A review of large language models and autonomous agents in chemistry, *Chem. Sci.*, 2025, **16**, 2514–2572.

3 J. Dagdelen, A. Dunn, S. Lee, N. Walker, A. S. Rosen, G. Ceder, K. A. Persson and A. Jain, Structured information extraction from scientific text with large language models, *Nat. Commun.*, 2024, **15**, 1418.

4 V. Perot, K. Kang, F. Luisier, G. Su, X. Sun, R. S. Boppana, Z. Wang, Z. Wang, J. Mu, H. Zhang, C.-Y. Lee and N. Hua, LMDX: Language Model-based Document Information Extraction and Localization, in *Findings of the Association for Computational Linguistics: ACL 2024*, Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 15140–15168, DOI: 10.18653/v1/2024.findings-acl.899.

5 N. Zhang, X. Xu, L. Tao, H. Yu, H. Ye, S. Qiao, X. Xie, X. Chen, Z. Li, L. Li, X. Liang, Y. Yao, S. Deng, P. Wang, W. Zhang, Z. Zhang, C. Tan, Q. Chen, F. Xiong, F. Huang, G. Zheng and H. Chen, DeepKE: A Deep Learning Based Knowledge Extraction Toolkit for Knowledge Base Population, in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Association for Computational Linguistics, Abu Dhabi, UAE, 2022, pp. 98–108, DOI: 10.18653/v1/2022.emnlp-demos.10.

6 Y. Labrak, M. Rouvier and R. Dufour, A Zero-shot and Few-shot Study of Instruction-Finetuned Large Language Models Applied to Clinical and Biomedical Tasks, in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, ELRA and ICCL, Torino, Italia, 2024, pp. 2049–2066, https://aclanthology.org/2024.lrec-main.185/.

7 W. Shao, R. Zhang, P. Ji, D. Fan, Y. Hu, X. Yan, C. Cui, Y. Tao, L. Mi and L. Chen, Astronomical knowledge entity

extraction in astrophysics journal articles via large language models, *Res. Astron. Astrophys.*, 2024, **24**, 065012.

8 R. O. Nunes, A. S. Spritzer, C. M. D. S. Freitas and D. G. Balreira, Out of Sesame Street: A Study of Portuguese Legal Named Entity Recognition Through In-Context Learning, in *Proceedings of the 26th International Conference on Enterprise Information Systems - Volume 1 ICEIS*, SciTePress, 2024, pp. 477–489, DOI: 10.5220/0012624700003690.

9 S. Kirklin, J. E. Saal, B. Meredig, A. Thompson, J. W. Doak, M. Aykol, S. Rühl and C. Wolverton, The Open Quantum Materials Database (OQMD): assessing the accuracy of DFT formation energies, *npj Comput. Mater.*, 2015, **1**, 1–15.

10 L. Foppiano, G. Lambard, T. Amagasa and M. Ishii, Mining experimental data from materials science literature with large language models: an evaluation study, *Sci. Technol. Adv. Mater.:Methods*, 2024, **4**, 2356506.

11 J. Cheung, Y. Zhuang, Y. Li, P. Shetty, W. Zhao, S. Grampurohit, R. Ramprasad and C. Zhang, POLYIE: A Dataset of Information Extraction from Polymer Material Scientific Literature, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Association for Computational Linguistics, Mexico City, Mexico, 2024, pp. 2370–2385, DOI: 10.18653/v1/2024.naacl-long.131.

12 N. Bölücü, M. Rybinski and S. Wan, Impact of sample selection on in-context learning for entity extraction from scientific writing, in *Findings of the Association for Computational Linguistics, EMNLP 2023*, 2023, pp. 5090–5107, DOI: 10.18653/v1/2023.findings-emnlp.338.

13 L. P. J. Gilligan, M. Cobelli, V. Taufour and S. Sanvito, A rule-free workflow for the automated generation of databases from scientific literature, *npj Comput. Mater.*, 2023, **9**, 222.

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

15 Q. Dong and J. M. Cole, Auto-generated database of semiconductor band gaps using ChemDataExtractor, *Sci. Data*, 2022, **9**, 193.

16 M. Schilling-Wilhelmi, M. Ríos-García, S. Shabih, M. V. Gil, S. Miret, C. T. Koch, J. A. Márquez and K. M. Jablonka, From text to insight: large language models for chemical data extraction, *Chem. Soc. Rev.*, 2025, 1125–1150.

17 M. Moradi, K. Blagec, F. Haberl and M. Samwald, Gpt-3 models are poor few-shot learners in the biomedical domain, *arXiv*, 2021, preprint, arXiv:2109.02555.

18 L. Foppiano, P. B. Castro, P. Ortiz Suarez, K. Terashima, Y. Takano and M. Ishii, Automatic extraction of materials and properties from superconductors scientific literature, *Sci. Technol. Adv. Mater.: Methods*, 2023, **3**, 2153633.

19 Y. Lu, Q. Liu, D. Dai, X. Xiao, H. Lin, X. Han, L. Sun and H. Wu, Unified Structure Generation for Universal Information Extraction, in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics*, Dublin, Ireland, 2022, pp. 5755–5772, DOI: 10.18653/v1/2022.acl-long.395.

20 X. Wang, W. Zhou, C. Zu, H. Xia, T. Chen, Y. Zhang, R. Zheng, J. Ye, Q. Zhang, T. Gui, J. Kang, J. Yang, S. Li and C. Du, InstructUIE: Multi-task Instruction Tuning for Unified Information Extraction, *arXiv*, 2023, preprint, arXiv:2304.08085, DOI: 10.48550/arXiv.2304.08085.

21 Y. Guo, Z. Li, X. Jin, Y. Liu, Y. Zeng, W. Liu, X. Li, P. Yang, L. Bai, J. Guo and X. Cheng, Retrieval-Augmented Code Generation for Universal Information Extraction, in *Natural Language Processing and Chinese Computing*, ed. D. F. Wong, Z. Wei and M. Yang, Springer Nature Singapore, Singapore, 2025, vol. 15360, pp. 30–42, DOI: 10.1007/978-981-97-9434-8_3.

22 Y. Zhong, T. Xu and P. Luo, Contextualized Hybrid Prompt-Tuning for Generation-Based Event Extraction, in *Knowledge Science, Engineering and Management*, ed. Z. Jin, Y. Jiang, R. A. Buchmann, Y. Bi, A.-M. Ghiran and W. Ma, Springer Nature Switzerland, Cham, 2023, vol. 14120, pp. 374–386, DOI: 10.1007/978-3-031-40292-0_31.

23 G. Bekoulis, J. Deleu, T. Demeester and C. Develder, Joint entity recognition and relation extraction as a multi-head selection problem, *Expert Systems with Applications*, 2018, **114**, 34–45.

24 J. Li, H. Fei, J. Liu, S. Wu, M. Zhang, C. Teng, D. Ji and F. Li, Unified Named Entity Recognition as Word-Word Relation Classification, *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022, **36**, 10965–10973.

25 N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson, A. Oprea and C. Raffel, Extracting Training Data from Large Language Models, in *Proceedings of the 30th USENIX Security Symposium*, 2021, pp. 2633–2650, https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting.

26 A. Vaswani, Attention is all you need, in *Advances in Neural Information Processing Systems Vol. 30*, ed. I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett, Curran Associates, Inc., 2017, https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

27 J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186, DOI: 10.18653/v1/N19-1423.

28 A. Radford, K. Narasimhan, T. Salimans and I. Sutskever, Improving language understanding by generative pre-training, arXiv preprint arXiv:1810.04805, 2018, DOI: 10.48550/arXiv.1810.04805.

29 M. C. Swain and J. M. Cole, ChemDataExtractor: A Toolkit for Automated Extraction of Chemical Information from

the Scientific Literature, *J. Chem. Inf. Model.*, 2016, **56**, 1894–1904.
30 J. Mavračić, C. J. Court, T. Isazawa, S. R. Elliott and J. M. Cole, ChemDataExtractor 2.0: Autopopulated Ontologies for Materials Science, *J. Chem. Inf. Model.*, 2021, **61**, 4280–4289.
31 LangChain, https://github.com/langchain-ai, accessed 2024-11-28 18:35:12.
32 K. Team, A. Du, B. Gao, B. Xing, C. Jiang, C. Chen, C. Li, C. Xiao, C. Du, C. Liao, *et al.*, Kimi k1.5: Scaling reinforcement learning with llms, *arXiv*, 2025, preprint, arXiv:2501.12599.
33 P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih and T. Rocktäschel, Retrieval-augmented generation for knowledge-intensive nlp tasks, *Advances in Neural Information Processing Systems*, 2020, **33**, 9459–9474.
34 S. Farquhar, J. Kossen, L. Kuhn and Y. Gal, Detecting hallucinations in large language models using semantic entropy, *Nature*, 2024, **630**, 625–630.
35 L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin and T. Liu, A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions, *ACM Transactions on Information Systems*, 2024, **43**, 1–55.
36 X. Jiang, Y. Tian, F. Hua, C. Xu, Y. Wang and J. Guo, A Survey on Large Language Model Hallucination via a Creativity Perspective, *arXiv*, 2024, preprint, arXiv:2402.06647, DOI: 10.48550/arXiv.2402.06647.
37 V. Rawte, S. Chakraborty, A. Pathak, A. Sarkar, S. M. T. I. Tonmoy, A. Chadha, A. Sheth and A. Das, The Troubling Emergence of Hallucination in Large Language Models - An Extensive Definition, Quantification, and Prescriptive Remediations, in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Singapore, 2023, pp. 2541–2573, DOI: 10.18653/v1/2023.emnlp-main.155.
38 M. Zhang, O. Press, W. Merrill, A. Liu and N. A. Smith, How Language Model Hallucinations Can Snowball, in *Proceedings of the 41st International Conference on Machine Learning*, ed. S. Ruslan, K. Zico, H. Katherine, W. Adrian, O. Nuria, S. Jonathan and B. Felix, PMLR, Proceedings of Machine Learning Research, 2024, pp. 59670–59684, https://proceedings.mlr.press/v235/zhang24ay.html.
39 S. Zhang, L. Pan, J. Zhao and W. Y. Wang, The Knowledge Alignment Problem: Bridging Human and External Knowledge for Large Language Models, in *Findings of the Association for Computational Linguistics: ACL 2024*, Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 2025–2038, DOI: 10.18653/v1/2024.findings-acl.121.
40 W. Xu, S. Agrawal, E. Briakou, M. J. Martindale and M. Carpuat, Understanding and detecting hallucinations in neural machine translation via model introspection, *Transactions of the Association for Computational Linguistics*, 2023, **11**, 546–564.
41 W. Ning, wenkaining/Bandgap-Extraction-Comparison, https://github.com/wenkaining/Bandgap-Extraction-Comparison, accessed 2025-03-31 19:22:03.
42 Kaggle, https://www.kaggle.com/dsv/7548853, (accessed 2025-05-25 22:29:00, DOI: 10.34740/KAGGLE/DSV/7548853).
43 H. Lei, K. Wang, M. Abeykoon, E. S. Bozin and C. Petrovic, New Layered Fluorosulfide SrFBiS2, *Inorg. Chem.*, 2013, **52**, 10685–10689.
44 S. Kaur, A. Kumar, S. Srivastava and K. Tankeshwar, van der Waals heterostructures based on allotropes of phosphorene and MoSe2, *Phys. Chem. Chem. Phys.*, 2017, **19**, 22023–22032.
45 C. Morari, F. Beiuşeanu, I. Di Marco, L. Peters, E. Burzo, S. Mican and L. Chioncel, Magnetism and electronic structure calculation of SmN, *J. Phys.: Condens. Matter*, 2015, **27**, 115503.
46 F. Bonnín-Ripoll, Y. B. Martynov, R. G. Nazmitdinov, K. Tabah, C. Pereyra, M. Lira-Cantú, G. Cardona and R. Pujol-Nadal, On performance of thin-film meso-structured perovskite solar cell through experimental analysis and device simulation, *Materials Today Sustainability*, 2023, **24**, 100548.
47 A. F. Santander-Syro, O. Copie, T. Kondo, F. Fortuna, S. Pailhès, R. Weht, X. G. Qiu, F. Bertran, A. Nicolaou, A. Taleb-Ibrahimi, P. Le Fèvre, G. Herranz, M. Bibes, N. Reyren, Y. Apertet, P. Lecoeur, A. Barthélémy and M. J. Rozenberg, Two-dimensional electron gas with universal subbands at the surface of SrTiO3, *Nature*, 2011, **469**, 189–193.
48 J. P. Sheckelton, J. R. Neilson, D. G. Soltan and T. M. McQueen, Possible valence-bond condensation in the frustrated cluster magnet LiZn2Mo3O8, *Nat. Mater.*, 2012, **11**, 493–496.
49 L. Gierster, S. Vempati and J. Stähler, Ultrafast generation and decay of a surface metal, *Nat. Commun.*, 2021, **12**, 978.
50 L. Yuan, B. Zheng, J. Kunstmann, T. Brumme, A. B. Kuc, C. Ma, S. Deng, D. Blach, A. Pan and L. Huang, Twist-angle-dependent interlayer exciton diffusion in WS2–WSe2 heterobilayers, *Nat. Mater.*, 2020, **19**, 617–623.
51 S. Sattar, R. Hoffmann and U. Schwingenschlögl, Solid argon as a possible substrate for quasi-freestanding silicene, *New J. Phys.*, 2014, **16**, 065001.
52 G. Thiering and A. Gali, Characterization of oxygen defects in diamond by means of density functional theory calculations, *Phys. Rev. B*, 2016, **94**, 125202.
53 J. A. Flores-Livas, A. Sanna, A. P. Drozdov, L. Boeri, G. Profeta, M. Eremets and S. Goedecker, Interplay between structure and superconductivity: Metastable phases of phosphorus under pressure, *Phys. Rev. Mater.*, 2017, **1**, 024802.
54 X. Wang, L. Cheng, D. Zhu, Y. Wu, M. Chen, Y. Wang, D. Zhao, C. B. Boothroyd, Y. M. Lam and J. X. Zhu, Ultrafast spin-to-charge conversion at the surface of topological insulator thin films, *Adv. Mater.*, 2018, **30**, 1802356.

55 B. Zou, Y. Zhou, Y. Zhou, Y. Wu, Y. He, X. Wang, J. Yang, L. Zhang, Y. Chen and S. Zhou, Reliable and broad-range layer identification of Au-assisted exfoliated large area MoS2 and WS2 using reflection spectroscopic fingerprints, *Nano Res.*, 2022, **15**, 8470–8478.
56 J. Gao, Q. Wu, C. Persson and Z. Wang, Irvsp: To obtain irreducible representations of electronic states in the VASP, *Comput. Phys. Commun.*, 2021, **261**, 107760.
57 K. L. Lima and L. R. Junior, A dft study on the mechanical, electronic, thermodynamic, and optical properties of gan and aln counterparts of biphenylene network, *Mater. Today Commun.*, 2023, **37**, 107183.
58 A. Yore, K. Smithe, W. Crumrine, A. Miller, J. Tuck, B. Redd, E. Pop, B. Wang and A. Newaz, Visualization of defect-induced excitonic properties of the edges and grain boundaries in synthesized monolayer molybdenum disulfide, *J. Phys. Chem. C*, 2016, **120**, 24080–24087.
59 M. A. Kempf, P. Moser, M. Tomoscheit, J. Schröer, J.-C. Blancon, R. Schwartz, S. Deb, A. Mohite, A. V. Stier and J. J. Finley, Rapid spin depolarization in the layered 2d ruddlesden–popper perovskite (BA)(MA) PbI, *ACS Nano*, 2023, **17**, 25459–25467.
60 T. Vogl, M. W. Doherty, B. C. Buchler, Y. Lu and P. K. Lam, Atomic localization of quantum emitters in multilayer hexagonal boron nitride, *Nanoscale*, 2019, **11**, 14362–14371.
61 S.-D. Guo, W.-Q. Mu, H.-T. Guo, Y.-L. Tao and B.-G. Liu, A piezoelectric quantum spin Hall insulator VCClBr monolayer with a pure out-of-plane piezoelectric response, *Phys. Chem. Chem. Phys.*, 2022, **24**, 19965–19974.
62 B. Deng, Y. Zhang, S. Zhang, Y. Wang, K. He and J. Zhu, Realization of stable ferromagnetic order in a topological insulator: Codoping-enhanced magnetism in 4 f transition metal doped B i 2 S e 3, *Phys. Rev. B*, 2016, **94**, 054113.
63 Y.-T. Huang, S. R. Kavanagh, D. O. Scanlon, A. Walsh and R. L. Hoye, Perovskite-inspired materials for photovoltaics and beyond—from design to devices, *Nanotechnology*, 2021, **32**, 132004.
64 L. Wen, Z. Li and Y. He, Optical conductivity of twisted bilayer graphene near the magic angle, *Chin. Phys. B*, 2021, **30**, 017303.
65 M.-Á. Sánchez-Martínez, I. Robredo, A. Bidaurrazaga, A. Bergara, F. de Juan, A. G. Grushin and M. G. Vergniory, Spectral and optical properties of Ag3Au (Se2, Te2) and dark matter detection, *J. Phys.: Mater.*, 2019, **3**, 014001.
66 J. W. Park, Y.-K. Jung and A. Walsh, Metal Halide Thermoelectrics: Prediction of High-Performance Cs Cu 2 I 3, *PRX Energy*, 2023, **2**, 043004.
67 G. Petretto and F. Bruneval, Comprehensive ab initio study of doping in bulk ZnO with group-V elements, *Phys. Rev. Appl.*, 2014, **1**, 024005.
68 D. Najer, N. Tomm, A. Javadi, A. R. Korsch, B. Petrak, D. Riedel, V. Dolique, S. R. Valentin, R. Schott and A. D. Wieck, Suppression of surface-related loss in a gated semiconductor microcavity, *Phys. Rev. Appl.*, 2021, **15**, 044004.
69 S. Michael and H. C. Schneider, Impact ionization dynamics in small band-gap two-dimensional materials from a coherent phonon mechanism, *Phys. Rev. B*, 2019, **100**, 035431.
70 S. Sharma, S. Kumar, G. C. Tewari, G. Sharma, E. F. Schwier, K. Shimada, A. Taraphder and C. Yadav, Magnetotransport and high-resolution angle-resolved photoelectron spectroscopy studies of palladium-doped Bi 2 Te 3, *Phys. Rev. B*, 2022, **105**, 115120.
71 S. Liu, C. Wang, H. Jeon, Y. Jia and J.-H. Cho, Emerging two-dimensional magnetism in nonmagnetic electrides Hf 2 X (X= S, Se, Te), *Phys. Rev. B*, 2022, **105**, L220401.
72 G. Bester and A. Zunger, Electric field control and optical signature of entanglement in quantum dot molecules, *Phys. Rev. B: Condens. Matter Mater. Phys.*, 2005, **72**, 165334.
73 W. Chen, C. Tegenkamp, H. Pfnür and T. Bredow, Anomalous molecular orbital variation upon adsorption on a wide band gap insulator, *J. Chem. Phys.*, 2010, **132**, 214706.
74 S. Cosentino, A. Terrasi, S. Mirabella, D. Lockwood and R. Costa Filho, Influence of interface potential on the effective mass in Ge nanostructures, *J. Appl. Phys.*, 2015, **117**, 154304.
75 H. Dicko, O. Pagès, F. Firszt, K. Strzałkowski, W. Paszkowicz, A. Maillard, C. Jobard and L. Broch, Near-forward Raman selection rules for the phonon-polariton in (Zn, Be)Se alloys, *J. Appl. Phys.*, 2016, **120**, 185702.
76 M. Stokey, R. Korlacki, S. Knight, A. Ruder, M. Hilfiker, Z. Galazka, K. Irmscher, Y. Zhang, H. Zhao and V. Darakchieva, Optical phonon modes, static and high-frequency dielectric constants, and effective electron mass parameter in cubic In2O3, *J. Appl. Phys.*, 2021, **129**, 225102.
77 A. Wang, K. Bushick, N. Pant, W. Lee, X. Zhang, J. Leveillee, F. Giustino, S. Poncé and E. Kioupakis, Electron mobility of SnO2 from first principles, *Appl. Phys. Lett.*, 2024, **124**, 172103.
78 Y. Yang, Q. Wang, S. Duan, H. Wo, C. Huang, S. Wang, L. Gu, D. Xiang, D. Qian and J. Zhao, Anomalous contribution to the nematic electronic states from the structural transition in FeSe revealed by time-and angle-resolved photoemission spectroscopy, *Phys. Rev. Lett.*, 2022, **128**, 246401.
79 A. Christianson, V. Fanelli, J. Lawrence, E. Goremychkin, R. Osborn, E. Bauer, J. Sarrao, J. Thompson, C. Frost and J. Zarestky, Localized excitation in the hybridization gap in YbAl 3, *Phys. Rev. Lett.*, 2006, **96**, 117206.
80 J. Xu, F. Han, T.-T. Wang, L. R. Thoutam, S. E. Pate, M. Li, X. Zhang, Y.-L. Wang, R. Fotovat and U. Welp, Extended Kohler's rule of magnetoresistance, *Phys. Rev. X*, 2021, **11**, 041029.
81 I. A. Sluchinskaya and A. I. Lebedev, An experimental and theoretical study of Ni impurity centers in Ba0. 8Sr0. 2TiO3, *Phys. Solid State*, 2017, **59**, 1512–1519.
82 Q. Zhang, B. Wang, V. S.-J. Huang, J. Zhang, Z. Wang, H. Liang, C. He and W. Zhang, Document parsing unveiled: Techniques, challenges, and prospects for

structured information extraction, *arXiv*, 2024, preprint, arXiv:2410.21169.
83 Y. Katsura, M. Kumagai, T. Kodani, M. Kaneshige, Y. Ando, S. Gunji, Y. Imai, H. Ouchi, K. Tobita and K. Kimura, Data-driven analysis of electron relaxation times in PbTe-type thermoelectric materials, *Sci. Technol. Adv. Mater.*, 2019, **20**, 511–520.
84 A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin and J. Clark, Learning transferable visual models from natural language supervision, in *International conference on machine learning, PmLR*, 2021, pp. 8748–8763, https://proceedings.mlr.press/v139/radford21a.
85 Z. Gan, L. Li, C. Li, L. Wang, Z. Liu and J. Gao, Vision-Language Pre-Training: Basics, Recent Advances, and Future Trends, *Foundations and Trends® in Computer Graphics and Vision*, 2022, **14**, 163–352.
86 pymupdf, pymupdf/PyMuPDF, https://pymupdf.readthedocs.io/, accessed 2024-12-01 16:03:06.
87 F. Pezoa, J. L. Reutter, F. Suarez, M. n. Ugarte and D. Vrgoč, Foundations of JSON schema, in *Proceedings of the 25th international conference on world wide web*, International World Wide Web Conferences Steering Committee, 2016, pp. 263–273, DOI: 10.1145/2872427.2883029.
88 obrink/chemdataextractor - Docker Image | Docker Hub, https://hub.docker.com/r/obrink/chemdataextractor, accessed 2025-03-31 19:19:09.
89 TheBloke/Llama-2-13B-chat-GGUF · Hugging Face, https://huggingface.co/TheBloke/Llama-2-13B-chat-GGUF, accessed 2024-11-26 20:13:03.
90 bartowski/Llama-3.1-Nemotron-70B-Instruct-HF-GGUF · Hugging Face, https://huggingface.co/bartowski/Llama-3.1-Nemotron-70B-Instruct-HF-GGUF, accessed 2024-11-26 20:09:19.
91 bartowski/Qwen2.5-14B_Uncensored_Instruct-GGUF · Hugging Face, https://huggingface.co/bartowski/Qwen2.5-14B_Uncensored_Instruct-GGUF, accessed 2024-11-26 20:12:21.
92 G. Gerganov, ggerganov/llama.cpp, https://github.com/ggerganov/llama.cpp, accessed 2024-11-26 12:16:56.
93 nomic-ai/nomic-embed-text-v1.5 · Hugging Face, https://huggingface.co/nomic-ai/nomic-embed-text-v1.5, accessed 2024-12-16 01:49:36.
94 BAAI/bge-m3 · Hugging Face, https://huggingface.co/BAAI/bge-m3, accessed 2024-12-16 01:53:22.
95 D. L. Olson and D. Delen, *Advanced data mining techniques*, Springer Science & Business Media, 2008.
96 C. J. van Rijsbergen, F. Crestani and M. Lalmas, *Information Retrieval: Uncertainty and Logics: Advanced Models for the Representation and Retrieval of Information*, Springer Science & Business Media, 2012.
97 R. Baeza-Yates, B. Ribeiro-Neto, *et al.*, *Modern information retrieval*, ACM Press, New York, 1999.
98 S. A. Hicks, I. Strümke, V. Thambawita, M. Hammou, M. A. Riegler, P. Halvorsen and S. Parasa, On evaluation metrics for medical applications of artificial intelligence, *Sci. Rep.*, 2022, **12**, 5979.
99 R. Rak, R. T. Batista-Navarro, A. Rowley, J. Carter and S. Ananiadou, Text-mining-assisted biocuration workflows in Argo, *Database*, 2014, **2014**, bau070.
100 S. R. Jonnalagadda, P. Goyal and M. D. Huffman, Automating data extraction in systematic reviews: a systematic review, *Syst. Rev.*, 2015, **4**, 78.
101 L. Schmidt, A. N. F. Mutlu, R. Elmore, B. K. Olorisade, J. Thomas and J. P. Higgins, Data extraction methods for systematic review (semi) automation: Update of a living systematic review, *F1000Research*, 2025, **10**, 401.
102 L. Li, L. Sleem, N. Gentile, G. Nichil and R. State, Exploring the Impact of Temperature on Large Language Models: Hot or Cold?, *Procedia Computer Science*, 2025, **264**, 242–251.
103 J. Kamp, L. Beinborn and A. Fokkens, Dynamic Top-k Estimation Consolidates Disagreement between Feature Attribution Methods, in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Singapore, 2023, pp. 6190–6197, DOI: 10.18653/v1/2023.emnlp-main.379.
104 Y. Huang, L. Zhang, H. Deng and J. Mao, Data-Driven Machine Learning Interface to Accelerate Material Design, *J. Chem. Inf. Model.*, 2024, **64**, 6477–6491.
105 T. Gupta, M. Zaki, N. A. Krishnan and Mausam, MatSciBERT: A materials domain language model for text mining and information extraction, *npj Comput. Mater.*, 2022, **8**, 102.
106 Y. Tang, W. Xu, J. Cao, J. Ma, W. Gao, S. Farrell, B. Erichson, M. W. Mahoney, A. Nonaka and Z. Yao, MatterChat: A Multi-Modal LLM for Material Science, *arXiv*, 2025preprint, arXiv:2502.13107.
107 M. Günther, I. Mohr, D. J. Williams, B. Wang and H. Xiao, Late chunking: contextual chunk embeddings using long-context embedding models, *arXiv*, 2024, preprint, arXiv:2409.04701.
108 B. Sheng, J. Yao, M. Zhang and G. He, Dynamic Chunking and Selection for Reading Comprehension of Ultra-Long Context in Large Language Models, in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Vienna, Austria, 2025, pp. 31857–31876, DOI: 10.18653/v1/2025.acl-long.1538.
109 P. Verma, S2 chunking: a hybrid framework for document segmentation through integrated spatial and semantic analysis, *arXiv*, 2025, preprint, arXiv:2501.05485.
110 A. Asai, Z. Wu, Y. Wang, A. Sil and H. Hajishirzi, Self-rag: Learning to retrieve, generate, and critique through self-reflection, in *International Conference on Representation Learning*, ed. B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan and Y. Sun, 2024, pp. 9112–9141, https://proceedings.iclr.cc/paper_files/paper/2024/file/25f7be9694d7b32d5cc670927b8091e1-Paper-Conference.pdf.
111 S. Jeong, J. Baek, S. Cho, S. J. Hwang and J. Park, Adaptive-RAG: Learning to Adapt Retrieval-Augmented Large Language Models through Question Complexity, in *Proceedings of the 2024 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), Association for Computational Linguistics, Mexico City, Mexico, 2024, pp. 7036–7050, DOI: 10.18653/v1/2024.naacl-long.389.

112 Y. Yuan, M. A. Shabani and S. Liu, Embedding-Based Context-Aware Reranker, *arXiv*, 2025, preprint, arXiv:2510.13329.

113 M. R. Rezaei, M. Hafezi, A. Satpathy, L. Hodge and E. Pourjafari, At-rag: An adaptive rag model enhancing query efficiency with topic filtering and iterative reasoning, *arXiv*, 2024, preprint, arXiv:2410.12886.

114 W. Zhai, SAM-RAG: An Self-adaptive Framework for Multimodal Retrieval-Augmented Generation, in *2025 International Joint Conference on Neural Networks (IJCNN)*, 2025, pp. 1–8, DOI: 10.1109/IJCNN64981.2025.11227819.

115 Z. Yao, W. Qi, L. Pan, S. Cao, L. Hu, W. Liu, L. Hou and J. Li, SeaKR: Self-aware Knowledge Retrieval for Adaptive Retrieval Augmented Generation, in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Vienna, Austria, 2025, pp. 27022–27043, DOI: 10.18653/v1/2025.acl-long.1312.

116 J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le and D. Zhou, Chain-of-thought prompting elicits reasoning in large language models, *Advances in Neural Information Processing Systems*, 2022, **35**, 24824–24837.

117 X. Bai, S. He, Y. Li, Y. Xie, X. Zhang, W. Du and J.-R. Li, Construction of a knowledge graph for framework material enabled by large language models and its application, *npj Comput. Mater.*, 2025, **11**, 51.

118 X. Ma, J. Li and M. Zhang, Chain of Thought with Explicit Evidence Reasoning for Few-shot Relation Extraction, in *Findings of the Association for Computational Linguistics: EMNLP 2023*, Association for Computational Linguistics, Singapore, 2023, pp. 2334–2352, DOI: 10.18653/v1/2023.findings-emnlp.153.