communications materials

A Nature Portfolio journal

Article

https://doi.org/10.1038/s43246-025-00979-w

# Question Answering models for information extraction from perovskite materials science literature



Matilda Sipilä [1], Farrokh Mehryary[2], Sampo Pyysalo[2], Filip Ginter[2] & Milica Todorović[1]

Scientific text is a promising source of data in materials science, with ongoing research into utilising textual data for materials discovery. In this study, we developed and tested a Question Answering (QA) approach to extract material-property relationships from scientific publications. QA performance was evaluated for information extraction of perovskite bandgaps based on a human query. We observed considerable variation in results with five different large language models fine-tuned for the QA task. Best extraction accuracy was achieved with the QA MatSciBERT and F1-scores improved on the current state-of-the-art. QA also outperformed three latest generative large language models on the information extraction task, except the GPT-4 model. This work demonstrates the QA workflow and paves the way towards further applications. The simplicity and versatility of the QA approach all point to its considerable potential for text-driven discoveries in materials research.

New sustainable technologies and materials are urgently needed, and materials design plays a key role in their development. In inorganic materials, compositional engineering provides an effective route for designing new materials and tuning functional properties towards intended applications. One group of materials where this is especially utilised is perovskites. They exhibit a promising combination of properties for photovoltaic applications, including high power conversion efficiencies[1–3], which would allow sustainable energy generation[4–6]. The stability and functional properties of perovskites critically depend on the material composition.

The crystal structure of perovskites is characterised by the generic formula $ABX_3$, were A and X sites are typically populated by cations and anions and a metal occupies the B site. This allows a very broad range of element substitutions, where organic molecules can also serve as A-site cations (hybrid perovskites). An entirely new range of functional properties can be accessed via substitutional engineering or alloying, where multiple element substitutions can be implemented on A, B, or X sites. The past decade has seen a proliferation of compositional engineering studies of perovskites to tune their properties, which have resulted in a large number of publications across disciplines and research areas[7–10]. There is a need to consolidate known properties of perovskites across different disciplines. This could be achieved by extracting perovskite information from scientific literature with language processing tools.

Natural language processing (NLP) is a field that relies on computation and machine learning to process, generate and analyse human language. In recent years, considerable advancements in NLP have been achieved by transformer neural networks and language models[11]. They exploit transfer learning, where a model is first pre-trained with a large dataset to learn general relationships in text, and then fine-tuned further for a certain specific task. Within NLP, we focus on the common task of information extraction (IE): finding information of interest from non-structured textual sources automatically.

In materials science, NLP has been used to extract information from publications with named entity recognition (NER) and relation extraction (RE). The aim of NER is to identify entities of interest, such as names or properties of materials from text. With RE, the aim is to determine entities in text, but also their relationships to each other. These tasks can be carried out by supervised machine learning algorithms or rule-based methods. The supervised machine learning NER approach has been deployed to extract materials synthesis parameters[12], polymer names[13], general materials information[14], general solid state materials information, gold nanoparticle synthesis descriptions and doping procedures of materials[15] from literature. Similar models have been applied to RE to extract synthesis parameters[16]. For rule-based methods, the state-of-the-art is the multi-purpose toolkit ChemDataExtractor2 (CDE2)[17], which is capable of RE for materials science. It combines grammar-based parsing rules with the probabilistic Snowball algorithm and was used to create databases about battery materials[18], thermoelectric materials[19] and semiconductor bandgaps[20] among others.

Supervised learning methods in NLP require manual annotation of entities to train the models. Analogously, rule-based methods require manual input to construct the syntactic rules by which the information is extracted from text. These present difficulties in applying such methods to

[1]Department of Mechanical and Materials Engineering, University of Turku, Turku, Finland. [2]TurkuNLP, Department of Computing, University of Turku, Turku, Finland. e-mail: milica.todorovic@utu.fi

different properties and materials: expert knowledge and human effort are needed to retrain the model for each new purpose. Moreover, models tend to focus on processing single sentences[17,21]. This leads to a loss of information where the relation between entities crosses sentence borders.

Due to their recent advances, generative large language models (LLMs) have also gained popularity in the materials science community. The generative LLMs have been used to extract information about metal–organic framework synthesis[22], dopants and host materials, metal-organic frameworks and general materials information[23] and experimental materials science data[24]. While generative models offer many possibilities, their main shortcoming is their tendency to produce values or texts not found in the original text. This behaviour, called "hallucination", can lead to incorrect values and unreliable databases. In addition, the most widely used generative models, GPT-3.5[25] and GPT-4[26] are commercial, so their use in large-scale information extraction would be expensive.

Here, we explore the potential of language models inherently incapable of hallucination for IE in materials science. We use language models with the Question Answering (QA) approach. QA is based on the model's capability to exploit previous training and identify information that it has not specifically been trained to extract. Starting with a pretrained language model, one can fine-tune it with the general structure of questions and answers to produce a QA model. This tool can then be applied to different domain extraction tasks without the need for retraining, in an unsupervised manner, which sets it apart from the task-specific approaches of supervised machine learning or rule-based RE and NER. Trained QA models can return answers to a natural language question (human query) from a context document of arbitrary length, crossing sentence borders. It returns the text span that is most likely the correct answer; otherwise, it returns an empty string. While QA models are promising, this technique has not been applied to RE in materials science to date, and it is unclear if the QA performance would be satisfactory.

In this study, our objectives were to build a QA framework for extracting information from perovskite materials science literature, evaluate its performance and apply it to a large textual dataset. To implement QA, we considered model choices that range from selecting the question, materials, properties of interest and language models, to deciding what kind of context documents were most appropriate. The context documents are materials science literature segments, referred to as snippets throughout this manuscript. Snippets are computationally more efficient to use than full-text publications, which contain a lot of non-relevant text and complex contexts, and could possibly lead to retrieving erroneous information.

The task was to extract material-property relationships as [material, property, value, unit] with the question being 'What is the numerical value of the property of material X?'. A typical text snippet with the question and extracted answer is illustrated in Fig. 1. For the purposes of testing the model, the property of interest was the bandgap. Bandgap greatly affects the optoelectrical properties of perovskites, which is important for the solar cell research community. To demonstrate the method, we focused on five different perovskite materials: three hybrid (MAPI, MAPB and FAPI) and two inorganic halide ($CsPbI_3$ and $CsPbBr_3$) perovskites. The narrow focus allowed us to keep the initial study compact, evaluate the effect of different model choices and pave the way towards further applications.

A key question was which pre-trained language model to use as the basis of the QA model, since the base model may have a considerable effect on the model's performance. We compared five different transformer models, pre-trained with different datasets: base BERT (Bidirectional Encoder Representations from Transformers)[11], SciBERT[27] and three BERT models trained with materials science texts[15,28,29]. Each of the models was fine-tuned towards QA by training with the general domain dataset SQuAD2[30]. This state-of-the-art QA training dataset also contains empty answers, which allow the model to correctly detect cases where the snippet does not contain any relevant values. Although SQuAD2 is not targeted towards scientific questions, we relied on the generic capability of the QA models to answer the questions. The performance of QA systems with different BERTs was compared first to baseline method CDE2 and then to four generative models. The best QA model performance was obtained with QA MatSciBERT, and the F1-scores of all QA models surpassed those of the state-of-the-art tool CDE2. Among the generative language models, GPT-4-0613 (GPT-4) achieved the strongest overall performance.

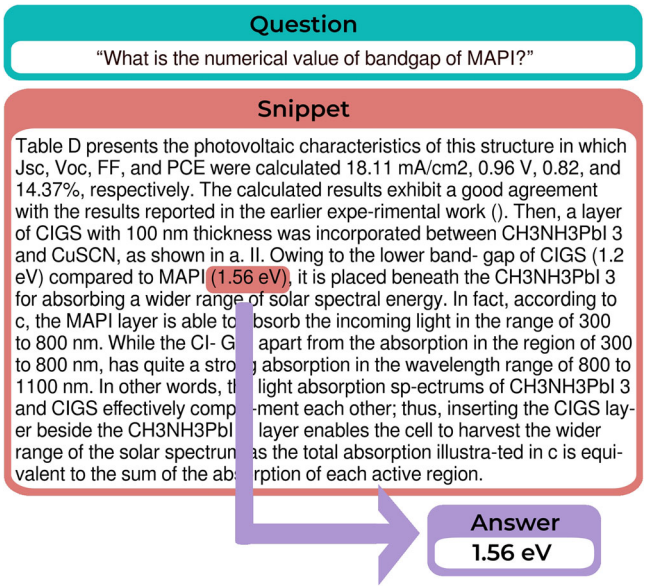


**Fig. 1 | An example of a question, snippet and answer.** The question is "What is the numerical value of the bandgap of material X?" and the answer contains both the value and the unit. The name 'CH3NH3PbI3' has been normalised in the snippet (extracted from ref. 74) to "MAPI".

## Results

The predictive power of QA for bandgap extraction was tested with respect to model choices, namely, different BERTs. We first address the model quality tests with different QA confidence thresholds. Performance metrics (precision, recall and F1-score) were evaluated on the annotated dataset for all five BERTs and compared to CDE2 and four generative model values. Next, all QA models were applied to limited-scale IE to examine the effect of model choices on the range the extracted values. Based on test results, we selected an optimal QA model and applied it to the complete corpus to demonstrate QA-based IE.

### Evaluating QA model performance

The predictive power of the QA models on the cross-validation (CV) validation dataset is illustrated in Fig. 2. Precision, recall, and F1-score were computed for all five BERTs as a function of confidence threshold, and compared to the CDE2 results. The values in Fig. 2 with their standard deviations are in SI Table S8. When the threshold increases, model precision rises, but recall decreases. This is expected behaviour, because high thresholds enforce model certainty, and simultaneously reduce the number of answers returned. Overall, the general BERT achieves the poorest outcomes due to the fewest answers returned (inferior recall).

The optimal F1-score of 61.3 was observed with QA MatSciBERT (at threshold 0.1), which can be associated with consistently high precision of this language model (Fig. 2b). In contrast, MatBERT produced consistently highest recall (Fig. 2c). MaterialsBERT was the least effective of the three materials language models and performed similarly to SciBERT on this task. We note that CDE2 favours precision over recall (by construction) and thus did not reach F1-scores beyond 45.6. QA models outperformed CDE2 on precision only at very high thresholds, but at the cost of much degraded recall and F1-scores. Throughout the different thresholds, MatSciBERT exhibited high standard deviations, which is evident from the data uncertainties in Fig. 2b, c. Elsewhere, the standard deviations are smaller, pointing to a reduced dependence on initial data in model training.

In Table 1, we observe that different confidence thresholds optimise the QA extraction for different BERTs. The top F1-score of 61.3 obtained with MatSciBERT and a threshold of 0.1 is closely followed by an F1-score of 58.6 for MatBERT (threshold 0.2). The F1-score range of 54–57 indicates a similar performance of SciBERT and MaterialsBERT, but BERT falls short with an F1-score of 47.5. Most language models achieved their best F1-scores at similar values, 0.05–0.2 in confidence. This finding was clarified once we computed the average confidence scores of the top answers (Table S10 in SI). While the average first answer confidence score was 0.7, the average second answer confidence score was as low as 0.06. It follows that simply selecting the top answer could produce good extraction, without the need to consider confidence score thresholds. To test this, we calculated the evaluation metrics when selecting only the one top answer from the QA results (see Table S9 in the SI). The best F1-score obtained with this approach was 60.7 with MatSciBERT. The results indicate that selecting just one top answer (the recommended QA approach) produces F1-scores that are very similar to the best threshold results. The only exception was the BERT QA model, where the low confidence score threshold produced a higher F1-score.

**Table 1 | QA model performance on the test set**

| | T | F1 | P | R |
|---|---|---|---|---|
| BERT | 0.025 | 47.5 (±3.4) | 46.0 (±8.2) | 51.0 (±6.5) |
| SciBERT | 0.2 | 56.7 (±2.6) | 55.8 (±3.1) | 58.1 (±5.0) |
| MatBERT | 0.2 | 58.6 (±4.9) | 58.1 (±6.2) | 59.1 (±7.1) |
| MaterialsB. | 0.05 | 54.6 (±4.4) | 50.4 (±5.5) | 61.0 (±8.8) |
| MatSciB. | 0.1 | 61.3 (±4.0) | 64.5 (±5.1) | 59.6 (±9.3) |

Evaluation metrics F1-score (F1), precision (P) and recall (R) for each QA model with the best validation set threshold (T). Results are reported as mean and standard deviation. MaterialsB. denotes MaterialsBERT and MatSciB. MatSciBERT.

The performance metrics indicated differences between the BERTs tested on the annotated dataset (600 snippets). Still, it is unclear to what extent this influences IE in real-world applications, with thousands of context documents. To explore the scale of this effect, we applied the 5 QA language models and CDE2 to our full dataset, and extracted the values based on the best thresholds defined in Table 1. We focused on the two materials with the most and least snippets available: MAPI had extensive literature coverage (7283 snippets), while FAPI was the least featured (1251 snippets). Here, we expect both precision and recall to play a role in the number of extracted values and their distribution. On average, it required 2.7 h and 0.5 h to extract MAPI and FAPI values with QA models, and 3.9 h and 0.7 h with CDE2. This is expected because CDE2 is designed to extract band gaps for all materials present in the snippets, rather than limit the analysis to only MAPI and FAPI.

The bandgap distributions presented in Fig. 3 were subject to statistical analysis for the number of extracted values (Table S12 in the SI). In the case of MAPI, all models recorded the mode of 1.55 eV and

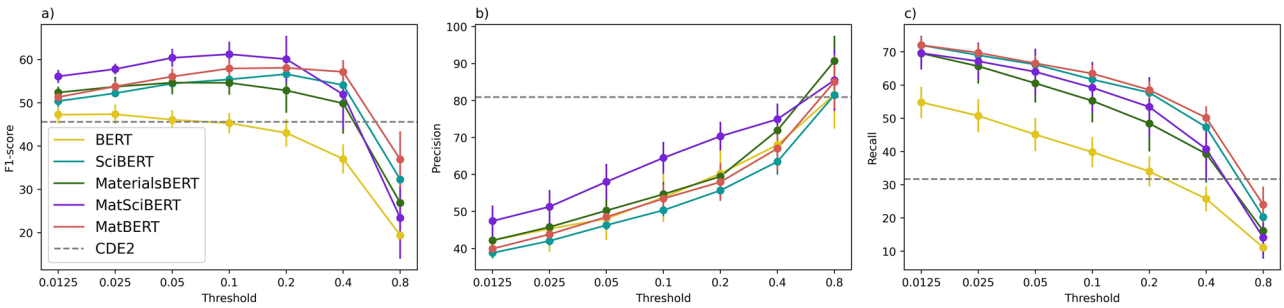


**Fig. 2 | IE evaluation metrics by confidence score threshold.** F1-score (**a**), precision (**b**), and recall (**c**) of the CV validation set answers, returned by different language models trained for the QA task. Error bars denote the standard deviation from the results of the same models trained with different seeds. The grey dashed line denotes the CDE2 performance, with a 45.6 F1-score, precision at 80.9 and recall at 31.7.

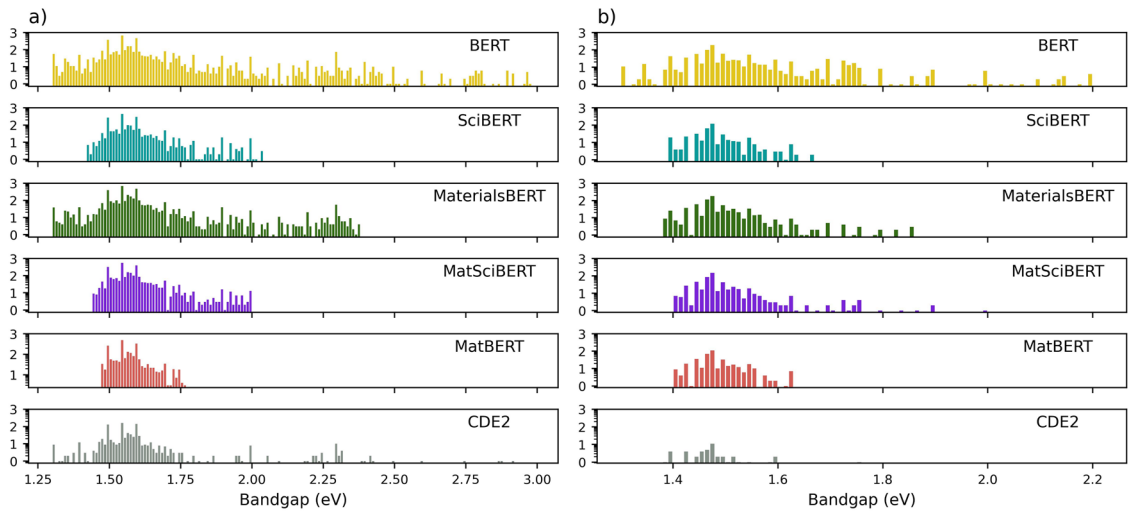


**Fig. 3 | Statistical distributions of extracted MAPI and FAPI bandgap values.** Histograms for (**a**) MAPI and (**b**) FAPI hybrid perovskites, using QA models based on BERT, SciBERT, MaterialsBERT, MatSciBERT, and MatBERT. The *y*-axis was modified by $\log_{10}$ for clearer comparison. The CDE2 values were post-processed using the CDE2 postprocessing script from previous work[20]. The x-axis range matches the broad value range of BERT QA answers, excluding some CDE2 values (97 for MAPI and 5 for FAPI).

median of 1.56 or 1.57 eV. Among QA models, base BERT band gaps were most diverse, with a range of 1.70 eV, while MatBERT returned a narrow range of 0.28 eV. A broad range suggests many retrieved values: QA BERT extracted 3995 MAPI bandgap values. For FAPI, we observed similar trends: QA BERT obtained the maximum of 993 bandgap results in contrast to QA MatBERT with 397 values. This was reflected in their ranges of 0.93 eV and 0.22 eV, respectively. The FAPI mode and median were 1.48 eV for all the models except BERT, where the median of 1.49 eV was affected by the large range of values.

While a large number of retrieved values is always desirable, an overly broad range of extracted band gaps may not indicate quality. Large band gaps over 2 eV identified for MAPI by BERT and MaterialsBERT are unlikely and indicate a lack of precision in extraction. MatSciBERT and MatBERT exhibited the best performance in this test. Here, we additionally clarified if the extremes of the distribution ranges contained correct bandgap values from the text or erroneous answers. We manually selected 10 random QA MatBERT retrieved MAPI band gaps from the high range (above 1.7 eV) and low range (below 1.49 eV), and inspected the original snippets. In all 10 examples, the extracted values were correct band gaps, but in two cases, MAPI had been doped by bromine, which resulted in a bandgap shift. The latter observation made clear that many scientific factors may produce a genuine difference in band gaps reported in the literature.

Based on the performance evaluation tests, we opted for the MatSciBERT in all subsequent IE tasks. It exhibited the highest F1-score in Table 1. Based on Fig. 3 and SI Table S9, MatSciBERT extracted much information (2733 MAPI band gaps) from the second most narrow numerical range, allowing less scope for error.

To evaluate how QA compares with state-of-the-art tools, we applied four generative models to the same information extraction task with the whole evaluation set (600 snippets). The metrics are summarised in Table 2 and indicate that F1-scores achieved by generative models varied greatly. The optimal F1-score of 68.6 was obtained with the GPT-4-0613 (GPT-4), improving upon GPT-3.5 Turbo (GPT-3.5) and surpassing the QA MatSciBERT results. In contrast, two popular high-quality models, Mixtral-8 × 7B-Instruct-v0.1 (Mixtral-8) and Llama3-ChatQA-1.5–8B (Llama3), exhibited lower F1-scores on account of poor precision. This observation is explained by the high number of hallucinated answers for Llama3 (87) and Mixtral (35), which are considered false positives and lower precision. Generally, the recall of generative models is similar to or better than that of the QA models. Data suggests that only the paid GPT-4 model surpasses the QA MatSciBERT on information extraction tasks. Additionally, we inspected the three hallucinated values from GPT-4 (different executions yielded a maximum of 3 hallucinated values). The first hallucinated value was found to be "2.39 eV". It originates from the sentence "Above bandgap photoexcitation was performed using a Duetto laser at a photon energy of 3.49 eV (i.e. 1.1 eV above the direct bandgap excitation)"[31], where it appears that GPT-4 had calculated the bandgap value to be 3.49 eV–1.1 eV. So it seems that the value was correctly calculated even though it was not directly stated in the article. In the second case, the bandgap originates from the bandgap value 1723 meV, which the GPT-4 model had converted to 1.723 eV[32]. The last value originates from the $CsPbBr_3$ bandgap value of 19,000 $cm^{-1}$[33], which GPT-4 had converted to 1.49 eV. This behaviour could be mitigated through more extensive prompt engineering in the future.

### Information extraction of perovskite bandgap values from the full dataset

We applied QA MatSciBERT to extract bandgap values for five perovskite materials: three hybrid and two inorganic ones. The number of snippets (Ns) and the number of extracted values (Nv) are presented in Table 3. The

**Table 2 | Comparison of model performance between generative models, CDE2 and a QA model**

| | F1 | P | R | H |
|---|---|---|---|---|
| Mixtral-8 | 58.3 | 46.9 | 77.1 | 35 |
| Llama3 | 45.7 | 33.6 | 71.4 | 87 |
| GPT-3.5 | 56.9 (±0.6) | 44.6 (±0.6) | 78.5 (±0.6) | 12.4 (±1.2) |
| GPT-4 | 68.6 (±1.7) | 56.8 (±4.7) | 87.7 (±4.6) | 2.0 (±0.6) |
| QA | 61.3 (±4.0) | 64.5 (±5.1) | 59.6 (±9.3) | 0 |
| CDE2 | 45.6 | 80.9 | 31.7 | 0 |

F1-score (F1), precision (P), recall (R) and number of hallucinated answers (H) from the generative models, CDE2 and the best QA model, MatSciBERT (QA).

**Table 3 | Statistics describing the bandgap values extracted from the full dataset**

| Material | Ns | Nv | Mode (eV) | Refs. |
|---|---|---|---|---|
| MAPI | 7283 | 2,733 | 1.55 | 59–61 |
| MAPB | 1646 | 641 | 2.30 | 62–64 |
| FAPI | 1251 | 536 | 1.48 | 65–67 |
| $CsPbI_3$ | 1734 | 1128 | 1.73 | 68–70 |
| $CsPbBr_3$ | 2029 | 1125 | 2.30 | 71–73 |

Number of extracted snippets (Ns), MatSciBERT extracted values (Nv), mode of extracted values after postprocessing and literature references for comparing the extracted values.

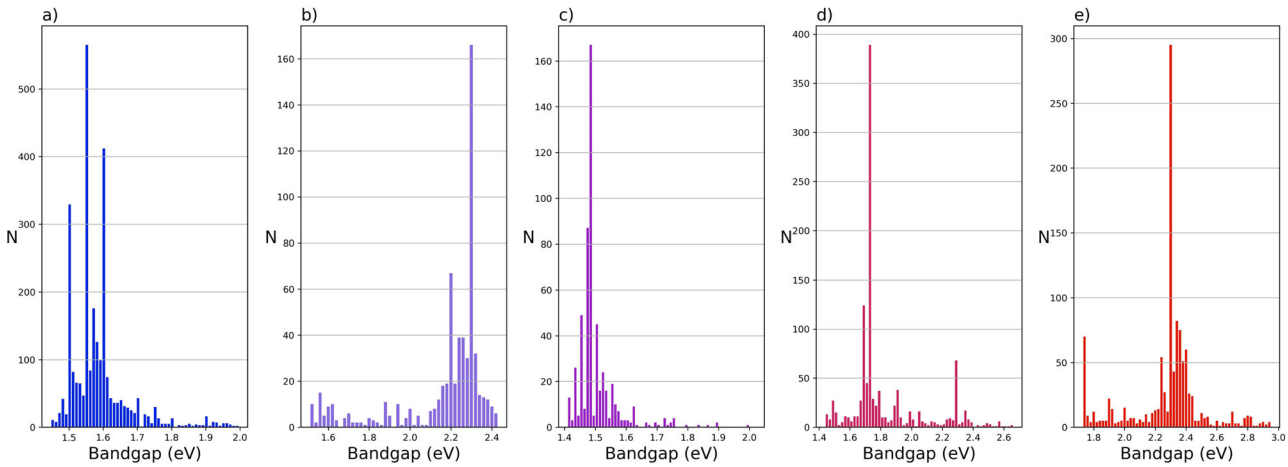


**Fig. 4 | Statistical distributions of bandgap values extracted with MatSciBERT.** Histograms for (**a**) MAPI, **b** MAPB, **c** FAPI, **d** $CsPbI_3$ and **e** $CsPbBr_3$, showing the extracted bandgap values.

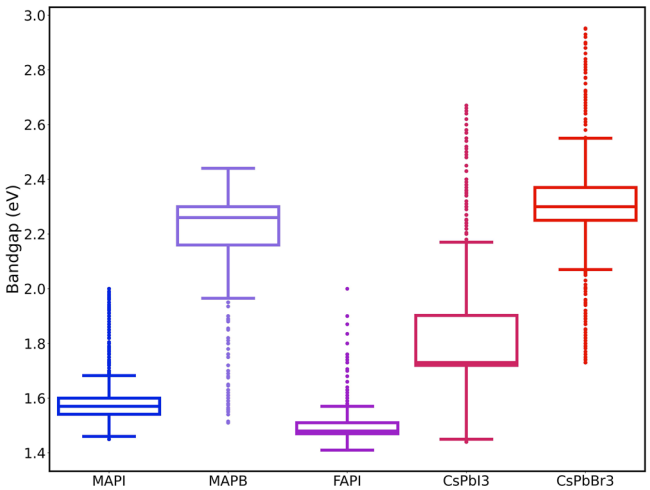


**Fig. 5 | Comparison of the bandgap distributions extracted for different perovskite materials.** The line inside the box denotes the median of the distribution. It divides the box into the upper quartile 25% (upper half), and lower quartile 25% (lower half). The whiskers mark the remaining two 25 % quartiles, and the dots indicate outliers.

resulting distributions are depicted in histograms in Fig. 4. We first verified the overall accuracy of the extracted data. The mode of the extracted values agrees well with the experimental values found from the literature sources (Table 3). MAPI data featured higher peaks at 1.5 and 1.6 eV in Fig. 4a (MAPI), which can be explained by the trend of researchers reporting values in two significant digits.

The statistical bandgap distributions for different materials are compared in Fig. 5. Statistical analysis can be found in Table S13 of the SI. Most of the observed distributions featured a broad spread and data bias. Bias towards lower values and a lack of low outliers were observed for the hybrid perovskites (MAPI and FAPI). The behaviour for inorganic perovskites was less clear: $CsPbI_3$ and $CsPbBr_3$ bandgap data presented long tails towards high bandgap values.

The range of extracted bandgap values could vary considerably. Overall, the ranges for hybrid perovskite band gaps were smaller than those for the inorganics. We compared distribution spreads to the number of extracted values in Table 3 to establish if more values simply lead to broader distributions. However, this was not the case, with MAPI registering both a large amount of extracted data and a narrow spread. As previously noted, there could be many factors behind the spread of extracted values, and this may not be a concern in database applications as long as the average values are accurate.

## Discussion

We had set out to explore the potential of QA models for IE in materials science, given the task to extract perovskite band gaps from literature. We found the QA models to be straightforward to implement, which is an advantage. The only model training required was with the general QA dataset, which was easy and fast. They are also versatile and could be directed towards specific knowledge domains via different base BERTs.

The different tests demonstrated that QA models are capable of extracting bandgap values from materials texts with good accuracy. Optimising confidence score thresholds with CV produced higher accuracies than just extracting the one answer with the greatest confidence score, which is common practice in QA[34–36]. This finding likely arises from multiple bandgap values found in several single snippets. The QA models are designed to extract only one value from text: here, the threshold parameter is not used, and no tuning is required[34–36]. This study highlights the occasional need to extract multiple values. In such complex IE cases, a careful approach to QA threshold optimisation is recommended for best model performance.

Varying the threshold would also allow tuning QA models to emphasise either precision or recall, depending on the task at hand. In addition to threshold optimisation, we calculated the F1-scores of different models for the top first answer. The relatively small differences between these results and the optimised threshold results can be explained by the small fraction of the snippets with more than one bandgap value in our evaluation dataset.

The choice of the QA language model was most important for the accuracy of information retrieval. In our tests, base BERT was least optimal, most likely because it had not been trained with domain literature in materials science. We expected the three BERTs to perform the best, but here we also observed differences. The F1-scores of QA MaterialsBERT were slightly lower, and it extracted a broader range of numerical values in application tests, which could lead to erroneous answers. QA MatBERT values exhibited the shortest range in the application test, but its F1-score was not as high as that of QA MatSciBERT, which also performed well on the extraction test.

The differences between QA models can be explained by the different training sets of the base language models. MatBERT was trained with 2 million publications from a broad materials science domain, which is much more text than the 2.4 million materials science abstracts used to train the MaterialsBERT. Also, abstracts can lack some concepts or words which full-text articles contain. MatSciBERT training set was smaller (153,978 publications), but the underlying SciBERT may have produced the high F1-score of the model by contributing the capability to process scientific text. Further tests are needed to establish if the comparative BERT performance carries over to different QA extraction tasks and datasets.

It was surprising to discover that the F1-scores of all five BERT language models exceeded the CDE2 baseline. We note, however, that CDE2 was optimised towards a different IE task: to retrieve all material properties for all materials found in texts. This may render it less sensitive to a particular property and material, a task made specific by the textual query we used to guide QA extraction. This also limits the QA procedure: the user must identify the material of interest in advance, as the algorithm does not extract values for all materials in the snippets like CDE2 does. The performance gap could be due to another practical difference: although CDE2 operated on entire snippets, it was designed to extract information from single sentences only. Text search analysis revealed that all necessary information was contained in one sentence for only 40.4% of the snippets. That might explain the lower overall F1-scores observed for CDE2. In precision of extracted values, CDE2 still outperformed the QA models.

There are substantial differences in the performance of generative models, as evident from Table 2. Unlike the models used through OpenAI generation API, Mixtral-8 and Llama3 models allow full control over the result generation and exhibit deterministic behaviour, but the F1-score of Llama3 falls short. Deterministic behaviour is desirable in IE tasks for reproducibility purposes. In terms of F1-score, the GPT-3.5 model performs similarly to the QA models, but cannot reproduce answers. The GPT-4 model F1-score exceeded the QA MatSciBERT F1-score at 68.6 to 61.3, which highlights the power of one of the largest generative models. Such a difference in performance is expected because, compared to BERT-based models, GPT models utilise much larger datasets in training. They benefit from massive scaling laws (e.g. more parameters, more data, longer context windows) and reinforcement learning from human feedback. As a result, they achieve stronger performance across both comprehension and generation tasks, making them far more versatile than BERT models. However, as long as the GPT-4 model is not open access and requires payment, the gain in terms of F1 would need to be weighted against the costs for very large-scale runs, where the comparatively light-weight QA model might have an advantage. The tendency to hallucinate answers is more evident with Mixtral-8 and Llama3, but even a small fraction of hallucinated answers (GPT-3.5 and GPT-4) is undesirable. In future work, it will be important to investigate whether generative models specifically fine-tuned for information extraction (IE) tasks[37] can reduce hallucinations and improve overall performance.

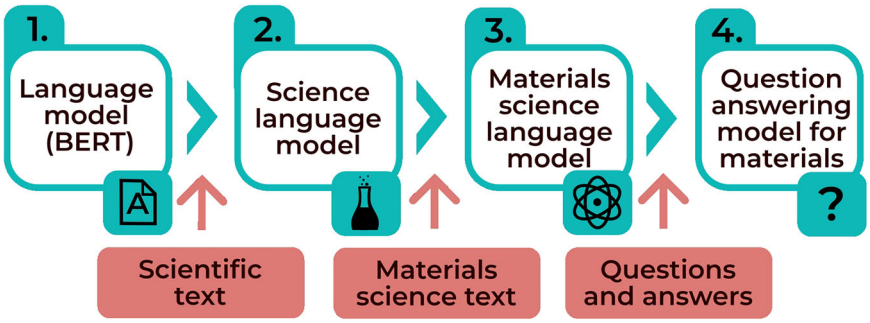


**Fig. 6 | Workflow for implementing and testing the QA method for an IE task.** The textual data is first obtained by downloading it from the APIs and scraping it from the web. The dataset is then converted into plain text, verified to contain unique publications and formatted into snippets. In the third step, a suitable machine learning model is selected and applied to the snippets. Eventually, the extracted information is postprocessed, used for model evaluation and analysed.

As the results above demonstrate, the QA method is capable of extracting bandgap values without being explicitly trained for this task. This presents an attractive alternative to more established machine learning methods for IE, where a large amount of manually annotated text is typically required to retrain models for each new IE task. While QA approaches demonstrate good performance in materials property extraction, their application to extracting more complex relationships (such as synthesis conditions) might be limited by the maximal token input lengths for context documents. Further method development on sequential extraction with multiple items would be needed to address these current limitations. From the end user point of view, QA models require only the question prompt and the context documents, which makes them more accessible to non-expert users.

Information extraction of materials properties inevitably results in a numerical distribution. With unsupervised use of QA, it is important to critically evaluate the results. The extracted values might deviate from the median for multiple reasons: experimental measurement conditions may vary, old references may contain incorrect measurement values, and computational results differ from the experimental ones. Previous work[38] has established that the computational bandgap of $PbI_2$ varies from 1.4 eV to nearly 3.6 eV depending on the computational approach. Since $PbI_2$ forms part of MAPI and CsPbI3, the computational bandgap results might vary considerably. A broad distribution does not necessarily mean poor performance. Still, if the range is very wide, there is no means to differentiate the origin of the values without supervision, so it is recommended to favour QA models that produce a narrower range. While our demonstrated application was limited in scope to five materials, it indicates the predictive power of the QA approach and will serve to guide further extraction work.

QA models seem to derive additional performance from the context found in surrounding texts. This could be attributed to the latent knowledge conferred by their base models. While this is an advantage, working with snippets can also present an application bottleneck. In this study, snippet generation was specific to the material name and the property of interest (bandgap). This step was designed to remove any irrelevant textual information; consequently, a search for any other materials property with the same snippet corpus would have been inappropriate (and not as accurate). Future work could also evaluate the impact of source text type (e.g. published book chapters, theses or manuscripts) on the quality of IE, so this information should be recorded in the metadata of further benchmark datasets. Applying QA models to other extraction tasks may not require retraining the models, but it would require generating new snippets from the full corpus. While IE performs well on bandgap properties, partly owing to our refined workflow, IE tests on additional properties are required to evaluate the generalisability of this tool. However, further quantitative tests are made challenging by the need for exhaustive synonym lists[39] as well as human-annotated benchmark datasets for multiple material properties, which are currently lacking in the community. In future studies, snippet generation with the same Elasticsearch approach could be automated to require minimal human involvement. Additionally, several aspects of the workflow could be parallelised for greater computational efficiency.

This study demonstrates the suitability of the QA method for extracting material-property relationships from materials science literature. QA-based IE of bandgap values was accurate and efficient with materials science literature snippets and reached F1-scores above the state-of-the-art. Moreover, QA allows extraction by human language queries, which would lower the barriers for non-expert users and promote diverse applications. Further tests are needed to explore the performance of QA tools on different tasks. The versatility of the approach indicates considerable potential of QA for NLP applications in the field of materials science, which could be deployed to accelerate discovery and materials design.

## Methods

Figure 6 illustrates the general workflow of our implementation of the QA method for IE. First, the textual data, i.e. corpus of scientific publications, was downloaded via publisher application programming interfaces (APIs) and web scraping. Next, this textual data, which was obtained in multiple formats, was converted to plain text, and duplicated text was removed. After dividing the text into snippets, we selected a suitable QA model and applied it to the selected snippets. Finally, the QA extracted values were post-processed, used to evaluate the model and analysed.

### Data acquisition

The first step in IE was to create a dataset of scientific publications. We downloaded publications from five sources: arXiv API[40], Elsevier Article Retriever API[41], Core API[42,43], Springer Nature API[44] and also web scraped the Royal Society of Chemistry (RSC) articles[45]. Permission to download publications was granted through the university licences, but permission to scrape publications through the RSC webpage was obtained directly from the RSC. We used the query "perovskite" or "perovskites" when downloading and web scraping publications: if a publication matched a word, it was added to the corpus.

We downloaded a total of 238,431 scientific publications from different sources combined. The number of articles obtained from each source is summarised in Table 4. In addition to full texts, we collected the following metadata: year of publication, source and article identifier. In most cases, the article identifier was the digital object identifier (DOI), but 1161 publications obtained via the arXiv API lacked DOIs, so they were labelled by the `arXiv id`. Similarly, 12,843 articles from the Core API lacked a DOI and were labelled by the `Core id`.

### Data processing into snippet text segments

Data processing involved three stages: converting data to plain text, removing duplicate publications and formatting snippets. We converted the scientific publications to plain text because language models cannot process PDFs, HTML or XML tags. We also removed all figures and all the tables marked with tags from the texts, since these cannot be processed as normal

**Table 4 | Summary of text sources used to compile the corpus**

| Source | Format | Number | Conversion tool |
|---|---|---|---|
| Elsevier Article Retrieval API | XML | 117,044 | CDE2 ElsevierXmlReader[17] |
| Springer Nature API | XML | 34,395 | Conversion script |
| Core API | Json | 65,924 | Text selected by json-tag |
| arXiV API | PDF | 3814 | CDE2 Document package[17] |
| Royal Society of Chemistry | HTML | 17,254 | CDE2 RscHtmlReader[17] |

File formats of publications from different sources, the number of downloaded texts and the conversion tools employed.

text and were out of the scope of this research. There are also currently no satisfactory methods for extracting text from figures, although there are several methods for extracting text from tables[46,47]. Figure captions were preserved. Any supplementary information located in the main manuscript file was included (some Core and arXiv publications). Where the supplementary information was placed in external files, these were omitted from consideration (all other publishers). Converting different publication formats to plain text is a challenging task because the formats of the publications vary, so we used multiple software tools. Table 4 describes the original format of publications and summarises the range of conversion tools employed. Here, 89 articles could not be converted with the available tools and were omitted. After conversion, the plain text dataset totalled 195,872 publications.

It is important that the dataset contains only unique publications to avoid redundancy of extracted values. By using DOI as a unique identifier for publications, we identified and removed 1,505 duplicate manuscripts. The articles without a DOI were checked against the entire dataset: we converted all plain-text articles to numerical vectors of token counts and compared the vector representations to each other. This allowed us to remove 45 publications without a DOI, where the token count vectors were near-identical to another publication. The conversion process and duplicate removal are described in detail in Section S1 of the Supplementary Information (SI). After removing duplicates, there were 194,322 unique scientific publications in the dataset.

The plain text articles were segmented into text snippets that contained valuable information. To keep the IE efficient and accurate, the remainder of the manuscript text was discarded at this stage. First, all the articles bearing this information of value were identified. We used the Elasticsearch search engine[48] to query articles by keywords. The keyword list contained the name of the material of interest, the word "bangap", and their multiple synonyms and acronyms (see Table S2 in SI). From the entire corpus, we retained for further processing only the manuscripts which contained the keywords for material name and property. For example, with the material MAPI, we selected 11,193 publications with bandgap information. This step is specific to the material name and the property of interest and was repeated for each material.

The snippet approach was adopted to purge the text of information irrelevant to the task, such as introductory passages, discussions of other materials or references. Based on preliminary experiments, we set the length of a snippet to seven sentences. Section S2 in SI describes related information analysis. While most of the information could be found within one sentence (40.4% of cases), a substantial portion of snippets had information spanning two (17.8%), three (12.7%) or more (29.1%) sentences. The seven-sentence snippet thus ensures more context than a single sentence and allows for the extraction of also information divided between multiple sentences, although using snippets instead of single sentences somewhat increases the computational requirements of the extraction method. The BERT models have a capability to process texts up to 512 tokens in lenght without loss of information, and 7 sentence snippets usually fall below this limit. We carried out tests to confirm this (details in SI table S4).

To build the snippets, we had to ensure they contained at least one mention of the material name (for example, "MAPI"), the property name ('bandgap') and the unit name ("eV"). Because the relevant terminology varies, we normalised the text by standardising pertinent nouns to one chosen form. Here, the previously-constructed list of synonyms was used to convert all materials and property names to, e.g. "MAPI" and "bandgap". For each article in the corpus subset, we extracted seven-sentence snippets which contained mentions of material, property and unit. The snippets were extracted by moving a seven-sentence text window down the manuscript text and saving the snippets which contained the necessary information (see SI Section S2). For example, for MAPI material, we extracted 7281 snippets from the 11,193 articles parsed.

### QA model implementation

The state-of-the-art methodology in the field of NLP is transformer neural networks based on self-attention[49]. This enables the language model to learn relationships between words and the context of the word, regardless of the text length. In this study, we opted for the encoder-only transformer model, since it is intended for analysing textual data (decoder-only and encoder–decoder models are designed for text generation). A major advantage of encoder-based models in IE is their inability to hallucinate answers, which makes them a valid choice in tasks where the answer is directly a substring of the snippet[11].

Transformers rely on transfer learning to acquire new model knowledge and adapt to more specific tasks. This approach allows models, which learned useful features, contexts or patterns from a previous dataset, to retain previous information and add to it with further training. Figure 7 illustrates the process of training a base language model in stages with a carefully selected set of domain-specific texts. It is common to start with the generic encoder-only language model BERT, trained on general-domain English language texts, such as English Wikipedia, books and text scraped from multiple web sources[11]. This model can be further trained with scientific texts to generate a science English language model, which can then be trained with materials science texts. In our approach, the final language model was further trained for question answering tasks to produce the QA model.

Different base models may have a profound impact on the performance of the QA model. Since there is no literature precedent, we tested five different language models: two general ones and three materials science BERTs. We tested the base BERT model, trained with general-domain English language text[11], and the scientific language model SciBERT, base BERT trained with scientific text from the computer science and biomedical domains[27]. The materials science models were MatBERT[15], MatSciBERT[29] and MaterialsBERT[28]. MatBERT was trained from base BERT with randomly sampled two million documents, mostly consisting of peer-reviewed materials, scientific journal articles[15]. On the other hand, MatSciBERT was trained from SciBERT[27] with 153,978 scientific publications about inorganic materials. MaterialsBERT was trained based on PubMedBERT[50] with 2.4 million scientific abstracts from multiple materials science sub-domains.

Following the standard practice in the application of encoder models to span retrieval tasks, we fine-tuned the above language models for the QA task with the SQuAD2[30]. During fine-tuning, the BERT-based model is assigned two probabilities to each token in the context document, indicating the likelihood of it being the start or end of the answer. The text span between the two tokens that are most likely to start and end the answer is returned as the QA answer[11]. If there is no answer, the beginning and ending token as predicted to the [CLS] token, which is prepended to all inputs in the model. Computational implementation was based on the Huggingface transformers-library[51] version 4.20.1 and Pytorch version 1.12[52]. All computing was performed on the NVIDIA Volta V100 GPU. The SQuAD2 features 107,785 question-answer pairs, as well as 53,775 unanswerable questions, on 526 Wikipedia articles. Section S3 in SI summarises the hyperparameter tuning procedure. In applications, we accounted for statistics by using 5 different seeds for weight initialisation in the training process to build and test 5 QA models for each BERT. The average values

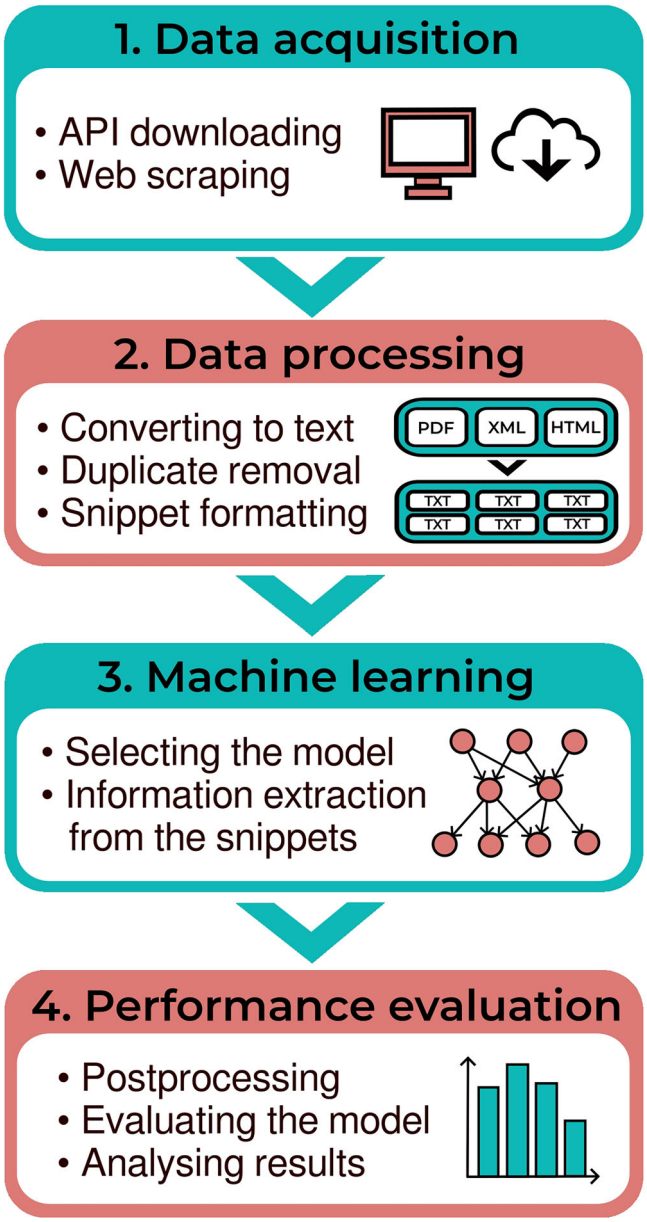


**Fig. 7 | A schematic figure of transfer learning that underpins the QA model for materials science.** The base BERT language model is trained first with scientific text, then with materials science text and finally fine-tuned with questions and answers to form the QA model for materials science.

and standard deviation were reported as the final performance metrics. The trained QA model, when provided with a question, generates a number of answers with a model confidence score associated with each one. By default only the top answer is retrieved since it is most likely to be correct (as illustrated in Fig. 1), but the workflow can be adjusted to retrieve multiple answers. Should the correct value be missing in the text, we would expect an empty string to be returned, based on SQuAD2 training.

### Performance evaluation

Before applying the QA model to the complete dataset, we benchmarked the performance of the different language models in the QA framework to identify the best-performing BERT. For this, it was necessary to postprocess model answers, evaluate suitable performance metrics and visualise the retrieved data.

In applied NLP, answers are typically postprocessed to standardise them and facilitate the computation of performance metrics. In response to our query prompt "What is the numerical value of the bandgap of material X?", the model could return a number of answers. We expected the correct answer to contain the correct numerical value of the bandgap and the corresponding unit "eV", but text answers, numerical ranges and conjunction words were also possible and valid. Several postprocessing schemes were devised and tested to standardise correct results (see Section S4 of the SI for details). For minimal processing and optimal outcomes, we omitted from consideration all the answers without any digits and with letters outside the set "e, v, t, o, a, n, d". The rest of the results were evaluated against gold standard answers to compute metrics. After this, any values denoting a range were averaged (the extracted range "1.5–1.6" eV was recorded as 1.55 eV) to ensure that the extracted values would not carry bias toward the ends of the range simply because of rounding practices.

Human opinion sets the gold standard in NLP tasks. From the complete dataset, we manually annotated a subset to establish the correct answers. The 600 text snippets were selected to evenly balance bandgap information on the five materials considered. Six materials science experts performed the annotation (including empty answers) so that each snippet was reviewed by two people. The process is described in Section S5 of the SI. After agreement was achieved, 188 snippets were found to contain 227 numerical bandgap values, and 412 featured none. Out of 188 snippets, the annotators identified 164 snippets with a single bandgap value, and 24 snippets with two or more (up to 6) values.

The annotated dataset was used to compute classification accuracy metrics for the QA models. A true positive was achieved if the QA result was numerically equivalent to the manually annotated value. This way, the QA returned answer "1.5" would be considered a true positive for an annotated value of "1.5 eV", but a returned value of "5" would not. Similarly, a true negative denoted a correctly returned empty text span. These metrics were employed to compute precision (the fraction of correct answers from all retrieved answers) and recall (the fraction of correct returned answers compared to all golden standard answers). We used F1-score, the harmonic mean between precision and recall, to identify the best QA BERT model. Results were compared against the well-established CDE2 code (version 2.1.2) for IE, which was applied to the annotated dataset following previously established procedures[20]. To ensure a fair comparison, all IE tools were utilised on the same hardware (a single NVIDIA Volta V100 GPU), except the GPT models, which can not be executed locally.

Next, we compared the QA results to the results obtained with four different generative models: Mixtral-8[53], Llama3[54], GPT-3.5[25] and GPT-4[26]. Mixtral-8 is based on the language model Mixtral-8 × 7B, which has been demonstrated to match or outperform GPT-3.5 and trained to follow instructions[53,55]. Llama3-ChatQA-1.5–8B is trained from the Llama-3 base model (the second latest Llama model) to excel at conversational question answering[54]. The results were generated using zero-shot prompting, optimised over four different prompts and postprocessed (see SIS7). Results were extracted using temperature 0 to ensure the highest possible determinism of the models. Our prompt engineering tests revealed that different models require different prompts for optimal results. Moreover, we established that even with temperature 0 and a fixed seed (12), the GPT models exhibited different results for repeated calculations (this is an unexpected, nevertheless publicly documented feature of the OpenAI API. *platform.openai.com/docs/advanced-usage/reproducible-outputs* and the seed parameter is still in Beta phase *platform.openai.com/docs/api-reference/chat/create*). Consequently, for GPT computations, we performed 5 repetitions with identical settings and prompts, and we reported the average and the standard deviation of the answers. For the generative models, we also reported the number of hallucinated values. Hallucination refers to the generation of information that is not present in the original context document. Encoder models (e.g., BERT) do not hallucinate, but may sometimes select the wrong value from the context document, characterised as a false positive.

Since the QA model returns the text span with the highest confidence score (the most likely answer), evaluating performance on snippets with a single bandgap value was straightforward. However, it was unclear how to

approach the 24 snippets with up to 6 bandgap values. An easy solution would have been to acknowledge any correct value returned, but this would not have allowed us to extract all possible answers, which is desirable in applications. For a more complete outcome, we shifted from considering just one result with the highest confidence score to considering several. From the many prospective answers returned by the model, we retrieved 6 with the highest confidence scores to ensure that all band gaps could be extracted from each snippet in the annotated dataset.

At this stage, we introduced the QA confidence threshold as a model parameter. A closer examination of the confidence scores associated with the snippet answers revealed that model certainty varies considerably for different snippets, and this should be taken into account (because less confidence points to a likely wrong answer). We examined model performance with different confidence thresholds applied to the top 6 answers: 0.0125, 0.025, 0.05, 0.1, 0.2, 0.4, and 0.8.

Here, all of the answers with the confidence score above the threshold were retained and performance metrics were computed (up to a maximum of 6), while the rest were discarded. We performed a fourfold CV using a manually annotated dataset of 600 snippets to determine the optimal threshold for each QA model. The dataset was split into four folds, with three folds used as the validation set in each iteration, and the remaining fold serving as the test set. The folds were shuffled between each iteration so that each fold served once as a test set. The final results were averaged across all iterations. Low thresholds would permit most answers at the expense of accuracy, which increases recall but lowers precision. The opposite is true for high thresholds, so considering the entire range is a helpful test to establish which confidence threshold best balances precision with recall, optimising F1-scores. For each cross-validated language model, we identified the optimal threshold on the CV validation set, then applied it to the CV test set to compute the final evaluation metrics.

All extracted answers comprised a distribution of bandgap values for a particular material. We performed a statistical analysis for the range, mean and median values. When extracting bandgap values with CDE2, only values corresponding to the set of five perovskites were selected, in order to reliably compare the QA method to CDE2.

## Data availability

The human-annotated dataset of perovskite band gaps used in this study has been made publicly available[56], along with the article identifiers of the 194,322 manuscripts included in the corpus[57]. Annotation workflow, dataset analytics and examples of dataset application can be found in a separate paper[58].

## Code availability

All codes used in this study are publicly available[52,57].

Received: 10 January 2025; Accepted: 29 September 2025;
Published online: 20 November 2025

## References

1. Snaith, H. J. Present status and future prospects of perovskite photovoltaics. *Nat. Mater.* **17**, 372–376 (2018).
2. Correa-Baena, J.-P. et al. Promises and challenges of perovskite solar cells. *Science* **358**, 739–744 (2017).
3. Wu, T. et al. The main progress of perovskite solar cells in 2020-2021. *Nano-Micro Lett.* **13**, 1–18 (2021).
4. Ang, T.-Z. et al. A comprehensive study of renewable energy sources: classifications, challenges and suggestions. *Energy Strategy Rev.* **43**, 100939 (2022).
5. Sathaye, J. et al. *Renewable Energy in the Context of Sustainable Development*, 707–790 (Cambridge University Press, Cambridge, 2011)
6. Bogdanov, D. et al. Radical transformation pathway towards sustainable electricity via evolutionary steps. *Nat. Commun.* **10**, 1–16 (2019).
7. Jeon, N. J. et al. Compositional engineering of perovskite materials for high-performance solar cells. *Nature* **517**, 476–480 (2015).
8. Mbumba, M. T. et al. Compositional engineering solutions for decreasing trap state density and improving thermal stability in perovskite solar cells. *J. Mater. Chem. C* **9**, 14047–14064 (2021).
9. Laakso, J., Todorović, M., Li, J., Zhang, G.-X. & Rinke, P. Compositional engineering of perovskites with machine learning. *Phys. Rev. Mater.* **6**, 113801 (2022).
10. Bush, K. A. et al. Compositional engineering for efficient wide band gap perovskites with improved stability to photoinduced phase segregation. *ACS Energy Lett.* **3**, 428–435 (2018).
11. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. Bert: pre-training of deep bidirectional transformers for language understanding. *NAACL*. 4171–4186 (2019).
12. Hiszpanski, A. M. et al. Nanomaterial synthesis insights from machine learning of scientific articles by extracting, structuring, and visualizing knowledge. *J. Chem. Inf. Model.* **60**, 2876–2887 (2020).
13. Tchoua, R. B. et al. Creating training data for scientific named entity recognition with minimal human effort, In *Computational Science-ICCS 2019: 19th International Conference, Faro, Portugal, June 12-14, 2019, Proceedings, Part I 19*, 398-411. (Springer, 2019).
14. Weston, L. et al. Named entity recognition and normalization applied to large-scale information extraction from the materials science literature. *J. Chem. Inf. Model.* **59**, 3692–3702 (2019).
15. Trewartha, A. et al. Quantifying the advantage of domain-specific pre-training on named entity recognition tasks in materials science. *Patterns* **3**, 100488 (2022).
16. Kim, E. et al. Machine-learned and codified synthesis parameters of oxide materials. *Sci. Data* **4**, 1–9 (2017).
17. Mavracic, J., Court, C. J., Isazawa, T., Elliott, S. R. & Cole, J. M. ChemDataExtractor 2.0: autopopulated ontologies for materials science. *J. Chem. Inf. Model.* **61**, 4280–4289 (2021).
18. Huang, S. & Cole, J. M. A database of battery materials auto-generated using ChemDataExtractor. *Sci. Data* **7**, 260 (2020).
19. Sierepeklis, O. & Cole, J. M. A thermoelectric materials database auto-generated from the scientific literature using ChemDataExtractor. *Sci. Data* **9**, 648 (2022).
20. Dong, Q. & Cole, J. M. Auto-generated database of semiconductor band gaps using ChemDataExtractor. *Sci. Data* **9**, 193 (2022).
21. Gilligan, L., Cobelli, M., Taufour, V. & Sanvito, S. A rule-free workflow for the automated generation of databases from scientific literature. *Npj Comput. Mater.* **9**, 222 (2023).
22. Zheng, Z., Zhang, O., Borgs, C., Chayes, J. & Yaghi, O. ChatGPT chemistry assistant for text mining and the prediction of MOF synthesis. *J. Am. Chem. Soc.* **145**, 18048–18062 (2023).
23. Dagdelen, J. et al. Structured information extraction from scientific text with large language models. *Nat. Commun.* **15**, 1418 (2024).
24. Foppiano, L., Lambard, G., Amagasa, T. & Ishii, M. Mining experimental data from materials science literature with large language models: an evaluation study. *Sci. Technol. Adv. Materials: Methods.* **4**, 2356506 (2024).
25. OpenAI, GPT-3.5 Turbo, platform.openai.com/docs/models/gpt-3-5-turbo (2022).
26. OpenAI, GPT-4 openai.com/gpt-4 (2024).
27. Beltagy, I., Lo, K. & Cohan, A. SciBERT: a pretrained language model for scientific text. Preprint at *arXiv* https://doi.org/10.48550/arXiv.1903.10676 (2019).
28. Shetty, P. et al. A general-purpose material property data extraction pipeline from large polymer corpora using natural language processing. *npj Comput. Mater.* **9**, 52 (2023).
29. Gupta, T., Zaki, M. & Krishnan, N. M. A. Mausam MatSciBERT: A materials domain language model for text mining and information extraction. *npj Comput. Mater.* **8**, 102 (2022).

30. Rajpurkar, P., Jia, R. & Liang, P. Know what you don't know: unanswerable questions for SQuAD. Preprint at *arXiv* https://doi.org/10.48550/arXiv.1806.03822 (2018).
31. Cannelli, O. et al. Quantifying photoinduced polaronic distortions in inorganic lead halide perovskite nanocrystals. *J. Am. Chem. Soc.* **143**, 9048–9059 (2021).
32. Yang, Z. et al. Impact of the halide cage on the electronic properties of fully inorganic cesium lead halide perovskites. *ACS Energy Lett.* **2**, 1621–1627 (2017).
33. Roh, J. et al. Yb3+ speciation and energy-transfer dynamics in quantum-cutting Yb3+-doped CsPbCl3 perovskite nanocrystals and single crystals. *Phys. Rev. Mater.* **4**, 105405 (2020).
34. Chen, Y. & Zulkernine, F. BIRD-QA: a BERT-based information retrieval approach to domain specific question answering. In *2021 IEEE International Conference On Big Data (Big Data)*. 3503–3510 (IEEE 2021).
35. Rawat, A. & Samant, S. Comparative analysis of transformer based models for question answering. In *2nd International Conference On Innovative Sustainable Computational Technologies (CISCT)* 1–6 (IEEE 2022).
36. Akhila, N. et al. Comparative study of BERT models and RoBERTa in transformer based question answering. In *2023 3rd International Conference On Intelligent Technologies (CONIT)* 1–5 (IEEE 2023).
37. Mishra, V. et al. Foundational large language models for materials research. Preprint *arXiv* https://doi.org/10.48550/arXiv.2412.09560 (2024).
38. Vona, C., Nabok, D. & Draxl, C. Electronic structure of (organic-) inorganic metal halide perovskites: the dilemma of choosing the right functional. *Adv. Theory Simul.* **5**, 2100496 (2022).
39. Hira, K. et al. Reconstructing the materials tetrahedron: challenges in materials information extraction. *Digital Discov.* **3**, 1021–1037 (2024).
40. arXiV API documentation. info.arxiv.org. Accessed 2023-10-24.
41. Elsevier Developer Portal. dev.elsevier.com. Accessed 2023-10-24.
42. Core API. core.ac.uk/services/api. Accessed 2023-10-24.
43. Knoth, P. et al. CORE: a Global aggregation Service for Open access Papers. *Sci. Data* **10**, 366 (2023).
44. Springer Nature API portal. dev.springernature.com/. Accessed 2023-10-24.
45. Royal Society of Chemistry publications. pubs.rsc.org. Accessed 2023-10-24.
46. Min, D. et al. Exploring the impact of table-to-text methods on augmenting llm-based question answering with domain hybrid data. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2402.12869 (2024).
47. Zhao, Y. et al. Investigating table-to-text generation capabilities of LLMs in real-world information seeking scenarios. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2305.14987 (2023).
48. Elasticsearch. www.elastic.co. Accessed 2023-10-30.
49. Vaswani, A. et al. Attention is all you need. *Proc. Adv. Neural Inf. Process. Syst.* **30**, 5998–6008 (2017).
50. Gu, Y. et al. Domain-specific language model pretraining for biomedical natural language processing. *ACM Trans. Comput. Healthc.* **3**, 1–23 (2021).
51. Huggingface Transformers-library. *huggingface.co/docs/transformers/index* (2024).
52. *QA code repository*. gitlab.com/mil-utu/QA_for_MS (2024).
53. Jiang, A. et al. Mixtral of experts. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2401.04088 (2024).
54. Liu, Z. et al. ChatQA: surpassing GPT-4 on conversational QA and RAG. *Adv. Neural Inf. Process. Syst*. **37**, 15416–15459 (2024).
55. Furman, Daniel. dfurman/Mixtral-8 × 7B-Instruct-v0.1. huggingface.co/dfurman/Mixtral-8x7B-Instruct-v0.1 (2023).
56. PV600 dataset. https://doi.org/10.5281/zenodo.15124019 (2025).
57. Publication identifiers and QA code. https://doi.org/10.5281/zenodo.17177042 (2025).
58. Sipilä, M., Mehryary, F., Pyysalo, S., Ginter, F. & Todorović, M. Annotated textual dataset PV600 of perovskite bandgaps for information extraction from literature. *Sci. Data* **12**, 1401 (2025).
59. Belayachi, W. et al. Study of hybrid organic–inorganic halide perovskite solar cells based on MAI[(PbI2)1-x(CuI)x] absorber layers and their long-term stability. *J. Mater. Sci.* **32**, 20684–20697 (2021).
60. Weber, O. J., Charles, B. & Weller, M. T. Phase behaviour and composition in the formamidinium–methylammonium hybrid lead iodide perovskite solid solution. *J. Mater. Chem. A* **4**, 15375–15382 (2016).
61. Mohan, S. et al. Static anti-solvent treatment on lead iodide in two-step deposition of methylammonium lead iodide perovskite thin films. *Optical Mater.* **147**, 114720 (2024).
62. Rehman, W. et al. Charge-carrier dynamics and mobilities in formamidinium lead mixed-halide perovskites. *Adv. Mater.* **27**, 7938–7944 (2015).
63. Qiu, L. et al. Perovskite MAPb(Br1-xClx)3 single crystals: solution growth and electrical properties. *J. Cryst. Growth* **549**, 125869 (2020).
64. Adonin, S. et al. Antimony (V) complex halides: lead-free perovskite-like materials for hybrid solar cells. *Adv. Energy Mater.* **8**, 1701140 (2018).
65. Masi, S. et al. Chemi-structural stabilization of formamidinium lead iodide perovskite by using embedded quantum dots. *ACS Energy Lett.* **5**, 418–427 (2020).
66. Eperon, G. E. et al. Formamidinium lead trihalide: a broadly tunable perovskite for efficient planar heterojunction solar cells. *Energy Environ. Sci.* **7**, 982–988 (2014).
67. Yang, Z. et al. Effects of formamidinium and bromide ion substitution in methylammonium lead triiodide toward high-performance perovskite solar cells. *Nano Energy* **22**, 328–337 (2016).
68. Ahmad, W., Khan, J., Niu, G. & Tang, J. Inorganic CsPbI3 perovskite-based solar cells: a choice for a tandem device. *Sol. RRL* **1**, 1700048 (2017).
69. Ye, Q. et al. Stabilizing γ-CsPbI3 perovskite via phenylethylammonium for efficient solar cells with open-circuit voltage over 1.3 V. *Small* **16**, 2005246 (2020).
70. Parida, B. et al. Two-step growth of CsPbI3-xBrx films employing dynamic CsBr treatment: toward all-inorganic perovskite photovoltaics with enhanced stability. *J. Mater. Chem. A* **7**, 18488–18498 (2019).
71. Yuan, H. et al. Enhanced charge extraction by setting intermediate energy levels in all-inorganic CsPbBr3 perovskite solar cells. *Electrochim. Acta* **279**, 84–90 (2018).
72. Liu, J. et al. Growing high-quality CsPbBr3 by using porous CsPb2Br5 as an intermediate: a promising light absorber in carbon-based perovskite solar cells. *Sustain. Energy Fuels* **3**, 184–194 (2019).
73. Liang, J. et al. All-inorganic perovskite solar cells. *J. Am. Chem. Soc.* **138**, 15829–15832 (2016).
74. Solhtalab, N., Mohammadi, M., Eskandari, M. & Fathi, D. Efficiency improvement of half-tandem CIGS/perovskite solar cell by designing nano-prism nanostructure as the controllable light trapping. *Energy Rep.* **8**, 1298–1308 (2022).


## Acknowledgements

The authors thank the CSC-IT Centre for Science in Finland for high-performance computing resources. We acknowledge Mahboubeh Hadadian, Aleksi Kamppinen, Christer Söderholm and Ransell D'Souza for participating in the evaluation dataset annotation and Emil Nuutinen for the QA training scripts. The research was funded by the Research Council of Finland through grant number 345698. M.S. thanks the University of Turku Graduate School (UTUGS) and Finnish Cultural Foundation (grant number 241085) grants for doctoral research.

## Author contributions

M.T., F.G. and S.P. conceived the original plan for the research and supervised the work. M.S. performed all computational and analysis work and drafted the manuscript. F.M. advised on the NLP training and application. Everyone contributed to the data analysis and writing the manuscript.

## Competing interests

Milica Todorović is an Editorial Board Member for Communications Materials and was not involved in the editorial review or the decision to publish this Article.

## Additional information

**Supplementary information** The online version contains supplementary material available at https://doi.org/10.1038/s43246-025-00979-w.

**Correspondence** and requests for materials should be addressed to Milica Todorović.

**Peer review information** *Communications Materials* thanks the anonymous reviewers for their contribution to the peer review of this work. A peer review file is available.