Digital Discovery

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal | View Issue

Cite this: *Digital Discovery*, 2025, **4**, 998

Auto-generating question-answering datasets with domain-specific knowledge for language models in scientific tasks†

Zongqian Li [a] and Jacqueline M. Cole *[ab]

Large language models (LLMs) have emerged as a useful tool for the public to process and respond to a vast range of interactive text-based queries. While foundational LLMs are well suited to making general user queries, smaller language models that have been trained on custom text from a specific domain of interest tend to display superior performance on queries about that domain, can operate faster and improve efficiency. Nonetheless, considerable resources are still needed to pre-train a language model with custom data. We present a pipeline that shows a way to overcome this need for pre-training. The pipeline first uses new algorithms that we have designed to produce a large, high-quality question-answering dataset (SCQA) for a particular domain of interest, solar cells. These algorithms employed a solar-cell database that had been auto-generated using the 'chemistry-aware' natural language processing tool, ChemDataExtractor. In turn, this SCQA dataset is used to fine-tune language models, whose resulting F_1-scores of performance far exceed (by 10–20%) those of analogous language models that have been fine-tuned against a general-English language QA dataset, SQuAD. Importantly, the performance of the language models fine-tuned against the SCQA dataset does not depend on the size of their architecture, whether or not the tokens were cased or uncased or whether or not the foundational language models were further pre-trained with domain-specific data or fine-tuned directly from their vanilla state. This shows that this domain-specific SCQA dataset produced by our algorithms has sufficient intrinsic domain knowledge to be directly fine-tuned against a foundational language model for immediate use with improved performance.

Received 25th September 2024
Accepted 19th February 2025

DOI: 10.1039/d4dd00307a

rsc.li/digitaldiscovery

1 Introduction

Foundational large language models (LLMs) have created a paradigm shift in text processing owing to their wide-ranging applications. In addition to the well-known black-box LLMs such as ChatGPT[1] and Gemini,[2] open-source LLMs such as LLaMA[3] and Mistral[4] are also accessible to the public and contain parameter counts ranging from billions to hundreds of billions.

However, their generalisability is predicated on their need to be pre-trained on a massive corpus whose knowledge base is diverse enough to contextualise information across all possible domains of user interest.[5] In practice, this generalisability in an LLM is difficult to achieve by all except those who have access to enough computing resources and widespread information to pre-train such models. Such LLMs also require considerable resources to run which restricts their application beyond local deployment, imposing a considerable and non-negligible financial and resources outlay.

Small language models (SLMs) have emerged as an increasingly popular alternative to LLMs for domain-specific applications where a user seeks information that focuses on a particular domain of interest. An SLM is pre-trained on a much smaller corpus than an LLM, whose texts focus on a target application. The SLM will tend to perform better than an LLM when a user queries it on a topic within the domain area for which it has been developed. Such an SLM can operate in a standalone fashion as it does not depend upon external information. It also responds faster to queries and consumes less resource than an analogous LLM.

Foundational SLMs whose architectures are based on bidirectional encoder representations from transformers (BERT) are a popular option (within the current state-of-the-art frame where BERT-base language models are now considered to be 'small'). Their popularity arises partly because they are efficient and a large open-source community has come together to aid their development. Furthermore, BERT models are less exposed to the issue of hallucinations that have plagued many LLMs;[6,7] the bidirectional nature of BERT models mitigates these issues.[8,9]

SLMs typified by BERT architectures are also far more environmentally friendly than LLMs, given that pre-training

[a]*Cavendish Laboratory, Department of Physics, University of Cambridge, J. J. Thomson Avenue, Cambridge, CB3 0HE, UK. E-mail: jmc61@cam.ac.uk*

[b]*ISIS Neutron and Muon Source, Rutherford Appleton Laboratory, Harwell Science and Innovation Campus, Chilton, Didcot, Oxfordshire, OX11 0QX, UK*

† Electronic supplementary information (ESI) available. See DOI: https://doi.org/10.1039/d4dd00307a

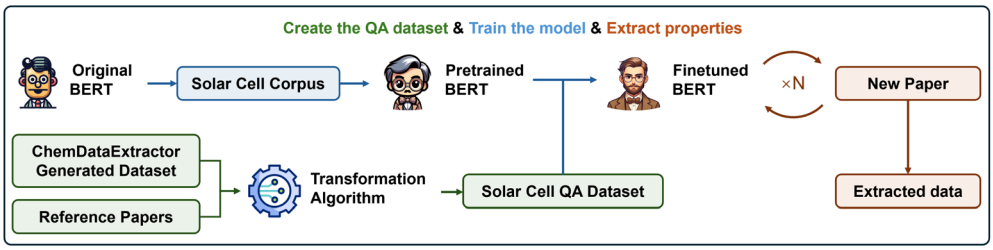

Fig. 1 Overall pipeline for creating the domain-specific QA dataset, training or fine-tuning the BERT models, and extracting properties from papers.

language models consumes a large amount of resources. Even the cost of pre-training a baseline BERT-base model is considerable, resulting in an estimated 1438 lbs of CO_2 emissions; to put this into perspective, this amount is on par with the CO_2 emission levels of a flight from New York to San Francisco.[10] So, one can only imagine the environmental cost of pre-training one of the latest GPT models. Even so, pre-training an SLM consumes considerable resources.

One way to circumvent the need to pre-train an SLM is to employ knowledge distillation. This is a method that effectively moves the knowledge from foundational language models by transferring all the domain-specific knowledge from rich sources into vanilla SLMs during their fine-tuning stages of development.[11] By structuring this domain-specific information into labelled data, the knowledge can be learned quickly by SLMs in a prompt-based way. The effectiveness of this process is nonetheless influenced by data quality, making the automatic generation of high-quality input data crucial.

In the field of materials science, high-quality question-answering (QA) datasets about materials and their properties have either been made manually or generatively *via* LLMs. For example, the manually-curated MaScQA dataset has been designed for materials science and includes four types of questions: multiple choice, numerical with multiple choices, matching type, and numerical questions.[12] Meanwhile, the DARWIN project has generatively-produced QA datasets for materials science using an LLM. Thereby, DARWIN fine-tunes a model based on Vicuna to generate QA instructions from text, which is then used to create large-scale datasets for fine-tuning LLaMA on tasks such as classification and regression.[13] Work by others employs an LLM-based instruction process to generatively create a dataset that fine-tunes an LLM.[14]

In this paper, we show how an SLM can be designed for materials-science applications using knowledge distillation with a large and high-quality question-answering (QA) dataset about materials and their properties. We show how this QA dataset first needs to be pre-processed from an existing materials database that was generated *via* the 'chemistry-aware' natural-language processing tool, ChemDataExtractor.[15–18] A ChemDataExtractor-generated materials database about solar cells[19] was selected as the case study for this work, given that it derives from a large corpus of papers and it contributes to environmentally friendly solutions.

The study begins by exploring how the performance of BERT-based language models is affected by further pre-training them with different-sized corpora, tokenization criteria and BERT architectures. With that demonstration in hand, we showcase the algorithms that are used to create large QA datasets for a domain-specific need from the selected ChemDataExtractor-generated database. We then employ these QA datasets to fine-tune BERT models, the performance of which is assessed against various metrics. Ultimately, we demonstrate that the performance of these BERT models is determined far more by their domain-specific aspect of fine-tuning rather than on their domain-specific further pre-training. Moreover, their performance is not dependent on the size of a foundational language model, at least down to the baseline size of a BERT-base model. This means that our methods could help to open up a new way to employ SLMs for domain-specific materials-science applications. The overarching project is illustrated in Fig. 1.

2 Methodology

2.1 Solar-cell corpora and further pretrained BERT models

To improve the pretraining of BERT models specifically for solar cells, we constructed three distinct corpora from the solar-cell papers. These corpora were created by extracting papers through a search on the keyword "solar cell" from three leading publishers: Elsevier, the Royal Society of Chemistry (RSC), and Springer. This comprehensive collection was then divided into three subsets based on size: Solar Cell Corpus Small (scsmall), Solar Cell Corpus Medium (scmedium), and Solar Cell Corpus Large (sclarge), as detailed in Table 1. A corpus of text containing a given number of papers with different content will

Table 1 Summary of the maximum (max.), average (ave.) and median (med.) number of tokens (token length) in the small, medium and large solar-cell corpora of papers from different publishers. E, R, and S represent Elsevier, RSC, and Springer, respectively. A token is the smallest unit processed by a language model and can represent either a word or a sub-word. Before being input into the language model, natural language text is segmented into tokens

Parameters	scsmall	scmedium	sclarge
Number of papers	8875	35 385	161 183
Publisher	E	E, R	E, R, S
Max. token length	74 749	74 749	818 446
Ave. token length	6915	5218	5589
Med. token length	6266	4673	5185
Total token count	61 372 439	184 646 322	900 806 049

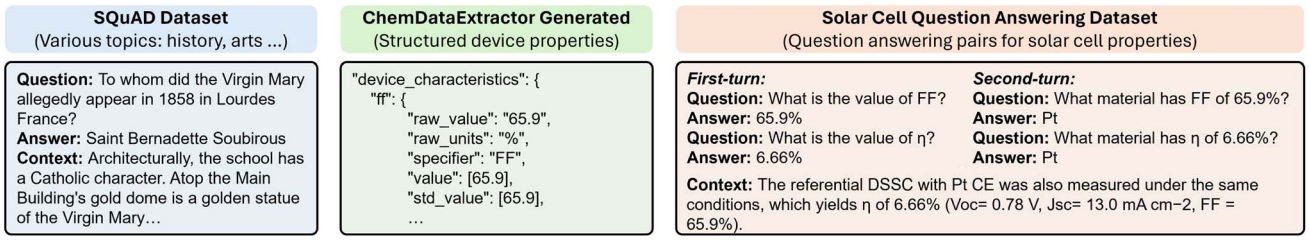

Fig. 2 (left) A QA pair in the SQuAD dataset. (middle) A database record for the fill factor, FF, a performance characteristic of solar cells, taken from our ChemDataExtractor-generated database. (right) The result of converting this database record into a QA pair *via* an in-house developed algorithm that first retrieves the sentence in the original paper that contains this FF information and then automatically reframes this into a question and answer, respectively.

contain a different total number of tokens to another corpus with different papers; additionally, each contains varying amounts of domain-specific knowledge, which can affect the performance of language models in downstream tasks.

We further pretrained existing BERT models on each of these tailored corpora, starting from the four foundational BERT-base-(un)cased and BERT-large-(un)cased models.[8] This process resulted in the generation of 12 novel BERT model variants. The terms "base" and "large" refer to the number of model parameters used in the BERT model, either 110 million or 340 million, respectively. Meanwhile, "cased" and "uncased" denote whether the model distinguishes upper and lowercased letters or not. For instance, the BERT-base-cased-scsmall model refers to the BERT-base-cased model that was further pretrained on the scsmall corpus.

2.2 The need for QA datasets with domain-specific knowledge: beyond the general English-language Stanford Question Answering Dataset

To enhance the capabilities of language models in extractive QA tasks, several datasets have been developed by others for fine-tuning purposes. Among these, the Stanford Question Answering Dataset version 1.1 (SQuAD) stands out as a particularly significant and widely utilized resource.[20] It comprises 107 785 QA pairs designed to test reading comprehension of general English language. Each pair includes a question, one or more corresponding answers, and a context passage, with the answer being a text span that has been taken directly from the context. The structure of QA pairs within the SQuAD dataset is illustrated in Fig. 2 (left).

Despite its extensive size and diverse range of topics, derived exclusively from Wikipedia articles, the SQuAD dataset lacks a focus on domain-specific language. This limitation highlights the need for large QA datasets that are enriched with domain-specific knowledge. The creation of such datasets is anticipated to improve the performance of language models on tasks within specific domains, addressing the gap in the current dataset offerings.

2.3 Algorithms for auto-generating domain-specific QA datasets

ChemDataExtractor has generated databases with data records that detail materials and their properties, by extracting information from papers.[17,19,21–29] These structured data records contain rich and high-quality domain-specific knowledge. While these records contain the Document Object Identifiers (DOIs) of the papers from which they were sourced, they tend to lack the originating sentences that contain these material names and their properties. Moreover, the format of these data records diverges from the format used in a QA dataset, illustrated in Fig. 2 (middle). This discrepancy poses challenges for directly integrating this valuable data into QA systems, underscoring the need for format harmonization to enhance usability and accessibility in applied contexts and scenarios.

We therefore developed a new way to fine-tune language models using ChemDataExtractor-generated materials databases as the source information. Thereby, a set of in-house algorithms (Fig. 3) was created that converts these materials databases into large domain-specific QA datasets that are then employed to fine-tune language models. For each record of a given ChemDataExtractor-generated database, an algorithm (Algorithm 1) first retrieves the text from the paper which ChemDataExtractor used to extract each property characteristic of a given material; a second algorithm (Algorithm 2) uses this text together with its associated database record, *i.e.*, material, property, value, unit (and error if present), to automatically reframe this information into a pair of questions and answers.

The process by which Algorithm 2 converts the original text and extracted data into pairs of questions and answers is worthy of further explanation. This algorithm classifies the property from each data record into a quantitative or non-quantitative property; *e.g.*, open-circuit voltage is a quantitative property; a material component is a non-quantitative property. This defines the type of question that the specified property will adopt. Its answer will be one of the extracted data value(s) with units, and perhaps error(s) if that information has been captured; pending that the answer is also shown in the retrieved text. This caveat ensures that the QA database afforded is highly accurate; in fact, this part of the algorithm can also validate the ChemDataExtractor-generated datasets, since it naturally filters out any inconsistency between the original text and the extracted data. The criteria listed in Table 2 demonstrate the types of questions and answers that are generated and the conditions upon which they are formed.

Having essentially cast a "what is the property?" type of QA database, the 'double-turn QA' workflow previously employed

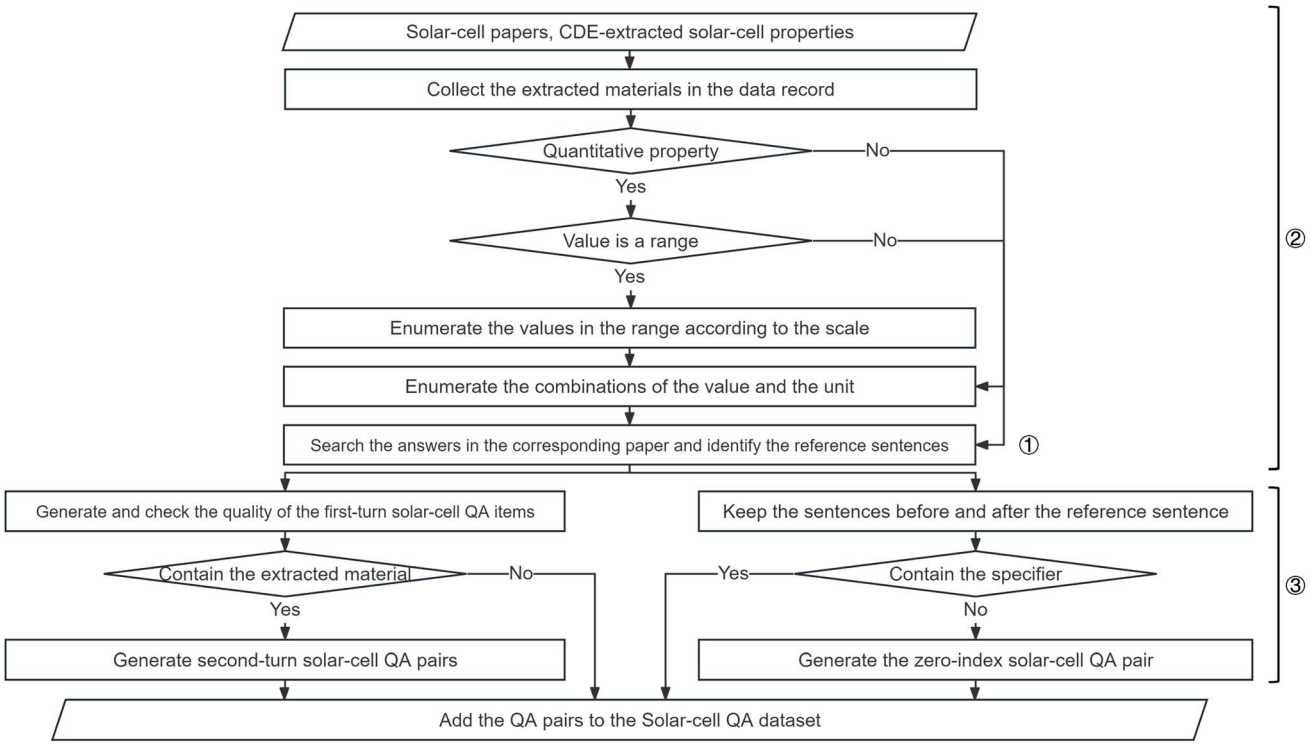

Fig. 3 Algorithms for transforming a ChemDataExtractor-generated structured database into a QA dataset that can be used for fine-tuning language models. The inputs to the algorithmic process are a database created by ChemDataExtractor and the papers used for extracting the material names and associated properties in this database. Algorithm 1 retrieves the originating text in the papers from which the material names and their properties were sourced. According to the kind of property and its type of value, different forms of answers are searched in this retrieved text. If one sentence contains both the answer and the specifier (a keyword in the retrieved text indicating the property), a first-turn QA pair is created, asking for the value of the property (Algorithm 2). If the sentence contains the material as well, the second-turn QA pair is generated from the first-turn QA pair, asking for the type of material associated with this property value. If the sentences near the reference sentence do not contain any information that is relevant to the answer, they are used to create zero-index QA pairs (unanswerable pairs, where the answer is not present in the text) (Algorithm 3). Finally, all the generated first-turn, second-turn, and zero-index QA pairs are collected to constitute the domain-specific QA dataset. The properties and values extracted in the first-turn process, along with the materials extracted in the second-turn process, form material–property–value pairs that can be used to build material databases.

by Huang and Cole[30] was applied in a subsequent step, to generate a "given a property with value, what is the material?" type of database. This is because one key goal of language models lies in their application in data-driven materials discovery, the success of which is governed by finding semantic structures in sentences that link structure–property relationships that exist about a given material application for different areas.[31–33]

Another algorithm, Algorithm 3, was developed that could be nested into the algorithm described above to realize this

Table 2 A list of criteria that were used to generate first and second-turn pairs of questions and answers

No.	Criteria
1	The first-turn question is "What is the value of 'property'?" and "What is 'property'?" for quantitative and non-quantitative properties, respectively. The second-turn question is "What material has a 'property' of 'value'?". Only first-turn QA items with quantitative properties have second-turn QA items
2	The answer is the combinations of "raw_value" and "raw_units" in different ways. If "raw_value" is a range, all values in the range will be searched in the paper
3	The context is one sentence that contains both the specifier and the answer in the paper. All the sentences that contain the specifier and the answer are considered
4	"device_characteristics", "device_metrology", "psc_material_metrology", and "dsc_material_metrology" are groups of quantitative properties
5	"psc_material_components" and "dsc_material_components" are groups of non-quantitative properties
6	The material should be from "psc_material_components" or "dsc_material_components" in each data record extracted by ChemDataExtractor. There should be only one kind of material in the context of second-turn QA item

'double-turn question-answering' capability.[30] Thereby, if the text of the original paper contains a material name, and the value of the target property is quantitative, then the question becomes "What material has the 'property' of 'value'?" The nesting of this algorithm makes the enquiry much more restrictive in terms of the number of questions and answers that it can generate because it is dependent on both material and property fields.

2.4 Solar-cell question-answering datasets

Our transformation algorithms were then applied to the case study on solar cell databases. Fig. 3 illustrates how the algorithms take a database record as input (middle) and produce a QA pair as output (right). Once applied to all records of the ChemDataExtractor-generated solar cell database, this algorithmic process automatically afforded a large domain-specific QA dataset about solar cell properties.

The resulting QA database, Solar Cell Question Answering (SCQA) Dataset, contains a total of 42 882 first-turn QA pairs that have 16 properties about solar cells and their associated materials; most answers are values, as can be judged by the average character-length of the answer, *cf.* Table 3. There are 4386 second-turn QA pairs that have 10 properties; this lower number arises because some of the extra properties, in the "what is the property?" type of QA pairs, are non-quantitative. There are also 1212 zero-index QA pairs where the answers are not present in the contexts as well.

3 Technical evaluation

3.1 Efficacy of the SCQA dataset

To assess the property values within the SCQA dataset, four metrics were employed: F_1 score, precision, recall, and exact match (EM). Precision measures the fraction of correctly predicted characters among all predicted characters. Recall quantifies the fraction of actual positive characters correctly identified in the predictions. The F_1 score, an amalgamation of precision and recall, provides a balanced measure of accuracy. These metrics are expressed through the equations:

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

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2)$$

$$F_1\ \text{score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (3)$$

where TP represents true positives, FP false positives, and FN false negatives. EM, in contrast to the F_1 score, evaluates if the predicted answer exactly matches the reference answer, calculating the ratio of completely correct predictions.

To verify the quality of the SCQA dataset, an evaluation set comprising 1000 first-turn QA pairs, 100 second-turn QA pairs, and 100 zero-index QA pairs was assembled. These were randomly selected from the SCQA dataset, reflecting the distribution of properties. While Table 4 details the outcomes for first-turn QA pairs, the EM for second-turn QA pairs was 72%, and the accuracy for zero-index QA pairs reached 100%. These findings attest to the quality of the SCQA dataset. Any discrepancies stem predominantly from inherent issues of the original data-extraction capabilities of ChemDataExtractor, illustrating that the QA dataset's generation algorithm operates with good efficacy, sidestepping almost all potential problems.

Table 3 A summary of the distribution of property-based questions and answers in the auto-generated QA dataset about solar cell properties

Parameter	First-turn	Second-turn
The number of:		
Properties	16	10
Total QA pairs	42 882	4386
QA pairs in the train set	34 305	3508
QA pairs in the test set	8577	878
The number of QApairsfor:		
Power-conversion efficiency	16 081	1856
Open-circuit voltage	8619	1207
Short-circuit current density	3405	460
The average length of:		
Context/characters	240	245
Answer/characters	6	4

3.2 Performance of the 64 BERT language models

Having developed large bespoke "what is the property?" and "what material has the property?" types of QA datasets about solar cell properties, we assessed the relative performance of 64 BERT language models, which differed by the QA dataset that was used to fine-tune them, the size of the foundational model and the corpus used for further pretraining.

Fig. 4 shows the performance (F_1 score) for 32 of the BERT models, 16 of which had been fine-tuned on Wikipedia-related QA pairs (SQuAD),[20] 16 of which had been fine-tuned on the "what is the property?" domain-specific QA dataset but not SQuAD; all these models had been tested on the first-turn QA pairs in the test set of the SCQA dataset.

The stark boost in performance by using the domain-specific QA dataset shows the importance of expertise knowledge in QA tasks. The size of the SCQA dataset is 39.16% of the SQuAD; while the F_1 scores for their language models are better by a maximum of 18.08% and improve by an average of 13.46% when the SCQA dataset is employed.

This SCQA-related performance contrasts starkly with the situation where there is not enough domain knowledge through the exclusive use of the SQuAD: cased models generally perform better than uncased models; the larger corpus contributes more to the model performance; and BERT-large models outperform BERT-base models. These influences of language model size, corpus size, or cased distinction of their tokens are all eliminated by fine-tuning the BERT models on a domain-specific SCQA dataset.

Table 4 Evaluation results for the first-turn QA pairs in the SCQA dataset

Property	Weight	F_1	Precision	Recall	EM
Power-conversion efficiency	37.50	92.12	92.63	92.01	91.84
Open-circuit voltage	20.10	97.34	97.75	97.17	95.50
Short-circuit current density	7.94	96.33	97.50	95.94	95.00
Fill factor	5.01	94.00	94.00	94.00	94.00
Active area	4.78	96.67	98.00	96.00	94.00
Solar simulator and irradiance	4.05	97.50	97.50	97.50	97.50
Counter electrode	3.34	64.33	66.67	63.33	60.00
Substrate	2.79	85.00	90.00	83.33	80.00
Other	14.49	49.18	50.71	48.39	45.71
All		86.68	87.55	86.35	85.19

The remaining 32 BERT models included 16 BERT models that were fine-tuned on a QA dataset that combined the SQuAD[20] and the "what is the property?" QA pairs of the SCQA dataset. Negligible differences in performance were observed among them or when compared with the aforementioned 16 BERT models that were fine-tuned exclusively on the "what is the property?" domain-specific QA pairs of the SCQA dataset. The other 16 BERT models were fine-tuned on the entire SCQA dataset including both "what is the property?" and "what material has the property?" types of QA pairs and their performance was similar to each other.

Overall, the results of this study indicate that BERT-based language models that have been fine-tuned on large domain-specific QA datasets offer far superior performance when used in that domain. This appears to be irrespective of the size of the language model, corpus size, or cased or uncased distinction of their tokens, within the range of sizes and distinctions studied. These results contrast markedly with those of the 16 BERT language models that were fine-tuned on a QA dataset whose QA pairs are only from general English language (SQuAD); in those results, the performance of the language models tracked approximately in proportion to: the number of parameters in

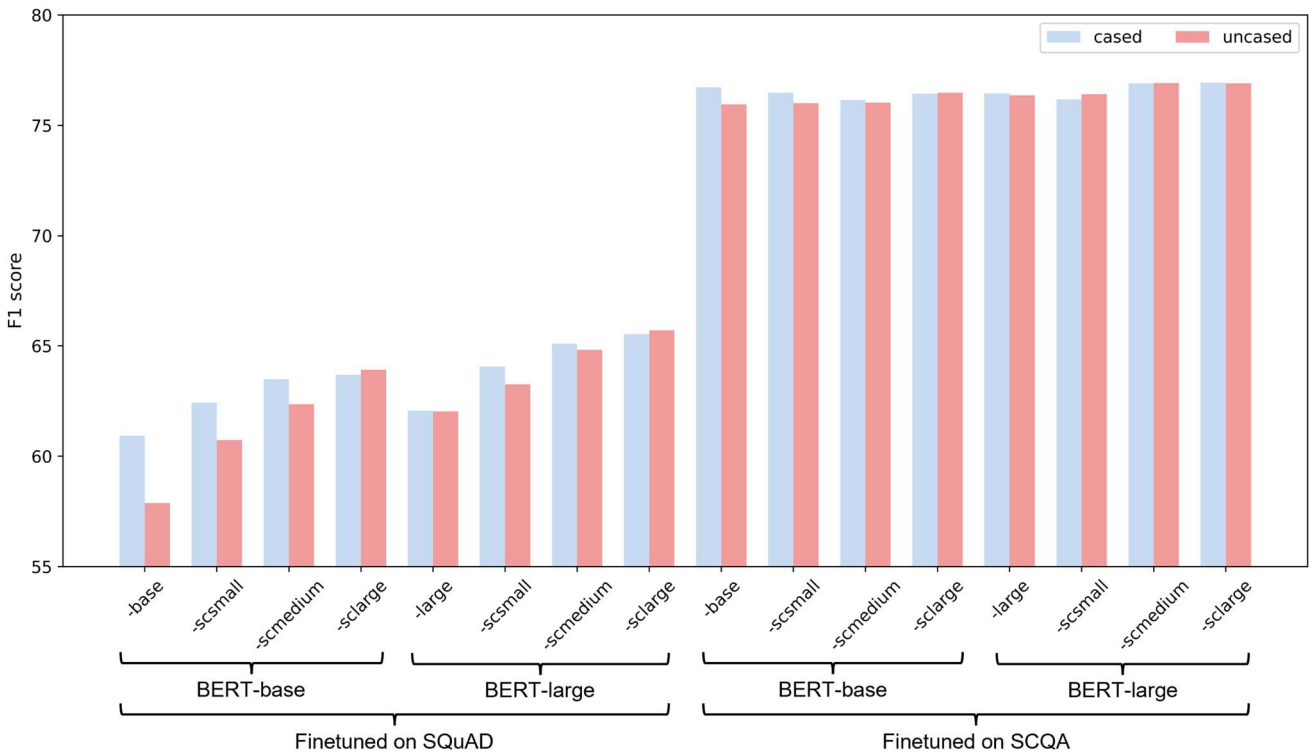

Fig. 4 F_1 scores for the BERT models fine-tuned on the SQuAD (left) and on the first-turn QA pairs in the SCQA dataset (right). BERT-base and BERT-large models were either used in their vanilla state where a column is marked as '-base' or '-large', or further pre-trained on one of three corpora of different sizes as judged by their number of tokens: scsmall (61.4 M tokens), scmedium (184.6 M tokens), sclarge (900.8 M tokens); their pre-training details are provided in the ESI,† section C. The performance of BERT models fine-tuned against a mixture of the SQuAD and SCQA dataset is also given in the ESI,† section D; unsurprisingly, their F_1 scores are similar to those from BERT models that were fine-tuned against SCQA datasets.

the language model, the corpus size and whether or not cased and uncased tokens were distinguishable from each other.

4 Conclusions

This study introduces a collection of innovative algorithms that are capable of automatically generating large-scale question and answer (QA) datasets that contain domain-expertise from materials databases that have been produced by ChemDataExtractor. These datasets are particularly effective for fine-tuning small language models (SLMs), where the enhancement in model performance is primarily attributed to the domain-specific nature of the fine-tuning process, rather than further pre-training in specific domains. The findings indicate a potential shift towards reducing the reliance on large foundational language models by focusing on domain-specific fine-tuning, which requires significantly fewer computational resources. Implementing this approach on a broader scale could play a significant role in making language models more accessible worldwide. This is especially pertinent for environmentally friendly development initiatives, where the ability to customize language model applications to meet unique national environmental requirements is essential.

Although QA datasets for the solar cell domain are showcased in this paper, the presented algorithms can be used to transform any database created by ChemDataExtractor into an extractive QA dataset containing domain-specific knowledge. Such a QA dataset can then be used to fine-tune SLMs for information extraction. In the future, additional types of questions, such as multiple choice and numerical questions, could be designed based on the extracted information to create more diverse tasks.[34] This would enhance the generalization capabilities of SLMs across various domains and tasks. Beyond the focus of current work on data and model size efficiency, efficient training methods, such as CRAMMING,[35] could decrease computational cost as well.

Data availability

The codebase for this work has been uploaded to Zenodo with **https://doi.org/10.5281/zenodo.14884904**. Datasets and models have been uploaded to Huggingface **https://huggingface.co/CambridgeMolecularEngineering**.

Author contributions

J. M. C. conceived the overarching project. The study was designed by Z. L. and J. M. C. Z. L. created the workflow, designed and deployed the algorithms and analysed the performance of the resulting language models under the supervision of J. M. C. J. M. C. drafted the manuscript with assistance from Z. L. The final manuscript was read and approved by all authors.

Conflicts of interest

The authors have no conflict of interest to declare.

Acknowledgements

J. M. C. is grateful for the BASF/Royal Academy of Engineering Research Chair in Data-Driven Molecular Engineering of Functional Materials, which is partly sponsored by the Science and Technology Facilities Council (STFC) *via* the ISIS Neutron and Muon Source. The authors also thank the Argonne Leadership Computing Facility, which is a DOE Office of Science Facility, for use of its research resources, under contract No. DE-AC02-06CH11357.

Notes and references

1 OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad and B. Zoph, *GPT-4 Technical Report*, 2024, **https://arxiv.org/abs/2303.08774**.
2 G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu and O. Vinyals, *Gemini: A Family of Highly Capable Multimodal Models*, 2024, **https://arxiv.org/abs/2312.11805**.
3 A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle and Z. Zhao, *The Llama 3 Herd of Models*, 2024, **https://arxiv.org/abs/2407.21783**.
4 A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix and W. E. Sayed, *Mistral 7B*, 2023, **https://arxiv.org/abs/2310.06825**.
5 G. Deletang, A. Ruoss, P.-A. Duquenne, E. Catt, T. Genewein, C. Mattern, J. Grau-Moya, L. K. Wenliang, M. Aitchison, L. Orseau, M. Hutter and J. Veness, *The Twelfth International Conference on Learning Representations*, 2024.
6 V. Rawte, S. Chakraborty, A. Pathak, A. Sarkar, S. T. I. Tonmoy, A. Chadha, A. Sheth and A. Das, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Singapore, 2023, pp. 2541–2573.
7 J. Lee, T. Le, J. Chen and D. Lee, *Proceedings of the ACM Web Conference 2023*, New York, NY, USA, 2023, p. 3637–3647.
8 J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, *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)*, Minneapolis, Minnesota, 2019, pp. 4171–4186.
9 J. Xu, S. Desai and G. Durrett, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online, 2020, pp. 6275–6281.
10 E. Strubell, A. Ganesh and A. McCallum, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, 2019, pp. 3645–3650.
11 G. Hinton, O. Vinyals and J. Dean, *Distilling the Knowledge in a Neural Network*, 2015, **https://arxiv.org/abs/1503.02531**.
12 M. Zaki, M. Jayadeva and N. M. A. Krishnan, *Digital Discovery*, 2024, **3**, 313–327.
13 T. Xie, Y. Wan, W. Huang, Z. Yin, Y. Liu, S. Wang, Q. Linghu, C. Kit, C. Grazian, W. Zhang, I. Razzak and B. Hoex, *DARWIN Series: Domain Specific Large Language Models for Natural Science*, 2023, **https://arxiv.org/abs/2308.13565**.

14 Y. Song, S. Miret, H. Zhang and B. Liu, *In Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023, pp. 5724–5739.
15 M. C. Swain and J. M. Cole, *J. Chem. Inf. Model.*, 2016, **56**, 1894–1904.
16 J. Mavracic, C. J. Court, T. Isazawa, S. R. Elliott and J. M. Cole, *J. Chem. Inf. Model.*, 2021, **61**, 4280–4289.
17 T. Isazawa and J. M. Cole, *Sci. Data*, 2023, **10**, 651.
18 T. Isazawa and J. M. Cole, *J. Chem. Inf. Model.*, 2022, **62**, 1207–1213.
19 E. J. Beard and J. M. Cole, *Sci. Data*, 2022, **9**, 329.
20 P. Rajpurkar, J. Zhang, K. Lopyrev and P. Liang, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas, 2016, pp. 2383–2392.
21 S. Huang and J. M. Cole, *Sci. Data*, 2020, **7**, 260.
22 O. Sierepeklis and J. M. Cole, *Sci. Data*, 2022, **9**, 648.
23 J. Zhao and J. M. Cole, *Sci. Data*, 2022, **9**, 192.
24 Q. Dong and J. M. Cole, *Sci. Data*, 2022, **9**, 193.
25 C. J. Court and J. M. Cole, *Sci. Data*, 2018, **5**, 1–12.
26 P. Kumar, S. Kabra and J. M. Cole, *Sci. Data*, 2022, **9**, 292.
27 P. Kumar, S. Kabra and J. M. Cole, *Sci. Data*, 2024, **11**, 1273.
28 D. Huang and J. M. Cole, *Sci. Data*, 2024, **11**, 80.
29 E. J. Beard, G. Sivaraman, Á. Vázquez-Mayagoitia, V. Vishwanath and J. M. Cole, *Sci. Data*, 2019, **6**, 307.
30 S. Huang and J. M. Cole, *Chem. Sci.*, 2022, **13**, 11487–11495.
31 J. Qu, Y. R. Xie, K. M. Ciesielski, C. E. Porter, E. S. Toberer and E. Ertekin, *npj Comput. Mater.*, 2024, **10**, 58.
32 A. N. Rubungo, C. Arnold, B. P. Rand and A. B. Dieng, *arXiv*, 2023, preprint, arXiv:2310.14029, DOI: **10.48550/arXiv.2310.14029**.
33 K. M. Jablonka, Q. Ai, A. Al-Feghali, S. Badhwar, J. D. Bocarsly, A. M. Bran, S. Bringuier, L. C. Brinson, K. Choudhary, D. Circi, *et al.*, *Digital Discovery*, 2023, **2**, 1233–1250.
34 Y. Song, S. Miret and B. Liu, in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, Volume 1: Long Papers*, Association for Computational Linguistics, Toronto, Canada, 2023, 1, pp. 3621–3639.
35 J. Geiping and T. Goldstein, *Proceedings of the 40th International Conference on Machine Learning*, 2023.