communications materials

Article

https://doi.org/10.1038/s43246-024-00449-9

Accelerating materials language processing with large language models

Check for updates

Jaewoong Choi [1] & Byungju Lee [1]

Materials language processing (MLP) can facilitate materials science research by automating the extraction of structured data from research papers. Despite the existence of deep learning models for MLP tasks, there are ongoing practical issues associated with complex model architectures, extensive fine-tuning, and substantial human-labelled datasets. Here, we introduce the use of large language models, such as generative pretrained transformer (GPT), to replace the complex architectures of prior MLP models with strategic designs of prompt engineering. We find that in-context learning of GPT models with few or zero-shots can provide high performance text classification, named entity recognition and extractive question answering with limited datasets, demonstrated for various classes of materials. These generative models can also help identify incorrect annotated data. Our GPT-based approach can assist material scientists in solving knowledge-intensive MLP tasks, even if they lack relevant expertise, by offering MLP guidelines applicable to any materials science domain. In addition, the outcomes of GPT models are expected to reduce the workload of researchers, such as manual labelling, by producing an initial labelling set and verifying human-annotations.

Materials language processing (MLP) has emerged as a powerful tool in the realm of materials science research that aims to facilitate the extraction of valuable information from a large number of papers and the development of knowledgebase[1–5]. MLP leverages natural language processing (NLP) techniques to analyse and understand the language used in materials science texts, enabling the identification of key materials and properties and their relationships[6–9]. Some researchers reported that the learning of text-inherent chemical/physical knowledge is enabled by MLP, showing interesting examples that text embedding of chemical elements is aligned with the periodic table[1,2,9–11]. Despite significant advancements in MLP, challenges remain that hinder its practical applicability and performance. One key challenge lies in the availability of labelled datasets for training deep learning-based MLP models, as creating such datasets can be time-consuming and labour-intensive[4,7,9,12,13]. Additionally, developing deep learning models for knowledge-intensive MLP tasks requires exhaustive fine-tuning with a large number of labelled datasets to achieve satisfactory performance, limiting their effectiveness in scenarios with limited labelled data.

In this study, we suggest generative pretrained transformer (GPT) models[14]-enabled MLP guidelines for materials scientists to employ the power of large language models (LLMs) for solving such knowledge-intensive tasks effectively. Recently, GPT-3, and GPT-3.5 models, the powerful LLMs, have demonstrated remarkable performance in various NLP tasks, such as text generation, translation, and comprehension, and has garnered growing interest even in the materials science field[15–17]. We aim to show how to use these GPT models (e.g., embeddings, few-shot learning or fine-tuning) for solving MLP tasks and investigate their characteristics, such as reliability, and generative property, beyond the comparison of performance with existing models. Our study focuses on two key MLP tasks: text classification, and information extraction, and the latter involves two subtasks, i.e., named entity recognition (NER), and extractive question answering (QA).

First, regarding a text classification task, we present a paper filtering method that leverages the strengths of zero-shot (without training data) and few-shot (with few training data) learning models, which show promising performance even with limited training data. This approach demonstrates the potential to achieve high accuracy in filtering relevant documents without fine-tuning based on a large-scale dataset. With regard to information extraction, we propose an entity-centric prompt engineering method for NER, the performance of which surpasses that of previous fine-tuned models on multiple datasets. By carefully constructing prompts that guide the GPT models towards recognising and tagging materials-related entities, we enhance the accuracy and efficiency of entity recognition in materials science texts. Also, we introduce a GPT-enabled extractive QA model that demonstrates improved performance in providing precise and informative answers to questions related to materials science. By fine-tuning

[1]Computational Science Research Center, Korea Institute of Science and Technology, Seoul, Republic of Korea.
e-mail: blee89@kist.re.kr

the GPT model on materials-science-specific QA data, we enhance its ability to comprehend and extract relevant information from the scientific literature.

Through our experiments and evaluations, we validate the effectiveness of GPT-enabled MLP models, analysing their cost, reliability, and accuracy to advance materials science research. Furthermore, we discuss the implications of GPT-enabled models for practical tasks, such as entity tagging and annotation evaluation, shedding light on the efficacy and practicality of this approach. In summary, our research presents a significant advancement in MLP through the integration of GPT models. By leveraging the capabilities of GPT, we aim to overcome limitations in its practical applicability and performance, opening new avenues for extracting knowledge from materials science literature.

Results and Discussion

General workflow of MLP

Figure 1 presents a general workflow of MLP, which consists of data collection, pre-processing, text classification, information extraction and data mining[18]. In Fig. 1, data collection and pre-processing are close to data engineering, while text classification and information extraction can be aided by natural language processing. Lastly, data mining such as recommendations based on text-mined data[2,10,19,20] can be conducted after the text-mined datasets have been sufficiently verified and accumulated. Most MLP studies proceed in a similar flow. This process is actually similar to the process of actual materials scientists obtaining desired information from papers. For example, if they want to get information about the synthesis method of a certain material, they search based on some keywords in a paper search engine and get information retrieval results (a set of papers). Then, valid papers (papers that are likely to contain the necessary information) are selected based on information such as title, abstract, author, and journal. Next, they can read the main text of the paper, locate paragraphs that may contain the desired information (e.g., synthesis), and organize the information at the sentence or word level. Here, the process of selecting papers or finding paragraphs can be conducted through a text classification model, while the process of recognising, extracting, and organising information can be done through an information extraction model. Therefore, this study mainly deals with how text classification and information extraction can be performed through LLMs.

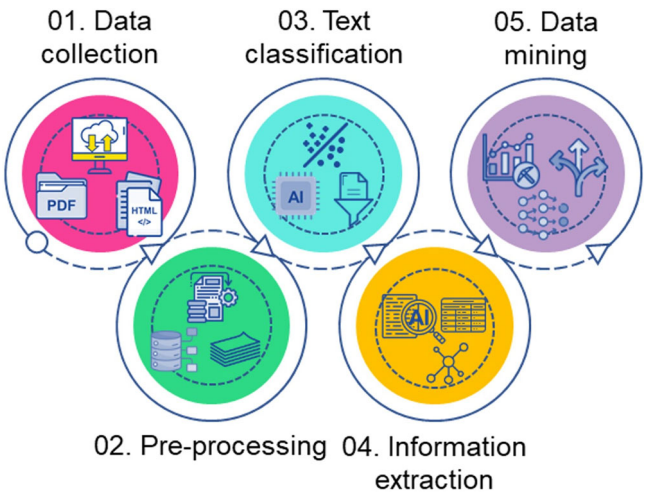


Fig. 1 | General workflow of MLP. The process of MLP consists of five steps; data collection, pre-processing, text classification, information extraction and data mining. Data collection involves the web crawling or bulk download of papers with open API services and sometime requires parsing of mark-up languages such as HTML. Pre-processing is an essential step, and includes preserving and managing the text encoding, identifying the characteristics of the text to be analysed (length, language, etc.), and filtering through additional data. Data collection and pre-processing steps are pre-requisite for MLP, requiring some programming techniques and database knowledge for effective data engineering. Text classification and information extraction steps are of our main focus, and their details are addressed in Section 3,4, and 5. Data mining step aims to solve the prediction, classification or recommendation problems from the patterns or relationships of text-mined dataset. After the data set extracted from the paper has been sufficiently verified and accumulated, the data mining step can be performed for purposes such as material discovery.

Text classification in MLP

Text classification, a fundamental task in NLP, involves categorising textual data into predefined classes or categories[21]. This process enables efficient organisation and analysis of textual data, offering valuable insights across diverse domains. With wide-ranging applications in sentiment analysis, spam filtering, topic classification, and document organisation, text classification plays a vital role in information retrieval and analysis. Traditionally, manual feature engineering coupled with machine-learning algorithms were employed; however, recent developments in deep learning and pre-trained LLMs, such as GPT series models, have revolutionised the field. By fine-tuning these models on labelled data, they automatically extract features and patterns from text, obviating the need for laborious manual feature engineering.

In the field of materials science, text classification has been actively used for filtering valid documents from the retrieval results of search engines or identifying paragraphs containing information of interest[9,12,13]. For example, some researchers have attempted to classify the abstracts of battery-related papers from the results of searching with keywords such as 'battery' or 'battery materials', which is the starting point of extracting battery-device information from the literature[22]. Furthermore, paragraph-level classification models have been developed to find paragraphs of interest using a statistical model such as Latent Dirichlet allocation or machine-learning models such as random forest or BERT classifier[13,23,24], e.g., for solid-state synthesis, gold-nanoparticle synthesis, multiclass of solution synthesis.

Information extraction in MLP

Information extraction is an NLP task that involves automatically extracting structured information from unstructured text[25–28]. The goal of information extraction is to convert text data into a more organized and structured form that can be used for analysis, search, or further processing. Information extraction plays a crucial role in various applications, including text mining, knowledge graph construction, and question-answering systems[29–33]. Key aspects of information extraction in NLP include NER, relation extraction, event extraction, open information extraction, coreference resolution, and extractive question answering.

Named entity recognition in MLP. First, NER is one of the representative NLP techniques for information extraction[34]. NER aims to identify and classify named entities within text. Here, named entities refer to real-world objects such as persons, organisations, locations, dates, and quantities[35]. The task of NER involves analysing text and identifying spans of words that correspond to named entities. NER algorithms typically use machine learning such as recurrent neural networks or transformers to automatically learn patterns and features from labelled training data. NER models are trained on annotated datasets where human annotators label entities in text. These annotations serve as the ground truth for training the model. The model learns to recognise patterns and contextual cues to make predictions on unseen text, identifying and classifying named entities. The output of NER is typically a structured representation of the recognised entities, including their type or category.

In the field of materials science, many researchers have developed NER models for extracting structured summary-level data from unstructured text. For example, domain-specific pretrained language models such as SciBERT[36], MatBERT[8], MatSciBERT[30], and MaterialsBERT[37] were used to extract specialised information from materials science literature, thereby extracting entities on solid-state materials, doping, gold nanoparticles (AuNPs), polymers, electrocatalytic CO_2 reduction, and solid oxide fuel cells from a large number of papers[8,9,37–39].

Extractive question answering in MLP. Extractive QA is a type of QA system that retrieves answers directly from a given passage of text rather than generating answers based on external knowledge or language understanding[40]. It focuses on selecting and extracting the most relevant information from the passage to provide concise and accurate answers to specific questions. Extractive QA systems are commonly built using machine-learning techniques, including both supervised and unsupervised methods. Supervised learning approaches often require human-labelled training data, where questions and their corresponding answer spans in the passage are annotated. These models learn to generalise from the labelled examples to predict answer spans for new unseen questions. Extractive QA systems have been widely used in various domains, including information retrieval, customer support, and chatbot applications. Although they provide direct and accurate answers based on the available text, they may struggle with questions that require a deeper understanding of context or the ability to generate answers beyond the given passage.

In the materials science field, the extractive QA task has received less attention as its purpose is similar to the NER task for information extraction, although battery-device-related QA models have been proposed[22]. Nevertheless, by enabling accurate information retrieval, advancing research in the field, enhancing search engines, and contributing to various domains within materials science, extractive QA holds the potential for significant impact.

Paper classification with LLMs

To explain how to classify papers with LLMs, we used the binary classification dataset from a previous MLP study to construct a battery database using NLP techniques applied to research papers[22].

Text classification dataset description. The authors reported a dataset specifically designed for filtering papers relevant to battery materials research[22]. Specifically, 46,663 papers are labelled as 'battery' or 'non-battery', depending on journal information (Supplementary Fig. 1a). Here, the ground truth refers to the papers published in the journals related to battery materials among the results of information retrieval based on several keywords such as 'battery' and 'battery materials'. The original dataset consists of training set (70%; 32,663), validation set (20%; 9333) and test set (10%; 4667), and its specific examples can be found in Supplementary Table 4. The dataset was manually annotated and a classification model was developed through painstaking fine-tuning processes of pre-trained BERT-based models.

Despite the reported SOTA performance is an accuracy of 97.5%, precision of 96.6%, and recall of 99.5%, such models require extensive training data and complex structures, and thus, we attempted to develop a simple, GPT-enabled model that can achieve high performance using only a small dataset. Specifically, we tested zero-shot learning with GPT Embeddings model. For few-shot learning models, both GPT 3.5 and GPT-4 were tested, while we also evaluated the performance of fine-tuning model of GPT-3 for the classification task (Supplementary Table 1). In these experiments, we focused on the accuracy to enhance the balanced performance in improving the true and false accuracy rates. The choice of metrics to prioritize in text classification tasks varies based on the specific context and analytical goals. For example, if the goal is to maximize the retrieval of relevant papers for a specific category, emphasizing recall becomes crucial. Conversely, in document filtering, where reducing false positives and ensuring high purity is vital, prioritizing precision becomes more significant. When striving for comprehensive classification performance, employing accuracy metrics might be more appropriate.

Zero-shot learning with LLMs for text classification. Zero-shot learning with embedding[41,42] allows models to make predictions or perform tasks without fine-tuning with human-labelled data. The zero-shot model works based on the embedding value of a given text, which is provided by GPT embedding modules. Using the distance between a given paragraph and predefined labels in the embedding space, which numerically represent their semantic similarity, paragraphs are classified with labels (Fig. 2a). For example, if one uses the model to classify an unseen text with the label of either 'batteries' or 'solar cells', the model will calculate the distance between the embedding value of the text and that of 'batteries' or 'solar cells', selecting the label with higher similarity in the embedding space.

Below are the results of the zero-shot text classification model using the text-embedding-ada-002 model of GPT Embeddings. First, we tested the original label pair of the dataset[22], that is, 'battery' vs. 'non-battery' ('original labels' of Fig. 2b). The performance of the existing label-based model was low, with an accuracy and precision of 63.2%, because the difference between the embedding value of two labels was small. Considering that the true label should indicate battery-related papers and the false label would result in the complementary dataset, we designed the label pair as 'battery materials' vs. 'diverse domains' ('crude labels' of Fig. 2b). We successfully improved the performance, achieving an accuracy of 87.3%, precision of 84.5%, and recall of 97.9%, by specifying the meaning of the false label.

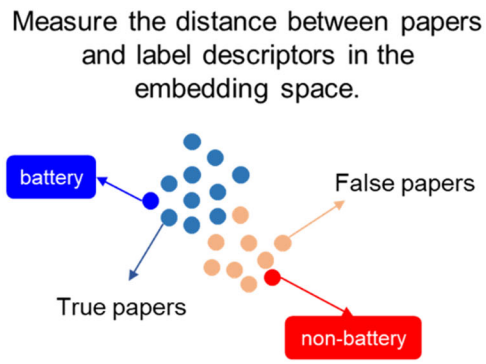
To further reduce the number of false positives, we designed the labels in an explicit manner, i.e., 'battery materials' vs. 'medical and psychological research' ('designated labels' of Fig. 2b). Here, the false label was selected by checking the titles of randomly sampled papers from the non-battery set (refer to Supplementary Table 4). Interestingly, we obtained slightly improved performance (accuracy, recall, and precision of 91.0%, 88.6%, and 98.3%). We were able to achieve even higher performance (ACC: 93.0, PRE: 90.8, REC: 98.9) if the labels were made even more verbose: 'papers related to battery energy materials' vs. 'medical and psychological research' ('verbose labels' of Fig. 2b). Although these values are relatively lower than those of the SOTA model, it is noteworthy that acceptable text-classification performance was achieved without exhaustive human labelling, as the proposed model is based on zero-shot learning with embeddings. These results imply that classifying a specific set among the paper data set in materials science can be achieved without labelling with zero-shot methods if a proper label corresponding to a representative embedding value for each category is selected. When utilizing our label descriptions for zero-shot learning, some papers may exactly fit into neither the positive nor negative labels, that is, outliers. Nevertheless, they will be assigned to a label that is relatively similar to one of the given categories.

Few-shot learning and fine-tuning of LLMs for text classification. Next, the improved performance of few-shot text classification models is demonstrated in Fig. 2c. In few-shot learning models, we provide the limited number of labelled datasets to the model. We tested 2-way 1-shot and 2-way 5-shot models, which means that there are two labels and one/five labelled data for each label are granted to the GPT-3.5 models ('text-davinci-003'). The example prompt is given in Fig. 2d. The 2-way 1-shot models resulted in an accuracy of 95.7%, which indicates that providing just one example for each category has a significant effect on the prediction. Furthermore, increasing the number of examples (2-way 5-shots models) leads to improved performance, where the accuracy, precision, and recall are 96.1%, 95.0%, and 99.1%. Particularly, we were able to find the slightly improved performance in using GPT-4 ('gpt-4-0613') than GPT-3.5 ('text-davinci-003'); the precision and accuracy increased from 0.95 to 0.954 and from 0.961 to 0.963, respectively.

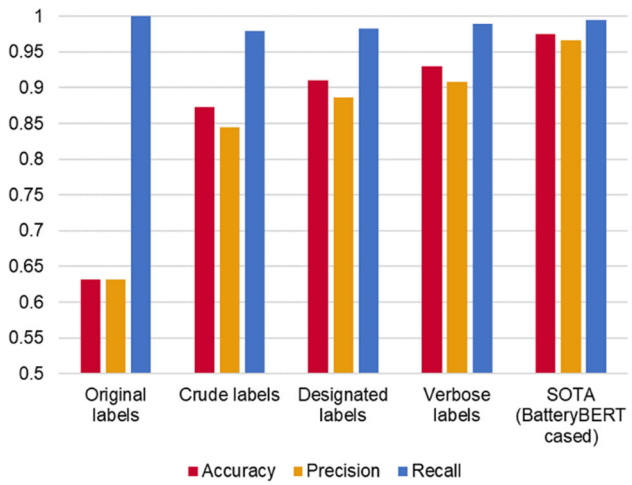
In addition, we used the fine-tuning module of the davinci model of GPT-3 with 1000 prompt–completion examples. The fine-tuning model performs a general binary classification of texts by learning the examples while no longer using the embeddings of the labels, in contrast to few-shot learning. In our test, the fine-tuning model yielded high performance, that is, an accuracy of 96.6%, precision of 95.8%, and recall of 98.9%, which are close to those of the SOTA model. Here, we emphasise that the GPT-enabled models can achieve acceptable performance even with the small number of datasets, although they slightly underperformed the BERT-based model trained with a large dataset. The summary of our results comparing the GPT-based models against the SOTA models on three tasks are reported in Supplementary Table 1.

(a)

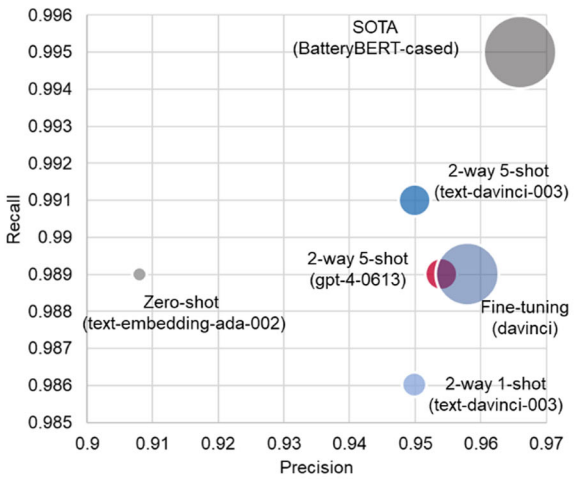
Label descriptor	TRUE	FALSE
Original	battery	non-battery
Crude	battery material	diverse domains
Designated	battery material	medical and psychological research
Verbose	papers related to battery energy materials	medical and psychological research



(b)



(c)



(d)

Prompt

The task is to classify whether a given document is TRUE or not. Examples of battery energy materials are given below.

Input: the importance of ni - based batteries in the present context is extremely valuable, as their electrochemical properties are expected to offer significant insights into their application as battery materials. …
Output: True

Input: introduction contact dermatitis to cosmetics is a common problem in the general population, although its prevalence appears to be underestimated. we reviewed cases of allergic contact dermatitis to cosmetics diagnosed in our dermatology department …
Output: False

Input:The electrochemical performance of NaCrO2 as a positive electrode material for an intermediate-temperature sodium secondary battery was evaluated …
Output:

Fig. 2 | Results of GPT-enabled text classification models. a Overall process of our zero-shot learning for text classification. **b** Results of zero-shot learning with GPT embedding. The accuracy, precision, and recall are reported. **c** Comparison of zero-shot learning (GPT Embeddings), few-shot learning (GPT-3.5 and GPT-4), and fine-tuning (GPT-3) results. The horizontal and vertical axes are the precision and recall of each model, respectively. The node colour and size are based on the rank of accuracy and the dataset size, respectively. **d** Example of prompt engineering for 2-way 1-shot learning, where the task description, one example for each category, and input abstract are given.

Understanding the calibration of LLMs in text classification. In addition to the accuracy, we investigated the reliability of our GPT-based models and the SOTA models in terms of calibration. The reliability can be evaluated by measuring the expected calibration error (ECE) score[43] with 10 bins. A lower ECE score indicates that the model's predictions are closer to being well-calibrated, ensuring that the confidence of a model in its prediction is similar to the actual accuracy of the model[44,45] (Refer to Methods section). The log probabilities of GPT-enabled models were used to compare the accuracy and confidence. The ECE score of the SOTA ('BatteryBERT-cased') model is 0.03, whereas those of the 2-way 1-shot model, 2-way 5-shot model, and fine-tuned model were 0.05, 0.07, and 0.07, respectively. Considering a well-calibrated model typically exhibits an ECE of less than 0.1, we conclude that our GPT-enabled text classification models provide high performance in terms of both accuracy and reliability with less cost. The lowest ECE score of the SOTA model shows that the BERT classifier fine-tuned for the given task was well-trained and not overconfident, potentially owing to the large and unbiased training set. The GPT-enabled models also show acceptable reliability scores, which is encouraging when considering the amount of training data or training costs required. In summary, we expect the GPT-enabled text-classification models to be valuable tools for materials scientists with less machine-learning knowledge while providing high accuracy and reliability comparable to BERT-based fine-tuned models.

Extraction of named entities with LLMs

To explain how to extract named entities from materials science papers with GPT, we prepared three open datasets, which include human-labelled entities on solid-state materials, doped materials, and AuNPs (Supplementary Table 2).

Extracting solid-state materials entities with LLMs. The solid-state materials dataset includes 800 annotated abstracts with the following categories: inorganic materials (MAT), symmetry/phase labels (SPL), sample descriptors (DSC), material properties (PRO), material applications (APL), synthesis methods (SMT), and characterisation methods (CMT)[38]. For example, MAT indicates inorganics solid/alloy materials or non-gaseous elements such as '$BaTiO_3$,' 'titania,' or 'Fe'. SPL indicates the name for crystal structures and phases such as 'tetragonal' or a symmetry label such as 'Pbnm' (Supplementary Fig. 1b). The original dataset consists of training/validation/test at a ratio of 6:2:2, which is used for fine-tuning of GPT models.

Because the fine-tuning model requires prompt–completion examples as a training set, the NER datasets are pre-processed as follows: the annotations for each category are marked with the special tokens[46], and then, the raw text and marked text are used as the prompt and completion, respectively. For example, if the input text is "LiCoO2 and LiFePO4 are used as cathodes of secondary batteries", the prompt is the same as the input text, and the completion for each category is as follows:

> MAT model → Completion: "LiCoO2 and LiFePO4 are used as cathodes of secondary batteries" / completion: "@@LiCoO2## and @@LiFePO4## are used as cathodes of secondary batteries."
>
> APL model → Completion: "LiCoO2 and LiFePO4 are used as cathodes of secondary batteries" / completion: "LiCoO2 and LiFePO4 are used as @@cathodes of secondary batteries##."

One of the examples used in the training set is shown in Fig. 3d. After pre-processing, we tested fine-tuning modules of GPT-3 ('davinci') models. The performance of our GPT-enabled NER models was compared with that of the SOTA model in terms of recall, precision, and F1 score. Figure 3a shows that the GPT model exhibits a higher recall value in the categories of CMT, SMT, and SPL and a slightly lower value in the categories of DSC, MAT, and PRO compared to the SOTA model. However, for the F1 score, our GPT-based model outperforms the SOTA model for all categories because of the superior precision of the GPT-enabled model (Fig. 3b, c). The high precision of the GPT-enabled model can be attributed to the generative nature of GPT models, which allows coherent and contextually appropriate output to be generated. Excluding categories such as SMT, CMT, and SPL, BERT-based models exhibited slightly higher recall in other categories. The lower recall values could be attributed to fundamental differences in model architectures and their abilities to manage data consistency, ambiguity, and diversity, impacting how each model comprehends text and predicts subsequent tokens. BERT-based models effectively identify lengthy and intricate entities through CRF layers, enabling sequence labelling, contextual prediction, and pattern learning. The use of CRF layers in prior NER models has notably improved entity boundary recognition by considering token labels and interactions. In contrast, GPT-based models focus on generating text containing labelling information derived from the original text. As a generative model, GPT doesn't explicitly label text sections but implicitly embeds labelling details within the generated text. This approach might hinder GPT models in fully grasping complex contexts, such as ambiguous, lengthy, or intricate entities, leading to lower recall values.

Extracting doped materials entities with LLMs. The doped materials entity dataset[8] includes 450 annotations on the base material (BASEMAT), the doping agent (DOPANT), and quantities associated with the doped material such as the doping density or the charge carrier density (DOPMODQ), with specific examples provided in Supplementary Fig. 1c. The original dataset consists of training/validation/test set at a ratio of 8:1:1. The SOTA model ('MatBERT-uncased') for this dataset had F1 scores of 72, 82, and 62 for BASEMANT, DOPANT, and DOPMODQ, respectively. We analysed this dataset using fine-tuning modules of GPT-3 such as the 'davinci' model with the same data composition.

The prompt–completion sets were constructed similarly to the previous NER task. As reported in Fig. 4a, the fine-tuning of 'davinci' model showed high precision of 93.4, 95.6, and 92.7 for the three categories, BASEMAT, DOPANT, and DOPMODQ, respectively, while yielding relatively lower recall of 62.0, 64.4, and 59.4, respectively (Fig. 4a). These results imply that the doped materials entity dataset may have diverse entities for each category but that there is not enough data for training to cover the diversity. In addition, the GPT-based model's F1 scores of 74.6, 77.0, and 72.4 surpassed or closely approached those of the SOTA model ('MatBERT-uncased'), which were recorded as 72, 82, and 62, respectively (Fig. 4b).

Extracting AuNPs entities with LLMs. The AuNPs entity dataset annotates the descriptive entities (DES) and the morphological entities (MOR)[23], where DES includes 'dumbbell-like' or 'spherical' and MOR includes noun phrases such as 'nanoparticles' or 'AuNRs'. More specific examples are provided in Supplementary Fig. 1d. The SOTA model for this dataset is reported as the MatBERT-based model whose F1 scores for DES and MOR are 0.67 and 0.92, respectively[8].

Instead of adopting fine-tuning, we used the few-shot learning[47] of the GPT-3.5 model ('text-davinci-003') for the AuNPs entities dataset, as there are not sufficient datasets ($N = 85$). Similar to the previous NER task, we designed three prompts such as random retrieval, task-informed random retrieval and kNN retrieval (Fig. 4 and Supplementary Table 2). First, we randomly select the three ground-truth examples (i.e., pair of text and the text with named entities) from the original training and validation set when extracting the named entities from the given text in the test set (random retrieval). These simple methods yield high recall performance of 63% and 97% for the DES and MOR categories, respectively. Here, it is noteworthy that prompts with the ground-truth examples can provide improved results on DES and MOR entity recognition, considering the recall values of 52% and 64% reported in prior works[23] (Supplementary Fig. 2). However, the F1 score of this few-shot learning model was lower than that of the SOTA model ('random retrieval' of Fig. 4c). Furthermore, we tested the effect of adding a phrase that directly specifies the task to the existing prompt; e.g., 'The task is to extract the descriptive entities of materials in the given text' ('task-informed random retrieval' of Fig. 4c). The example prompt is shown in Fig. 4d. Some performance improvements, namely a 1%–2% increase in recall and a 6%–11% increase in precision, were observed.

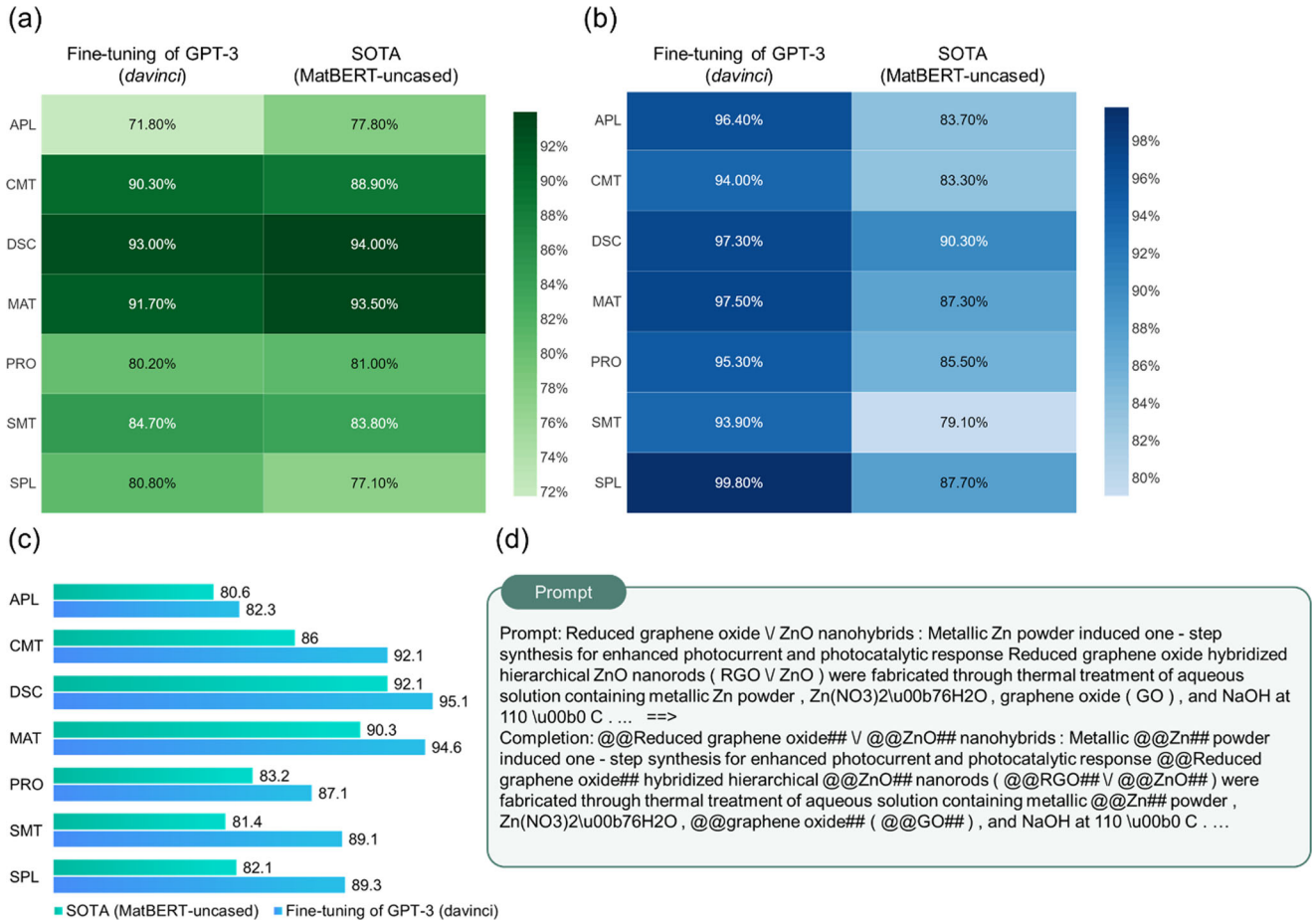


Fig. 3 | Performance of GPT-enabled NER models on solid-state materials compared to the SOTA model ('MatBERT-uncased'). The proposed models are based on fine-tuning modules based on prompt–completion examples. **a–c** Comparison of recall, precision, and F1 score between our GPT-enabled model and the SOTA model for each category. **d** Example of prompt–completion for MAT entity recognition.

Finally, to more elaborately perform the few-shot learning, 'similar' ground-truth examples to each test set, that is, the examples for which the document embedding value are similar to that of each test set, were selected for the NER extraction in the test set ('kNN retrieval' of Fig. 4c). Interestingly, compared to the performance of the previous method (i.e., task-informed random retrieval), we confirmed that the recall value of the kNN method was the same or slightly lower and that the precision increased by 15%–20% (Supplementary Table 2 and Supplementary Fig. 2). Particularly, the recall of DES was relatively low compared to its precision, which indicates that providing similar ground-truth examples enables more tight recognition of DES entities. In addition, the recall of MOR is relatively higher than the precision, implying that giving k-nearest examples results in the recognition of more permissive MOR entities. In summary, we confirmed the potential of the few-shot NER model through GPT prompt engineering and found that providing similar examples rather than randomly sampled examples and informing tasks had a significant effect on performance improvement. In terms of the F1 score, few-shot learning with the GPT-3.5 ('text-davinci-003') model results in comparable MOR entity recognition performance as that of the SOTA model and improved DES recognition performance (Fig. 4c). In addition, we applied the same prompting strategy for GPT-4 model (gpt-4-0613), and obtained the improved performance in capturing MOR and DES entities.

Extraction of answers to questions with LLMs

To explain how to extract answer to questions with GPT, we prepared battery device-related question answering dataset[22].

Few-shot learning and fine-tuning of GPT models for extractive QA. This dataset consists of questions, contexts, and answers, and the questions are related to the principal components of battery systems, i.e., 'What is the anode?', 'What is the cathode?', and 'What is the electrolyte?'. For example, the context is the raw text such as "The blended slurry was then cast onto a clean current collector (Al foil for the cathode and Cu foil for the anode) and dried at 90 °C under vacuum overnight" and the answer to the question what a cathode is can be 'Al foil'. This dataset was proposed to train the deep learning models to identify the battery system component, which can be extended based on battery literature[48–50]. The publicly available dataset includes 427 annotations, which is generated by battery experts but requires several pre-processing[22]. We also found redundant or incorrect annotations, e.g., when there is no mention of the anode in the given context, the question is about the anode and the answer is about the cathode. In the end, we refined the given dataset into 331 QA data (anode: 90; cathode: 161; electrolyte: 80) based on the outcomes of GPT-enabled models.

Also, we reproduced the results of prior QA models including the SOTA model, 'BatteryBERT (cased)', to compare the performances between our GPT-enabled models and prior models with the same measure. The performances of the models were newly evaluated with the average values of token-level precision and recall, which are usually used in QA model evaluation. In this way, the prior models were re-evaluated, and the SOTA model turned out to be 'BatteryBERT (cased)', identical to that reported (Fig. 5a).

We tested the zero-shot QA model using the GPT-3.5 model ('text-davinci-003'), yielding a precision of 60.92%, recall of 79.96%, and F1 score of 69.15% (Fig. 5b and Supplementary Table 3). These relatively low

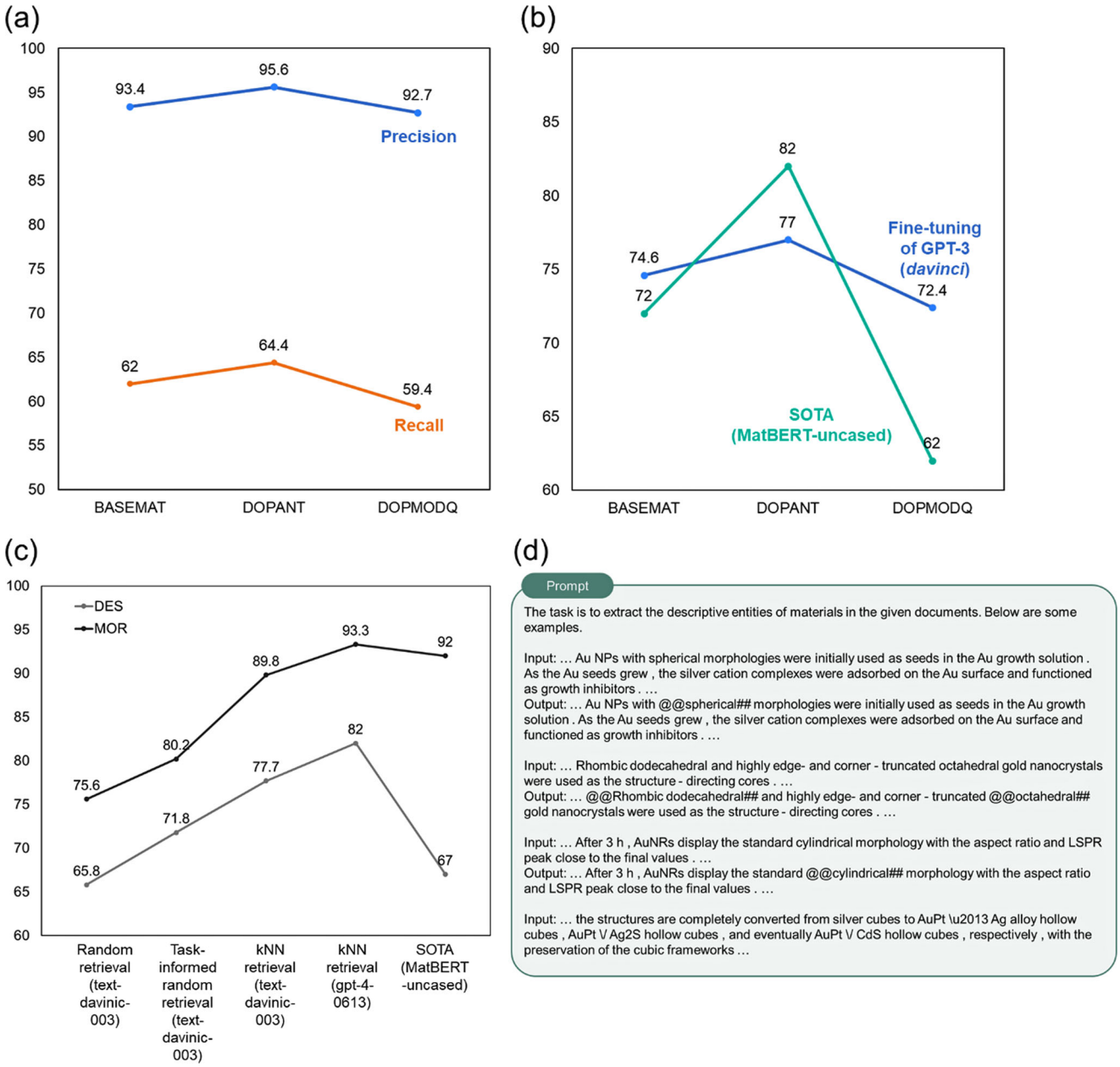
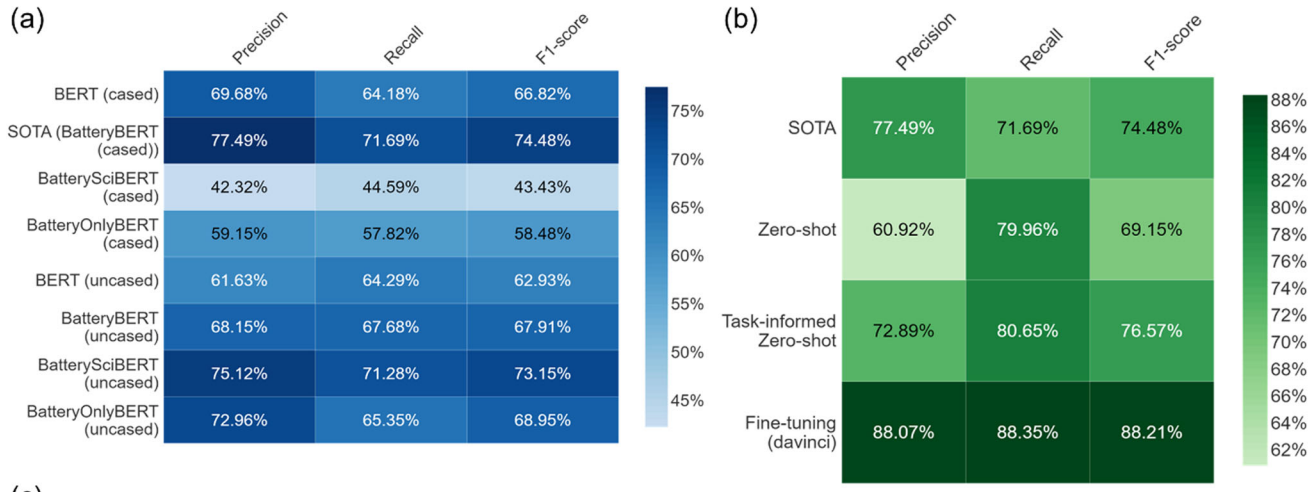


Fig. 4 | Performance of GPT-enabled NER models on doped materials and AuNPs, compared to the SOTA model. a Doped materials entity recognition performance of fine-tuning of GPT-3 (*davinci*), **b** doped materials entity recognition performance (F1 score) comparison between SOTA ('MatBERT-uncased') and fine-tuning of GPT-3 (*davinci*), **c** AuNPs entity recognition performance (F1-score) comparisons between GPT 3.5 davinci (random retrieval, task-informed random retrieval, kNN retrieval) and SOTA ('MatBERT-uncased') model, **d** Example of prompt for DES entity recognition (task informed random retrieval).

performance values can be derived from the domain-specific dataset, from which it is difficult for a vanilla model to find the answer from the given scientific literature text. Therefore, we added a task-informing phrase such as 'The task is to extract answers from the given text.' to the existing prompt consisting of the question, context, and answer. Surprisingly, we observed an increase in performance, particularly in precision, which increased from 60.92% to 72.89%. By specifying that the task was to extract rather than generate answers, the accuracy of the answers appeared to increase. Next, we tested a fine-tuning module of GPT-3 models ('davinci'). We achieved higher performance with an F1 score of 88.21% (compared to that of 74.48% for the SOTA model).

Understanding the generative property of GPT. In addition to the improved performance, we were able to examine the possibility of correcting the existing annotations with our GPT-based models. As mentioned earlier, we modified and used the open QA data set. Here, in addition to removing duplicates or deleting unanswered data, finding data with incorrect answers was based on the results of the GPT model (Fig. 5c). For example, there is an incorrect question–answer pair: the anode materials are not mentioned in the given context and 'nano-meshed' is mentioned as the cathode material; however, the annotated question is 'what is the anode material?', and the corresponding answer is 'nano-meshed'. For this case, most BERT-based models yield the answer 'nano-meshed' similar to the annotation, whereas the GPT models provide the answer 'the anode is not mentioned in the given text'. In addition, there were annotations that could increase the confusion of the model by making each question–answer pair for the answer in which the two tokens were combined by OR. For example, GPT models answered "sulfur or air cathode", but the original annotations annotate 'sulfur' and 'air' as different answers.



Model	Answer
GPT-3 Fine-tuning (davinci)	***"The anode is not mentioned in the given text."***
BatteryBERT(cased)	*"nanomesh-based cathodes"*
BatteryOnlyBERT(uncased)	*"MnO2"*
BatterySciBERT(uncased)	*"nanomesh"*
BatteryBERT(uncased)	*"nanomesh-based cathodes"*
BatterySciBERT(cased)	*"nanomesh-based cathodes"*
BERT (cased)	*"nanomesh-based cathodes"*
BatteryOnlyBERT(cased)	*"nanomesh"*
BERT (uncased)	*"nanomesh-based cathodes"*

Fig. 5 | Performance of GPT-enabled QA model. a Reproduced results of BERT-based model performances, **b** comparison between the SOTA and fine-tuning of GPT-3 (*davinci*), **c** correction of wrong annotations in QA dataset, and prediction result comparison of each model. Here, the difference in the cased/uncased version of the BERT series model is the processing of capitalisation of tokens or accent markers, which influenced the size of vocabulary, pre-processing, and training cost.

Conclusion

This work presents a GPT-enabled pipeline for MLP tasks, providing guidelines for text classification, NER, and extractive QA. Through an empirical study, we demonstrated the advantages and disadvantages of GPT models in MLP tasks compared to the prior fine-tuned models based on BERT.

In text classification, we conclude that the GPT-enabled models exhibited high reliability and accuracy comparable to that of the BERT-based fine-tuned models. This GPT-based method for text classification is expected to reduce the burden of materials scientists in preparing a large training set by manually classifying papers. Next, in NER tasks, we found that providing similar examples improves the entity-recognition performance in few-shot GPT-enabled NER models. These findings indicate that the GPT-enabled NER models are expected to replace the complex traditional NER models, which requires a relatively large amount of training data and elaborate fine-tuning tasks. Lastly, regarding extractive QA models for battery-device information extraction, we achieved an improved F1 score compared with prior models and confirmed the possibility of using GPT models for correcting incorrect QA pairs. Recently, several pioneering studies have showed the possibility of using LLMs such as chatGPT for extracting information from materials science texts[15,51–53]. In this regard, our novelty lies in comparing the characteristics of GPT series models with the BERT-based fine-tuned models in depth as well as introducing various strategies such as embedding, zero-shot/few-shot learning, and fine-tuning for each MLP task.

We note the potential limitations and inherent characteristics of GPT-enabled MLP models, which materials scientists should consider when analysing literature using GPT models. First, considering that GPT series models are generative, the additional step of examining whether the results are faithful to the original text would be necessary in MLP tasks, particularly information-extraction tasks[15,16]. In contrast, general MLP models based on fine-tuned LLMs do not provide unexpected prediction values because they are classified into predefined categories through cross entropy function. Given that GPT is a closed model that does not disclose the training details and the response generated carries an encoded opinion, the results are likely to be overconfident and influenced by the biases in the given training data[54]. Therefore, it is necessary to evaluate the reliability as well as accuracy of the results when using GPT-guided results for the subsequent analysis. In a similar vein, as GPT is a proprietary model that will be updated over time by openAI, the absolute value of performance can be changed and thus continuous monitoring is required for the subsequent uses[55]. Finally, the GPT-enabled model would face challenges in more domain-specific, complex, and challenging tasks (e.g., relation extraction, event detection, and event extraction) than those presented in this study, as it is difficult to explain the tasks in the prompt. For example, extracting the relations of entities would be challenging as it is necessary to explain well the complicated patterns or relationships as text, which are inferred through black-box models in general NLP models[15,16,56]. Nonetheless, GPT models will be effective MLP tools by allowing materials scientists to more easily analyse literature effectively without knowledge of the complex architecture of existing NLP models[17]. As LLM technologies advance, creating quality prompts that consist of specific and clear task descriptions, appropriate input text for the task, and consistently labelled results (i.e., classification categories) will become more important for materials scientists.

Methods

Data processing

We used the python library 'openai' to implement the GPT-enabled MLP pipeline. We mainly used the prompt–completion module of GPT models for training examples for text classification, NER, or extractive QA. We used zero-shot learning, few-shot learning or fine-tuning of GPT models for MLP task. Herein, the performance is evaluated on the same test set used in prior studies, while small number of training data are sampled from the training set and validation set and used for few-shot learning or fine-tuning of GPT models.

Given a sufficient dataset of prompt–completion pairs, a fine-tuning module of GPT-3 models such as 'davinci' or 'curie' can be used. The prompt–completion pairs are lists of independent and identically distributed training examples concatenated together with one test input. Herein, as open datasets used in this study had training/validation/test separately, we used parts of training/validation for training fine-tuning models and the whole test set to confirm the general performance of models. Otherwise, for few-shot learning which makes the prompt consisting of the task-informing phrase, several examples and the input of interest, can be alternatives. Here, which examples to provide is important in designing effective few-shot learning. Similar examples can be obtained by calculating the similarity between the training set for each test set. That is, given a paragraph from a test set, few examples similar to the paragraph are sampled from training set and used for generating prompts. Specifically, our kNN method for similar example retrieval is based on TF-IDF similarity (refer to Supplementary Fig. 3). Lastly, in case of zero-shot learning, the model is tested on the same test set of prior models.

Regarding the preparation of prompt–completion examples for fine-tuning or few-shot learning, we suggest some guidelines. Suffix characters in the prompt such as ' →' are required to clarify to the fine-tuned model where the completion should begin. In addition, suffix characters in the prompt such as ' \n\n###\n\n' are required to specify the end of the prediction. This is important when a trained model decides on the end of its prediction for a given input, given that GPT is one of the autoregressive models that continuously predicts the following text from the preceding text. That is, in prediction, the same suffix should be placed at the end of the input. In addition, prefix characters are usually unnecessary as the prompt and completion are distinguished. Rather than using the prefix characters, simply starting the completion with a whitespace character would produce better results due to the tokenisation of GPT models. In addition, this method can be economical as it reduces the number of unnecessary tokens in the GPT model, where fees are charged based on the number of tokens. We note that the maximum number of tokens in a single prompt–completion is 4097, and thus, counting tokens is important for effective prompt engineering; e.g., we used the python library 'titoken' to test the tokenizer of GPT series models.

GPT model usage guidelines

After pre-processing, the splitting process of train, validation, and test set was conducted with the same random seed and ratio used in previous studies, that is, the training/validation set is used for fine-tuning GPT models and test set for confirming their general performances. In the fine-tuning of GPT models, there are some hyperparameters such as the base model, batch size, number of epochs, learning rate multiplier, and prompt loss weight. The base models for which fine-tuning is available are GPT-3 models such as 'ada', 'babbage', 'curie', and 'davinci', which can be tested using the web service provided by OpenAI (https://gpttools.com/comparisontool). For a simple prompt–completion task such as zero-shot learning and few-shot learning, GPT-3.5 models such as 'text-davinci-003' can be used. The batch size can be dynamically configured and its maximum is 256; however, we recommend 1% or 0.2% of the training set. The learning rate multiplier adjusts the models' weights during training, and a high learning rate leads to a sub-optimal solution, whereas a low one causes the model to converge too slowly or find a local minimum. The default values are 0.05–0.2 depending on the batch size, and we set the learning rate multiplier as 0.01. The prompt loss weight is the weight to use for loss on the prompt tokens, which should be reduced when prompts are relatively long to the corresponding completions to avoid giving undue priority to prompt learning over the completion learning. We set the prompt loss weight as 0.01.

With the fine-tuned GPT models, we can infer the completion for a given unseen dataset that ends with the pre-defined suffix, which are not included in training set. Here, some parameters such as the temperature, maximum number of tokens, and top P can be determined according to the purpose of analysis. First, temperature determines the randomness of the completion generated by the model, ranging from 0 to 1. For example, higher temperature leads to more randomness in the generated output, which can be useful for exploring creative or new completions (e.g., generative QA). In addition, lower temperature leads to more focused and deterministic generations, which is appropriate to obtain more common and probable results, potentially sacrificing novelty. We set the temperature as 0, as our MLP tasks concern the extraction of information rather than the creation of new tokens. The maximum number of tokens determines how many tokens to generate in the completion. If the ideal completion is longer than the maximum number, the completion result may be truncated; thus, we recommend setting this hyperparameter to the maximum number of tokens of completions in the training set (e.g., 256 in our cases). In practice, the reason the GPT model stops producing results is ideally because a suffix has been found; however, it could be that the maximum length is exceeded. The top P is a hyperparameter about the top-p sampling, i.e., nucleus sampling, where the model selects the next word based on the most likely candidates, limited to a dynamic subset determined by a probability threshold (p). This parameter promotes diversity in generated text while allowing control over randomness.

Performance evaluation

We evaluated the performance of text classification, NER, and QA models using different measures. The fine-tuning module provides the results of accuracy, actually the exact-matching accuracy. Therefore, post-processing of the prediction results was required to compare the performance of our GPT-based models and the reported SOTA models. For the text classification, the predictions refer to one of the pre-defined categories. By comparing the category mentioned in each prediction and the ground truth, the accuracy, precision, and recall can be measured. For the NER, the performance such as the precision and recall can be measured by comparing the index of ground-truth entities and predicted entities. Here, the performance can be evaluated strictly by using an exact-matching method, where both the start index and end index of the ground-truth answer and prediction result match. The boundaries of named entities are likely to be subjective or ambiguous in practice, and thus, we recommend the boundary-relaxation method to generously evaluate the performance, where a case that either the start or end index is correct is considered as a true positive[57,58]. For the extractive QA, the performance is evaluated by measuring the precision and recall for each answer at the token level and averaging them. Similar to the NER performance, the answers are evaluated by measuring the number of tokens overlapping the actual correct answers.

ECE score calculation

To compare the reliability of text classification models in this study, we used ECE score, which assesses the calibration of probabilistic predictions of models. To calculate the ECE score, the following steps are typically taken. First, predictions and true labels are collected. These predictions are class probabilities, not just labels. For each data point, the predicted probability distribution over the possible classes and the true class label are required. Second, based on the predefined number of bins (i.e., M, typically 10-20 bins), similar predicted probabilities are grouped together to analyse calibration within each bin. Next, for each bin, expected accuracy and average confidence are calculated. The expected accuracy is the average of the true accuracy for all data points in each bin; for example, the true accuracy would be 1 if the predicted class matches the true class and 0 otherwise. The average

confidence is the confidence level of the model's predictions within each bin. Also, the relative frequency of data points should be calculated for each bin, by dividing the number of data points in each bin by the total number of data points. Finally, the ECE score is calculated as the weighted average of the absolute difference between the expected accuracy and the average confidence within each bin:

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \left| acc(B_m) - conf(B_m) \right|,$$

where the dataset is divided into M interval bins based on confidence, and B_m is the set of indices of samples of which the confidence scores fall into each interval, while $acc(B_m)$ and $conf(B_m)$ are the average accuracy and confidence for each bin, respectively.

The ECE score is a measure of calibration error, and a lower ECE score indicates better calibration. If the ECE score is close to zero, it means that the model's predicted probabilities are well-calibrated, meaning they accurately reflect the true likelihood of the observations. Conversely, a higher ECE score suggests that the model's predictions are poorly calibrated. To summarise, the ECE score quantifies the difference between predicted probabilities and actual outcomes across different bins of predicted probabilities.

Data availability

Data used in this study are available in https://github.com/AIHubForScience/GPT_MLP.

Code availability

Source codes used in this study are available in https://github.com/AIHubForScience/GPT_MLP.

Received: 19 October 2023; Accepted: 15 January 2024;
Published online: 15 February 2024

References

1. Tshitoyan, V. et al. Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature* **571**, 95–98 (2019).
2. He, T. et al. Precursor recommendation for inorganic synthesis by machine learning materials similarity from scientific literature. *Sci. Adv.* **9**, eadg8180 (2023).
3. Choudhary, K. & Kelley, M. L. ChemNLP: A Natural Language-Processing-Based Library for Materials Chemistry Text Data. *J. Phys. Chem. C* **127**, 17545–17555 (2023).
4. Hatakeyama-Sato, K. & Oyaizu, K. Integrating multiple materials science projects in a single neural network. *Commun. Mater.* **1**, 49 (2020).
5. Choi, J., & Lee, B. Quantitative topic analysis of materials science literature using natural language processing. *ACS Appl Mater Interfaces* **16**, 1957–1968 (2024).
6. Olivetti, E. A. et al. Data-driven materials research enabled by natural language processing and information extraction. *Appl. Phys. Rev.* **7**, 041317 (2020).
7. Huo, H. et al. Semi-supervised machine-learning classification of materials synthesis procedures. *npj Comput. Mater.* **5**, 62 (2019).
8. Trewartha, A. et al. Quantifying the advantage of domain-specific pre-training on named entity recognition tasks in materials science. *Patterns* **3**, 100488 (2022).
9. Choi, J. et al. Deep Learning of Electrochemical CO_2 Conversion Literature Reveals Research Trends and Directions. *J. Mater. Chem. A* **11**, 17628–17643 (2023).
10. Pei, Z., Yin, J., Liaw, P. K. & Raabe, D. Toward the design of ultrahigh-entropy alloys via mining six million texts. *Nat. Commun.* **14**, 54 (2023).
11. Fujinuma, N., DeCost, B., Hattrick-Simpers, J. & Lofland, S. E. Why big data and compute are not necessarily the path to big materials science. *Commun. Mater.* **3**, 59 (2022).
12. Wang, L. et al. A corpus of CO_2 electrocatalytic reduction process extracted from the scientific literature. *Sci. Data* **10**, 175 (2023).
13. Kononova, O. et al. Text-mined dataset of inorganic materials synthesis recipes. *Sci. Data* **6**, 203 (2019).
14. Brown, T. et al. Language models are few-shot learners. *Adv. Neural Inform. Process. Syst.* **33**, 1877–1901 (2020).
15. Walker, N. et al. Extracting structured seed-mediated gold nanorod growth procedures from scientific text with LLMs. *Digi. Discov.* **2**, 1768–1782 (2023).
16. Zheng, Z., Zhang, O., Borgs, C., Chayes, J. T. & Yaghi, O. M. ChatGPT Chemistry Assistant for Text Mining and the Prediction of MOF Synthesis. *J. Am. Chem. Soc.* **145**, 18048–18062 (2023).
17. Zheng, Z. et al. A GPT-4 Reticular Chemist for Guiding MOF Discovery. *Angewandte Chemie Int. Edit.* **62**, e202311983 (2023).
18. Kononova, O. et al. Opportunities and challenges of text mining in materials research. *Iscience* **24**, 102155 (2021).
19. Keith, J. A. et al. Combining machine learning and computational chemistry for predictive insights into chemical systems. *Chem. Rev.* **121**, 9816–9872 (2021).
20. Zhao, S. & Birbilis, N. Searching for chromate replacements using natural language processing and machine learning algorithms. *npj Mater. Degrad.* **7**, 2 (2023).
21. Kim, J., Jang, S., Park, E. & Choi, S. Text classification using capsules. *Neurocomputing* **376**, 214–221 (2020).
22. Huang, S. & Cole, J. M. BatteryBERT: A Pretrained Language Model for Battery Database Enhancement. *J. Chem. Inform. Model.* **62**, 6365–6377 (2022).
23. Cruse, K. et al. Text-mined dataset of gold nanoparticle synthesis procedures, morphologies, and size entities. *Sci. Data* **9**, 234 (2022).
24. Wang, Z. et al. Dataset of solution-based inorganic materials synthesis procedures extracted from the scientific literature. *Sci. Data* **9**, 231 (2022).
25. Huang, S. & Cole, J. M. A database of battery materials auto-generated using ChemDataExtractor. *Sci. Data* **7**, 260 (2020).
26. Huang, S. & Cole, J. M. BatteryDataExtractor: battery-aware text-mining software embedded with BERT models. *Chem. Sci.* **13**, 11487–11495 (2022).
27. Wilary, D. M. & Cole, J. M. ReactionDataExtractor 2.0: A deep learning approach for data extraction from chemical reaction schemes. *J. Chem. Inform. Model.* **63**, 6053–6067 (2023).
28. Swain, M. C. & Cole, J. M. ChemDataExtractor: a toolkit for automated extraction of chemical information from the scientific literature. *J. Chem. Inform. Model.* **56**, 1894–1904 (2016).
29. Manica, M. et al. An information extraction and knowledge graph platform for accelerating biochemical discoveries. *arXiv preprint arXiv:1907.08400* (2019).
30. Gupta, T., Zaki, M., Krishnan, N. A. & Mausam. MatSciBERT: A materials domain language model for text mining and information extraction. *npj Comput. Mater.* **8**, 102 (2022).
31. Shetty, P. & Ramprasad, R. Automated knowledge extraction from polymer literature using natural language processing. *Iscience* **24**, 101922 (2021).
32. Gao, Y., Wang, L., Chen, X., Du, Y. & Wang, B. Revisiting Electrocatalyst Design by a Knowledge Graph of Cu-Based Catalysts for CO2 Reduction. *ACS Catal.* **13**, 8525–8534 (2023).
33. Nie, Z. et al. Automating materials exploration with a semantic knowledge graph for Li-ion battery cathodes. *Adv. Funct. Mater.* **32**, 2201437 (2022).
34. Li, J., Sun, A., Han, J. & Li, C. A survey on deep learning for named entity recognition. *IEEE Trans. Knowledge Data Engineer.* **34**, 50–70 (2020).
35. Yadav, V. & Bethard, S. A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics* pp. 2145–2158 (2018).

36. Beltagy, I., Lo, K. & Cohan, A. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* 3615–3620 (2019).
37. 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).
38. Weston, L. et al. Named entity recognition and normalization applied to large-scale information extraction from the materials science literature. *J. Chem. Inform. Model* **59**, 3692–3702 (2019).
39. Shetty, P. & Ramprasad, R. Machine-guided polymer knowledge extraction using natural language processing: The example of named entity normalization. *J. Chem. Inform. Model.* **61**, 5377–5385 (2021).
40. Lewis, P., Oguz, B., Rinott, R., Riedel, S. & Schwenk, H. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 7315–7330.
41. Zhang, Z. & Saligrama, V. In *Proceedings of the IEEE international conference on computer vision*. 4166–4174.
42. Yin, W., Hay, J. & Roth, D. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 3914–3923.
43. Guo, C., Pleiss, G., Sun, Y. & Weinberger, K. Q. In *International conference on machine learning*. 1321–1330 (PMLR).
44. Hendrycks, D., Mu, N., Cubuk, E. D., Zoph, B., Gilmer, J., & Lakshminarayanan, B. Augmix: A simple data processing method to improve robustness and uncertainty. In *International Conference on Learning Representations* (2019).
45. Desai, S. & Durrett, G. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 295–302.
46. Wang, S. et al. Gpt-ner: Named entity recognition via large language models. *arXiv preprint arXiv:2304.10428* (2023).
47. Yang, Y. & Katiyar, A. Simple and effective few-shot named entity recognition with structured nearest neighbor learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* 6365–6375 (2020).
48. Xu, K. Navigating the minefield of battery literature. *Commun. Mater.* **3**, 31 (2022).
49. Duan, S. et al. Three-dimensional reconstruction and computational analysis of a structural battery composite electrolyte. *Commun. Mater.* **4**, 49 (2023).
50. Dai, F. & Cai, M. Best practices in lithium battery cell preparation and evaluation. *Commun.s Mater.* **3**, 64 (2022).
51. Xie, T. et al. Large Language Models as Master Key: Unlocking the Secrets of Materials Science with GPT. *arXiv preprint arXiv:2304.02213* (2023).
52. Polak, M. P. & Morgan, D. Extracting Accurate Materials Data from Research Papers with Conversational Language Models and Prompt Engineering--Example of ChatGPT. *arXiv preprint arXiv:2303.05352* (2023).
53. Polak, M. P. et al. Flexible, Model-Agnostic Method for Materials Data Extraction from Text Using General Purpose Language Models. *arXiv preprint arXiv:2302.04914* (2023).
54. Li, B. et al. Evaluating ChatGPT's Information Extraction Capabilities: An Assessment of Performance, Explainability, Calibration, and Faithfulness. *arXiv preprint arXiv:2304.11633* (2023).
55. Chen, L., Zaharia, M. & Zou, J. Analyzing ChatGPT's Behavior Shifts Over Time. In *R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models* (2023).
56. Kumar, S. A survey of deep learning methods for relation extraction. *arXiv preprint arXiv:1705.03645* (2017).
57. Tsai, R. T.-H. et al. In *BMC bioinformatics*. 1–14 (BioMed Central).
58. Tsai, R. T.-H. et al. Various criteria in the evaluation of biomedical named entity recognition. *BMC Bioinform.* **7**, 1–8 (2006).

Acknowledgements

This work was supported by the National Research Foundation of Korea funded by the Ministry of Science and ICT (NRF-2021M3A7C2089739) and Institutional Projects at the Korea Institute of Science and Technology (2E31742 and 2E32533).

Author contributions

Jaewoong Choi: Conceptualisation, Methodology, Programming, Data analysis, Visualisation, Interpretation, Writing – original draft, Writing – review & editing. Byungju Lee: Conceptualisation, Interpretation, Writing – review & editing, Supervision, Resources, Funding acquisition.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s43246-024-00449-9.

Correspondence and requests for materials should be addressed to Byungju Lee.

Peer review information *Communications Materials* thanks Shu Huang and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Primary Handling Editors: Milica Todorović and Aldo Isidori.