Material Property Prediction with Joint Reasoning based on Large Language Models and Knowledge Graphs for Lithium Batteries

Gang Lei[1], Ziyu Yin[1], Jianmao Xiao[1]*, Xinji Qiu[2], Yuanlong Cao[1], Qian Zhang[3], Musheng Wu[2]

[1]Department of School of Software, Jiangxi Normal University, Nanchang, China

[2]Department of Physics, Jiangxi Normal University, Nanchang, China

[3]Department of Innovation Laboratory, Contemporary Amperex Technology Co.,Limited, Fujian, China

Research Article

Received: 11-Apr-2024, Manuscript No. JOMS-24-132112; **Editor assigned:** 15-Apr-2024, PreQC No. JOMS-24-132112 (PQ); **Reviewed:** 29- Apr-2024, QC No. JOMS-24-132112; **Revised:** 06-May-2024, Manuscript No. JOMS-24-132112 (R); **Published:** 13-May-2024, DOI: 10.4172/2321-6212.12.1.004

***For Correspondence:**

Jianmao Xiao, Department of School of Software, Jiangxi Normal University, Nanchang, China

E-mail: m xiao@jxnu.edu.cn

Citation: Lei G, et al. Material Property Prediction with Joint Reasoning based on Large Language Models and Knowledge Graphs for Lithium Batteries. RRJ Mater Sci. 2024;12:004.

ABSTRACT

Lithium batteries, as a crucial part of modern energy storage, rely heavily on the properties of their materials, which affect energy density, cycle life, charge/dis-charge rates, and safety. Traditional experimental methods for predicting these properties are often costly and time-consuming. While data-driven machine learning approaches can predict performance by analysing the impact of various factors, inconsistencies in data measurement and reporting, along with a lack of semantic integration of material information, hinder research on lithium battery materials. This paper introduces a lithium battery material property prediction method based on Joint Reasoning with Large Language models and Knowledge Graphs (JRLKG). Knowledge relationships are extracted from extensive literature to construct a knowledge graph encompassing material structure, properties, and research methods. By assessing material similarity within the graph, related information is fed into the GPT-4 model. Prompt learning techniques guide GPT-4 to leverage explicit knowledge from the external knowledge graph and its implicit knowledge for joint reasoning predictions, providing results and rationales. Experiments show that JRLKG improves the accuracy and interpretability of material property predictions, offering new avenues and methods for research and shortening the Research and development cycle for lithium-ion battery materials.

Keywords: Lithium-ion battery; Knowledge graph; Large language model; Material property prediction; Joint reasoning

INTRODUCTION

With the continuous increase in energy demand, particularly in areas such as renewable energy and electric vehicles, the prediction of battery material properties has become increasingly important. Among these, lithium-ion batteries, as a high-performance, high-energy-density energy storage device, have their performance critically dependent on the characteristics of the positive and negative electrode materials. By accurately predicting the properties of battery materials, researchers can quickly identify materials with high energy density, long cycle life, and good safety, thereby promoting the advancement of lithium battery technology. This is of great significance for reducing fossil fuel use, lowering greenhouse gas emissions, and promoting the transition to clean energy. In the field of lithium battery material research, evaluating battery material performance through manual experiments usually requires a significant amount of human resources and time cost and cannot cover a wide search of large-scale materials. As a result, predicting material properties has become a more efficient alternative. With accurate property prediction, scientists can understand the characteristics of materials in advance, helping them select the most promising lithium battery materials and accelerate the research and development process of new materials. This efficient prediction method not only saves time and cost but also provides strong support for the development of lithium battery material science.

Currently, the theoretical calculation of lithium-ion battery materials mainly adopts the first-principles method based on Density Functional Theory (DFT), which can predict the structural properties of materials according to fundamental physical laws and provide a theoretical basis for explaining some experimental phenomena. For instance, some researchers studied the application of this principle in voltage calculation, structural stability, and new material prediction of lithium-ion battery materials [1,2]. Shi Xiao Hong, et al. calculated the formation energy of oxygen vacancy clusters in lithium battery cathode materials using DFT [3]. Zhu Yanrong, et al. reviewed the research progress of the first-principles calculation of lithium-ion battery cathode materials, illustrating its research value and prospects [4]. Although DFT has high applicability and predictive power, its calculation time and cost are still expensive, and its calculations rely on the researcher’s experience and intuition. In contrast, machine learning-based methods utilize a large amount of data for training. By identifying and learning patterns within the data, these methods can predict the properties of new materials in an extremely short time and without the need for substantial computational resources once the model training is completed. This approach does not require detailed physical knowledge and manual experience, significantly saving the prediction time and cost. Researchers have utilized machine learning techniques to predict the coefficient of thermal expansion and thermodynamic properties of lithium-ion battery materials, and have found that the majority of the prediction results are consistent with the actual values [5,6].

While machine learning-based prediction methods have alleviated experimental constraints to some extent, their results largely depend on the input data pattern and lack a deep understanding and semantic analysis capability. Moreover, machine learning models often lack the ability for logical reasoning, unable to make logical inferences about unknown situations based on the information they have learned. Recent applications of AI technologies such as knowledge graphs and Large Language Models (LLMs) offer possibilities for addressing this challenge. Knowledge graphs represent relationships between entities in a graphical form, providing a rich knowledge representation that can translate complex entity relationships into a computable format and facilitate structured reasoning. For instance, researchers like Nie, et al. have conducted studies on the construction of knowledge graphs for lithium-ion battery materials in recent years [7,8]. They mined lithium-ion battery material literature and proposed a semantic representation framework, DATWEM, to embed terms related to lithium battery materials.

Using vector similarity, they built a semantic knowledge graph for lithium-ion battery cathode materials, revealing potential material relationships from the scientific literature. However, graphs based solely on vector similarity may not effectively reflect the connections between different materials' structures, properties, and synthesis methods, and lack rich entity category associations. Additionally, as a powerful natural language processing tool, LLMs provide strong support for understanding and processing vast text data and mining knowledge in the field of lithium battery materials. For example, researchers like Tong Xie, et al. fine-tuned Generative Pre-Trained Transformer 3 (GPT-3) to perform structured information reasoning for perovskite solar cells, including tasks like entity recognition and relationship extraction [9]. They could predict continuous property values of perovskite solar cells produced by specific synthesis methods. However, this fundamentally relies on regression prediction using LLMs based on different feature data and relying solely on the LLM's generative capabilities can lead to hallucination issues. Combining the factual associations in knowledge graphs with LLMs can effectively reduce model hallucinations. Existing knowledge graphs can provide LLMs with rich background information, ensuring that model predictions are not purely dependent on data patterns and contributing to more reliable inference predictions. At the same time, LLMs can add new entities and relationships to knowledge graphs, achieving completion and enhancement of the graphs. Such interaction and integration hold the promise of new breakthroughs in the field of lithium battery material prediction, bringing new opportunities and possibilities for finding and designing high-performance lithium battery materials.

This paper proposes an innovative method for predicting the properties of lithium battery materials by constructing a knowledge graph for lithium-ion battery materials that includes a wealth of entity relationship categories to serve as an external domain knowledge base. With the aid of the GPT-4 model, which possesses enhanced natural language understanding and learning capabilities, this method utilizes prompt engineering techniques to guide the model in combining explicit knowledge from the external knowledge graph with its own implicit knowledge for joint reasoning, thereby achieving predictions of material properties. The main contributions of this work are as follows:

(1) Constructed a literature knowledge graph in the field of lithium-ion battery materials, containing information on the structures, properties, research methods, and applications of various materials, as well as their interrelationships, providing rich data support for subsequent predictions.

(2) Achieved effective named entity recognition in the field of battery materials using LLM through fine-tuning the Llama2-7B model with prompts.

(3) Utilized weighted Jaccard similarity to calculate material similarity based on entity neighbor information in the knowledge graph, identifying potential similar materials.

(4) Integrated information on similar materials from the knowledge graph with the implicit knowledge in GPT-4 using prompt engineering techniques, enabling joint reasoning and returning results and reasoning explanations.

Related work

AI-based material property prediction: The application of Artificial Intelligence (AI) in the prediction of battery material properties has achieved some important breakthroughs. Some research teams have used machine learning algorithms to predict key properties of battery materials, training them with a large amount of theoretical and experimental data to achieve high predictive accuracy. Kauwe, et al. described how to use machine learning tools to predict the characteristics of battery materials, and by utilizing a dataset of cathode materials along with various statistical models, they predicted the specific discharge capacity of batteries after 25 cycles [10]. Takagishi, et al. designed the mesoscale porous structure of lithium-ion battery electrodes using three-dimensional virtual structures and machine learning techniques, establishing a regression model with artificial neural networks to predict the charge and discharge specific resistance of porous structure features [11]. Joshi, et al. used machine learning algorithms such as deep neural networks, support vector machines, and kernel ridge regression to predict the voltage of metal-ion battery electrode materials [12]. Liu, et al. proposed an interpretable machine learning framework that can effectively predict battery product characteristics and explain the interactions of dynamic effects and manufacturing parameters [13]. Although machine learning is sensitive to the intrinsic patterns and regularities of data, it lacks a semantic understanding of the underlying principles and has other shortcomings. Addressing the inadequacies of machine learning in considering material space information, Louis, et al. proposed two attention-based graph convolutional neural network techniques to learn the average voltage of electrodes [14]. This method, which combines the atomic composition and atomic coordinates in 3D space, significantly improved the accuracy of voltage predictions. Yang, et al. proposed a deep learning method to accurately predict the microstructure of electrode materials, addressing issues that machine learning cannot model, such as the microscopic structure of particles and output time-related problems [15]. Although deep learning can compensate for some of the shortcomings of machine learning, its black-box nature still makes it challenging to understand its internal mechanisms and the reasons behind its predictions.

Knowledge graph reasoning: Knowledge graphs can integrate scattered knowledge about battery materials into structured graphs, preserving semantic information between components. Their ability to manage and connect can reveal hidden patterns and knowledge associations in the data. In the field of knowledge graphs, reasoning is a crucial task aimed at addressing the incompleteness of knowledge graphs. Past research has mainly focused on logic rule-based reasoning methods, which typically rely on manually defined rules, limiting the flexibility and scalability of reasoning. Spectral graph embedding methods, on the other hand, map entities and relationships in knowledge graphs into low-dimensional vector spaces, allowing for the capture of a variety of complex relationships between entities by designing different embedding models. The earliest embedding-based methods include Trans E and Trans R and many variants and improved algorithms have been derived subsequently [16,17]. For example, Ren, et al. proposed the embedding-based knowledge graph reasoning framework query2box, which can handle arbitrary positive first-order logic queries in a scalable manner [18]. However, embedding methods often focus only on the direct connections between nodes, overlooking the more complex indirect relationships between them. Graph Neural Network (GNN) based methods can effectively capture high-order neighborhood information of nodes, thereby discovering complex relationships between entities. Teru, et al. proposed the GNN-based inductive knowledge graph reasoning framework Grail, which reasons about local subgraph structures and can predict relationships between nodes not seen during training [19]. Zhang M, et al. proposed a link prediction framework SEAL, which learns from locally closed subgraphs, embeddings, and attributes based on GNN [20]. Despite researchers proposing various knowledge graph-based reasoning methods, when confronted with some complex reasoning tasks, relying solely on graph-based reasoning methods often fails to achieve satisfactory results.

LLM combined with knowledge graph: Models such as Bert, GPT-3 have deep language understanding abilities through pre-training on massive text data [21,22]. Recently, openAI released ChatGPT and GPT-4, which can achieve good performance in various tasks with zero-shot or few-shot learning with just prompts. Many scholars have started to pay attention to the combination of LLM and knowledge graphs, such as using knowledge graph to enhance LLM, reduce hallucinations, or conduct reasoning. Yang L, et al. classified the existing methods of enhancing pretrained language models with knowledge graphs into before-training enhancement, during-training enhancement, and post-training enhancement, and explained their applications in entity recognition, relation extraction, knowledge graph completion, and question answering [23]. Wang S, et al. transformed the sequence labeling task of entity recognition into a generation task that can be adapted by LLM, achieving performance comparable to fully supervised baselines [24]. In reasoning tasks, Feng C, et al. proposed a knowledge resolver to prompt LLM to retrieve knowledge from an external knowledge graph, improving the interpretability of the reasoning process [25]. Wen Y et al. used prompt engineering techniques to enable LLM to understand input graph structures, guide it to combine graph information for reasoning, and generate mind maps for answers [26]. This approach of combining explicit knowledge from knowledge graphs with the implicit knowledge of LLM can overcome the limitations of traditional reasoning methods, combining LLM's contextual understanding abilities with the structured information of knowledge graphs, reducing LLM hallucinations, and achieving more precise and efficient reasoning. Applying this method to the prediction of lithium battery material properties can fully integrate the semantic knowledge of materials to achieve better reasoning and prediction results

MATERIALS AND METHODS

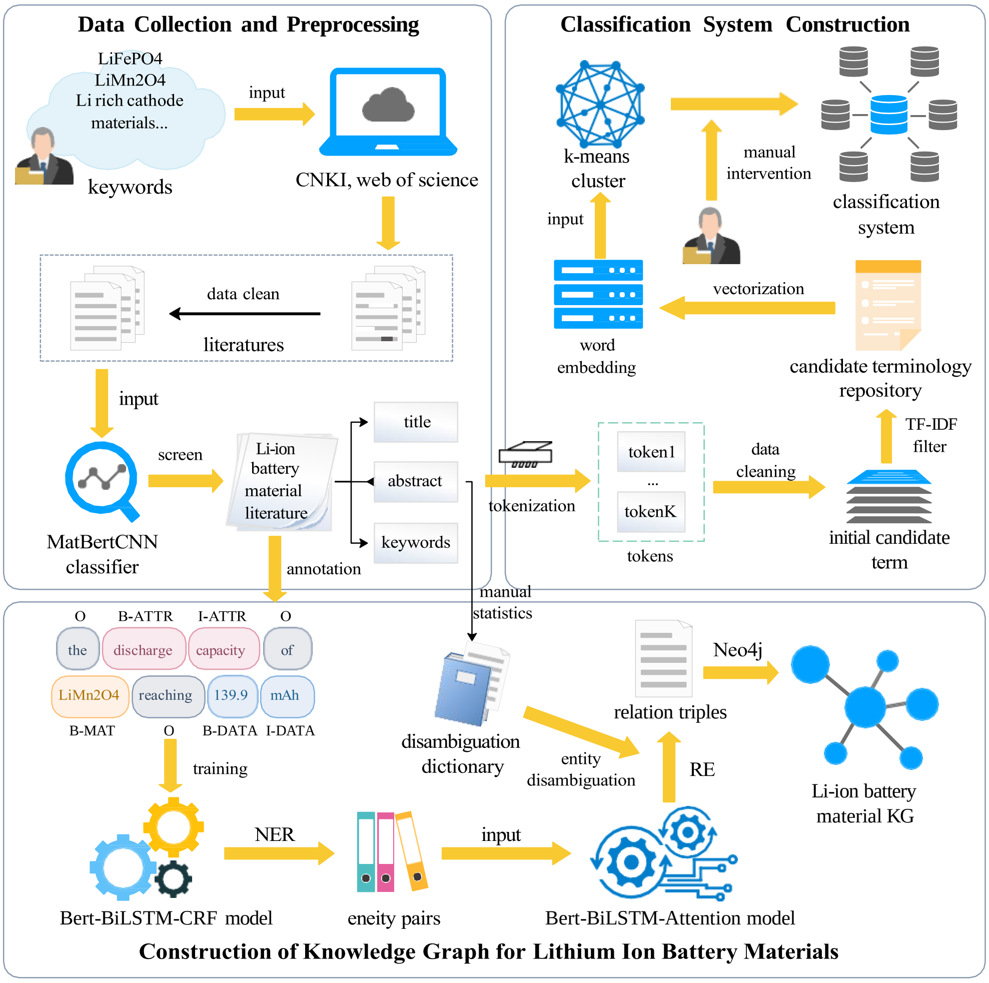
Construction of material knowledge graph

The framework for constructing a knowledge graph of lithium-ion battery materials: The construction process includes three parts: Data collection and preprocessing, classification system construction, and knowledge graph construction. The data collection and preprocessing process integrates and cleans the material literature data needed to construct the graph to ensure the high quality of the data. To better determine the research entity categories involved in battery material literature, we conducted clustering operations on candidate terms, combined with manual analysis and evaluation by Contemporary Amperex Technology Co., Limited (CATL) professional battery material research and development engineers, and determined the entity and relationship classification system of the lithium-ion battery material knowledge graph. Then, through named entity recognition and relationship extraction, the unstructured knowledge information in the material literature is constructed into a structured material knowledge graph.

Data collection and preprocessing: This article uses CNKI and Web of Science as the main sources for literature retrieval. We discussed with CATL professional battery material research and development engineers to determine the search keywords and rules for lithium-ion battery cathode material literature, and then exported the titles, abstracts, and keywords of the literature in bulk. The collected literature was deduplicated based on the title, incomplete data was removed, and the content was cleaned based on regular expressions. To further select the most relevant lithium-ion battery material literature, we used the BertCNN binary classification model to classify and screen the literature, removing articles outside the research field.

Classification system construction: Entities and relationships are the basic components of a knowledge graph. Constructing an entity relationship classification system can help us effectively organize and represent knowledge, making knowledge entries no longer isolated, but interconnected through relationships, forming a rich knowledge network. In this paper, we used a method of clustering candidate terms and discussed with CATL professors to comprehensively determine the knowledge classification system. First, we utilized the NLTK tokenization tool to tokenize the content of the article abstracts, keywords, and titles, and converted the words into lowercase, retaining verbs and nouns, performing lemmatization, and then removing stop words. The processed tokens were included in the initial candidate term set, and representative candidate terms were selected by calculating their TF-IDF values. Word2Vec's CBOW model was used to vectorize the tokenized results, converting them into 2D feature vectors, and then the K-means algorithm was used for cluster analysis of the candidate terms is shown in Figure 1.

Figure 1. Framework for the construction of the lithium-ion battery materials knowledge graph.



To achieve the best clustering effect, we evaluated the clustering results under different k values and ultimately chose a k value of 5 for clustering. Through manual analysis of the clustering results, we determined seven categories of entities:” Materials”,” Description”,” Method”,” Attribution”,” Application”,” Condition”, and” Data”. The definitions of each entity category are shown in Table 1.

Table 1. Descriptions of the seven entity categories.

Entity	Definition	Example
Materials (MAT)	Chemical formula and materials related to the study substance	Lithium, LiMn2O4, NMC
Description (DSC)	Characteristics and the internal structure of the material	Spinel, Layer, Mg-dopped
Method (MET)	The synthesis and characterization methods of the materials	Carbonization, DFT
Attribution (ATTR)	Evaluation indicators and attributes with measurable values	Good capacity, Energy density
Application (APL)	Application scenarios of the materials	Cathode, Solution, Catalytic
Condition (CON)	The prevailing conditions of the experiment	1 C, 300 cycles, 0.1 A g(-1)
Data (DATA)	Values of the measured experimental results	167.2 mAh g(-1), 80%, 3 nm

In consideration of the complexity inherent in subsequent relation extraction, this study focuses exclusively on the relationships between entities of different categories, temporarily disregarding relationships between entities of the same category. Through manual analysis of the potential associative relationships between entities and through consultation and discussion with experts in the field of battery materials, this research ultimately identifies six types of relationships for investigation: Materials-description, materials-application, materials-method, materials-attribution, materials-data, and data-condition.

RESULTS AND DISCUSSION

Named entity recognition

An entity refers to a named or referential object within the text that carries a specific meaning, which may be a tangible person, place, time, or an abstract concept, product, etc. Entities are a critical component of constructing knowledge graphs and encompass key knowledge points within a given domain; they can be composed of single or multiple words. By recognizing entities, key information can be extracted from the text, providing rich semantics for the knowledge graph and subsequent reasoning. Named entity recognition involves identifying the content from the start to the end position of an entity within domain-specific texts and classifying the entity into one of several pre-defined types. This process encompasses two tasks: Confirming the boundaries of an entity and identifying its category. Therefore, entity recognition can be transformed into a sequence labeling problem, predicting labels for each word based on features. In this study, the Bio tagging scheme was used to manually annotate the material literature abstracts. After a series of comparative experiments, we ultimately selected a NER model based on Bert-BiLSTM-CRF. Bert is utilized to convert the input word sequence into vectors, offering superior contextual understanding capabilities compared to general models. The BiLSTM layer is responsible for receiving incoming word vectors and learning textual features within the context. The Conditional Random Fields (CRF) layer models the conditional probability distribution of the label sequence, considering the dependencies between labels to ensure the consistency and rationality of the annotations.

Relation extraction

Relation extraction further identifies the interactive relationships between entities, transforming unstructured text data into structured information. This structured information is easy to store, search, and analyze, which is crucial for deeply understanding the text, extracting valuable knowledge, and further knowledge reasoning. This method matches different types of entities within the same sentence according to six predefined relationship categories. Manual screening is conducted to retain the correct relationships, while others are labeled as the ’other’ category. All of the data are used to train the relation extraction model. This paper utilizes a relation extraction model based on Bert-BiLSTM-Attention, which introduces an attention mechanism to selectively focus on and weight information at different positions in sequence data. This helps the model concentrate on the parts of the sentence that are most meaningful for relationship judgment and inputs their weighted vectors into a softmax classifier to obtain the final prediction results.

Entity disambiguation

Analysis of domain-specific literature reveals that professional terminologies in the materials field are often replaced with abbreviations, leading to redundancy issues between the full names and acronyms of entities. Furthermore, entity naming conventions may vary across different documents, resulting in various expressions for entities with the same meaning. To address this, entity disambiguation work is required to unify entity standards. This paper performs semantic-level and character-level disambiguation on the entities in the relation triples. Semantic-level disambiguation aims to unify entities that have the same meaning but different expressions, utilizing a manually compiled entity dictionary to standardize entities with the same meaning. Character-level disambiguation is based on the edit distance between characters to eliminate the influence of special characters. Regular expressions are used to extract the full name and acronym pairs from abstracts, and a full name-acronym dictionary is constructed, which allows for the replacement of all professional term acronyms in the database with their full names. The resulting knowledge graph, which includes information on battery materials and their structures, properties, research methods, and applications, is stored in the Neo4j graph database for easy retrieval. The rich material knowledge in the graph provides a solid factual foundation for subsequent inference and prediction tasks.

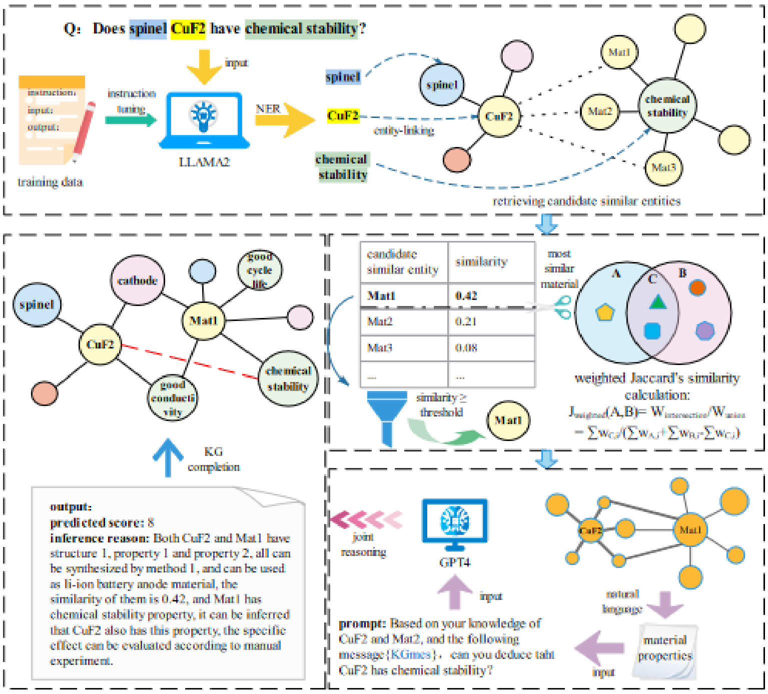
LLM combined with knowledge graph for joint reasoning: This paper employs prompt engineering techniques to guide the teacher-level LLM for reasoning. By transforming the structured knowledge from the knowledge graph into natural language and incorporating it into the prompts of the LLM, the model is directed to engage in joint reasoning based on existing facts and its own knowledge to predict material properties. The process of joint reasoning is illustrated in Figure 2 and consists of three main parts:

LLM-based named entity recognition and entity linking: Upon receiving a user query specifying the material and properties for prediction, we fine-tuned the open-source Llama2 model to automatically extract material entities from the query and link them to the entities in the knowledge graph.

Knowledge graph-based similar material retrieval: Based on the linked material entities and their predicted properties within the graph, potential candidate similar materials are identified by analyzing their positional relationships. Weighted Jaccard similarity is computed to filter out the most probable similar materials.

Guiding GPT-4 for materials property prediction with prompt engineering: Transform the association of common attributes in the graph between the retrieved similar materials and the target material into a form of natural language, and incorporate it into the prompts for GPT-4, guiding it to perform joint reasoning based on internal and external knowledge, to predict whether the material possesses the target attribution and provide the reasons.

Figure 2. Joint reasoning process of LLM combined with material knowledge graph.



Q: Does spinel CuF2 have chemical stability?

candidate similar entity	similarity
Mat1	**0.42**
Mat2	0.21
Mat3	0.08
...	...

weighted Jaccard's similarity calculation:
$J_{weighted}(A,B)= W_{intersection}/W_{union}$
$= \sum w_{C,i}/(\sum w_{A,i}+\sum w_{B,i}-\sum w_{C,i})$

output:
predicted score: 8
inference reason: Both CuF2 and Mat1 have structure 1, property 1 and property 2, all can be synthesized by method 1, and can be used as li-ion battery anode material, the similarity of them is 0.42, and Mat1 has chemical stability property, it can be inferred that CuF2 also has this property, the specific effect can be evaluated according to manual experiment.

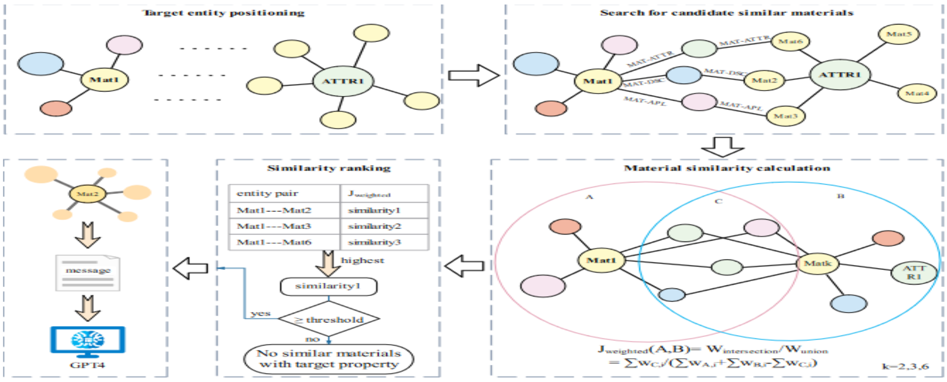
prompt: Based on your knowledge of CuF2 and Mat2, and the following message{KGmes}， can you deduce taht CuF2 has chemical stability?

LLM-based named entity recognition and entity linking: This article uses LLM's natural language question-answering ability to predict the properties of specific materials. Firstly, it identifies and extracts the entities from the questions posed by users to determine the material and attribution required for reasoning. The method of instruction fine-tuning is employed to enable LLM to automatically identify material entities from natural language sentences. Compared to traditional sequence labeling NER methods, instruction fine-tuning LLM is an end-to-end training approach, where the model can automatically learn features of input-output text by inputting a certain amount of task requirements and answer example data, thus adapting to tasks in different domains. Additionally, LLM's strong contextual understanding and good generalization ability can effectively solve the problem of discontinuous entities in sequence labeling tasks, without requiring large-scale annotated data, and still perform well in situations with small amounts of data.

Among many open-source LLMs, LLAMA has achieved excellent predictive results by pre-training on different domain datasets, with its performance in some benchmark tests even surpassing the current teacher-leveled GPT-3 model. Many researchers have further fine-tuned and expanded various professional LLM based on this model, such as TransGPT, MetaMath, LawGPT, and others [27-30]. Recently, Meta has released LLAMA2, which is free for commercial use and has shown further performance improvement compared to its predecessor [31]. This paper fine-tuned the LLAMA2-7B model to achieve named entity recognition in the field of lithium battery materials. Following the fine-tuning documentation of Stanford's Alpaca the designed instructions consist of three parts: Instruction, input, and output [32]. The "instruction" specifies the task for the model to complete, which is to perform named entity recognition on the input sentences; "input" refers to the sentences for entity recognition; "output" represents the model's output, which is the set of entities extracted from the sentences. Entities are not classified in this part. As the extracted entities may not all exist in the knowledge graph, it is necessary to link them with the entities in the knowledge graph. Specifically, this paper disambiguates the entities in the set and then sequentially places them into the knowledge graph for retrieval, retaining those stored entities in the knowledge graph, and performing reasoning based on the information association between them.

Knowledge graph-based similar material retrieval: Upon locating the relevant entities within the knowledge graph, the initial step involves categorizing each entity and identifying the target material along with the attribution to be predicted. Subsequently, assess whether a direct attribute association exists between them in the knowledge graph; if such a connection is present, it indicates that the material possesses the target attribution. In the absence of a direct link, the next step is to search for materials potentially similar to the target material, with the search methodology depicted in Figure 3.

Figure 3. Similar material retrieval process.


Target entity positioning
Mat1
ATTR1
Search for candidate similar materials
Mat1
MAT-ATTR
MAT-ATTR
MAT-DSC
MAT-DSC
MAT-APL
MAT-APL
Mat6
Mat5
Mat2
ATTR1
Mat4
Mat3
Material similarity calculation
A
C
B
Mat1
Matk
ATT R1
$J_{weighted}(A,B)= W_{intersection}/W_{union}$
$= \sum w_{C,i}/(\sum w_{A,i}+\sum w_{B,i}-\sum w_{C,i})$
k=2,3,6
Similarity ranking
highest
similarity1
yes
≥ threshold
no
No similar materials with target property
Mat2
message
GPT4

entity pair	$J_{weighted}$
Mat1---Mat2	similarity1
Mat1---Mat3	similarity2
Mat1---Mat6	similarity3

Based on the relationships between nodes in the material knowledge graph, it is known that two material entities are often linked through common one-hop neighbor nodes of other categories, which means that two materials may share similar characteristics, such as descriptions, attributions, research methods, and applications. This study identifies potential similar materials and predicts attributions based on these common features. Initially, all candidate similar materials that possess the target attribution and share common features with the target material are retrieved.

Subsequently, the similarity between the two materials is calculated based on their respective neighboring nodes and associations. This method considers each material entity as the center and views all its neighbor nodes as a set. The common nodes between the two materials are considered as the intersection of the two sets, and the weighted Jaccard similarity is used to measure their similarity. The Jaccard similarity is a commonly used metric for measuring the similarity between two sets, calculated by the ratio of the intersection to the union of the two sets. Considering the different impacts of various feature categories on the attributions of materials, different weights are assigned to different node categories in the calculation of the weighted Jaccard similarity. Specifically, the weight for “description” and “application” nodes is 0.3, while the weight for “attribution” and “method” nodes is 0.2. Other node categories are not included in the similarity calculation. The formula for the calculation of the weighted Jaccard similarity is as follows:

$$J_{weighted}\left(A,B\right)=\frac{W_{A\cap B}}{W_{A\cup B}}=\frac{W_{C}}{W_{A\cup B}}=\frac{\sum_{i}^{k} w_{C,i}}{\sum_{i}^{n} w_{B,\mathrm{i}}-\sum_{i}^{k} w_{C,\mathrm{i}}} \quad \text{.... (1)}$$

A and B represent the sets of neighboring nodes corresponding to the two materials, C is the set of their common nodes, with m and n being the number of neighboring nodes for each material, and k is the number of common nodes. W stands for the total weight of the corresponding set, with w representing the weight of the ith node within that set. After ranking all candidate similar materials by their similarity scores, the material with the highest score is outputted. This material is then subject to threshold filtering. The impact of different threshold values on the experimental results varies, and the optimal threshold is selected through experimentation. The information of similar material that meets the criteria is used for subsequent attribution prediction to ensure the reasoning is logical and credible.

Guiding GPT-4 for materials property prediction with prompt engineering: In this step, the similar material information previously retrieved is utilized to prompt the GPT-4 model to generate the final inference prediction results. First, based on the similar material entities identified in the previous step, information retrieval is conducted within the knowledge graph, returning common feature information between it and the target material. The common features between the two materials are then translated into a natural language format that is readily comprehensible to the LLM and incorporated into the prompt. As shown in Figure 4, the designed prompt information contains four parts: System directives, task descriptions, graph facts, and output requirements. The system directive guides the model to answer questions in the identity of a specific battery material field, adding professionalism to the output; the task description informs the model of the current material property prediction task and guides the model to use different knowledge for reasoning; it also enables the model to understand and combine external knowledge graph fact information; finally, the model is instructed on the result format to output, requiring it to provide a predicted score and reasoning behind the prediction.

Figure 4. The prompt template for final input to LLM.

```
def llm reasoning(kg mes: str, main mat: str, sim mat: str, pre attr: str):
conversation = [
    "role": "system",
"Content": "I am an AI-powered assistant specialized in knowledge and question
answering related to battery materials.",
    "role": "user",
    "content": "Based on your knowledge of " + main mat + ", Please provide a specific
score
(integer from 0-10) on how likely material " + main mat + " is to have properties " +
pre_attr + " and explain the reasoning. You can analyze the common characteristics of
similar materials " + sim mat + " by referring to their shared features. The common
characteristics between the two are as follows: " + kg mes + "format: Predicted score:
reason: "]
    response = openai.ChatCompletion.create(
    model="gpt-4-1106-preview",
    messages=conversation,
    max tokens=1000)
  message = response['choices'][0]['message']['content']
  return message
```

The research finds that in the absence of a specialized LLM specifically tailored for battery materials, direct questioning of a general-domain LLM about battery materials leads to issues such as lack of domain knowledge, inaccurate responses, and hallucinations, even with extensive knowledge learning. Relying solely on structured knowledge graphs for reasoning presents challenges related to incomplete graphs, missing entity relationships and attributes, and limited capabilities in complex reasoning tasks, lacking the contextual understanding abilities of LLM. Integrating knowledge from both the knowledge graph and the LLM's internal knowledge allows for comprehensive and accurate reasoning by leveraging the structured information within the knowledge graph and integrating the textual knowledge of LLM. The knowledge graph provides rich domain knowledge to LLM, offering essential support for tasks in different domains and enhancing the model's generalization capabilities. By combining the knowledge from the knowledge graph and LLM, the interpretability of material attribution prediction results is improved, aiding users in understanding the reasoning process and evaluating the validity of the results.

Materials knowledge graph construction

Named entity recognition: We randomly selected 500 literature abstracts and manually annotated them using the Yedda annotation tool, subsequently exporting the data in the BIO format [33]. BIO annotation allows for the labeling of continuous multi-word entities, with “B” denoting the beginning of an entity, “I” for intermediate or ending words, and “O” for text content not belonging to specified entity categories, accurately representing entity boundaries. The annotated data was divided into training, validation, and testing sets in an 8:1:1 ratio. Precision, recall, and F1-score were chosen as evaluation metrics for model performance. Precision measures the accuracy of the model in predicting positive samples, indicating the proportion of true positive samples among those predicted as positive by the model; recall measures the model's ability to identify positive samples, representing the proportion of correctly identified positive samples among all true positive samples; F1-score is the harmonic mean of precision and recall, providing a comprehensive measure of the performance of the classification algorithm, suitable for imbalanced datasets or tasks with high demands for both precision and recall. The formulas for calculating these metrics are as follows:

$$\text{Precision} = \text{TP}/(\text{TP}+\text{FP}) \quad \text{.................} \quad (2)$$

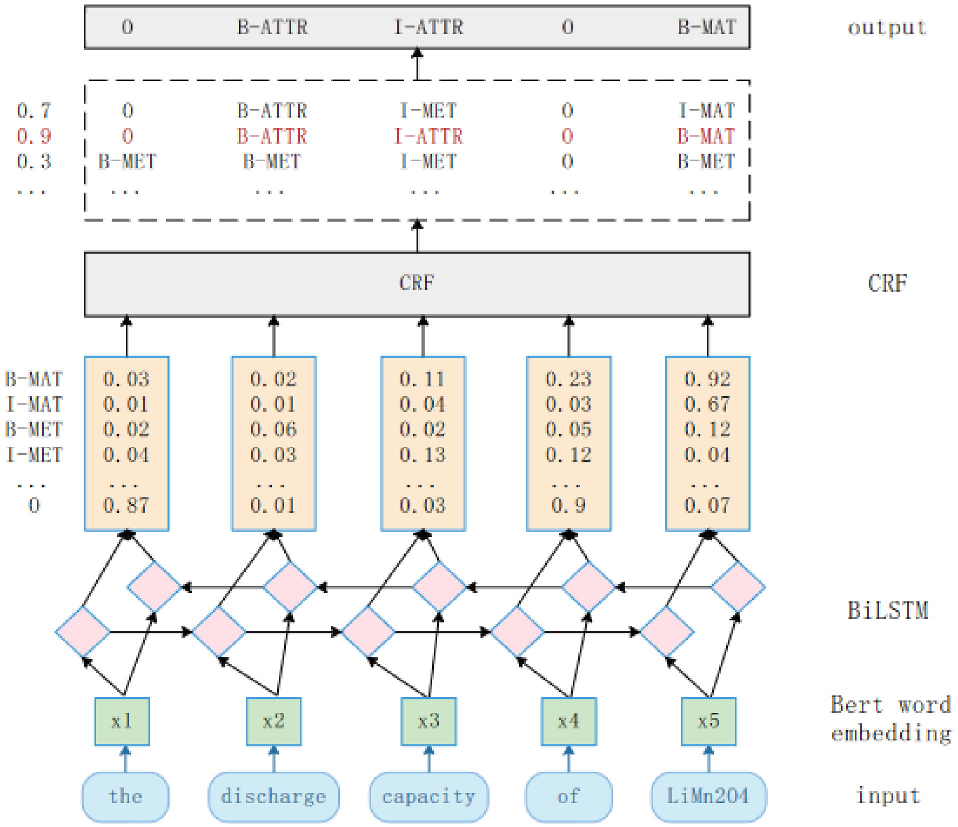
$$\text{Recall} = TP/(TP + FN) \quad \text{.................} (3)$$

$$F1 = 2\,\text{PrecisionRecall}/(\text{Precision} + \text{Recall}) \quad \text{.................} (4)$$

Where,

TP (True Positive), FP (False Positive) and FN (False Negative) are metrics used in the confusion matrix to measure the performance of a classification model. TP (True Positive) represents the number of instances where the model correctly predicts a positive class as positive, FP (False Positive) represents the instances where the model incorrectly predicts a negative class as positive, and FN (False Negative) represents the instances where the model incorrectly predicts a positive class as negative. The text describes the architecture and function of the Bert-BiLSTM-CRF model used for named entity recognition. This model combines several deep learning components and is commonly employed in natural language processing tasks. The model consists of five main parts: the input layer, word embedding layer, BiLSTM layer, CRF layer, and model output, as illustrated in Figure 5. The input layer processes annotated BIO-corpus data. The word embedding layer transforms the text data into continuous vectors to capture semantic and syntactic relationships between words. Bert is a pre-trained language model based on Transformer, which learns universal.

Figure 5. Bert-BiLSTM-CRF model architecture.



Language representations through large-scale unsupervised training, providing more accurate and rich word embedding representations. The BiLSTM layer combines forward and backward state information to effectively capture word context features and outputs scores for each word vector across various labels, with the highest score corresponding to the best predicted label. The CRF layer learns the dependencies between labels to ensure consistency and rationality in the annotations, avoiding invalid entity tags such as $I-X\ I-X$ or $B-X\ I-Y$. Finally, the optimal label sequence is selected for output by computing the CRF loss. The CRF loss function involves two components: The score of the true path and the sum of the scores of all possible paths. During training, the objective is to maximize the score of the true path and minimize the scores of other paths, with the difference between the two being the CRF loss, as per the following formula:

$$L=-\left[\log\left(P\left(\frac{y}{x}\right)\right)\right]=\log(Z)-\log\left(P\left(\frac{y}{x}\right)\right) \quad \text{.................... (5)}$$

Where y is the true label sequence, x is the input sequence, Z is the sum of scores for all possible label sequences, and $\log\left(P\left(\frac{y}{x}\right)\right)$ is the score of the true label sequence. The relevant model parameters are detailed in Table 2

Table 2. Bert-BiLSTM-CRF model parameters.

Parameter	Definition	Values
Batch size	Batch size	10
Max len	The maximum length of the processed sequence data	50
Lstm hidden	Dimensions of the hidden state of the Lstm	768
Lr	Learning rate	1e-04
Epoch	Iterations	30
Weight Decay	Regularized coefficient of the weight decay	0.01

To demonstrate the effectiveness of the entity recognition model, we compared the performance of the models under different algorithms, and the accuracy, recall, and F1 values of each model are shown in Table 3. We found that the entity recognition effectiveness of the Support Vector Machine (SVM) machine learning model was the lowest, with an F1 value of only around 20%, while the deep learning model CNN showed improvement compared to the machine learning model. Using pre-trained models such as Bert for word embedding significantly improved the effectiveness of entity recognition. Additionally, to demonstrate the necessity of introducing LSTM and CRF layers, we compared the performance of the model before and after adding the CRF and LSTM layers, and replaced BiLSTM with the classic BiGRU model for comparison. It can be observed that the introduction of LSTM and CRF layers effectively improved the precision and recall of the model, and BiLSTM has better contextual feature extraction ability than BiGRU.

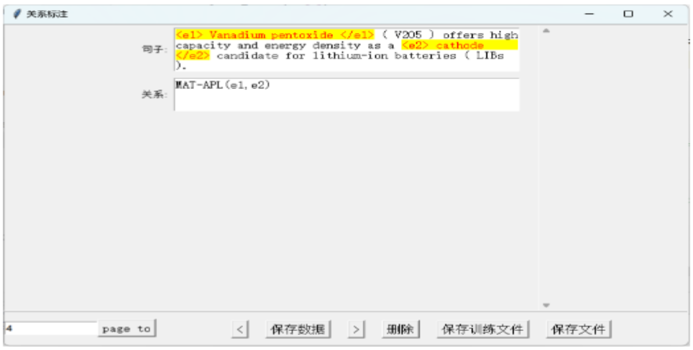
Table 3. Results of the different entity recognition model.

NER model	Precision	Recall	F1-score
SVM	0.79	0.31	0.19
CNN	0.78	0.68	0.55
Bert	0.82	0.76	0.79
Bert-CRF	0.82	0.77	0.8
Bert-BiGRU-CRF	0.8	0.31	0.81
Bert-BiLSTM-CRF	0.82	0.31	0.83

Relation extraction

Based on the annotation results of entity recognition, relationships between different entities within the same sentence were labeled. Specifically, this study developed a relationship annotation tool called gui-annotation. First, the bio results of entity recognition were segmented into sentences using blank lines as delimiters. Then, based on the defined relationship categories, the tool matched entities within each sentence and saved all matched entity relationships in JSON format, which was then passed to the annotation tool. The tool highlighted the entity pairs within the sentence on the interface, allowing annotators to see the relationship category between the entities, the position of each entity within the sentence, and the entity type. Annotators then manually judged the relationship, saving the data if the relationship was correct, or categorizing it as" other" if no relationship existed between the entities. To ensure data quality, relationships that annotators could not accurately assess could be selected for removal and excluded from the training data. The developed annotation tool had a GUI interface as shown in Figure 6. A total of 1700 manually annotated relationship data were used to train and evaluate the model.

Figure 6. GUI interface of the annotation tool.

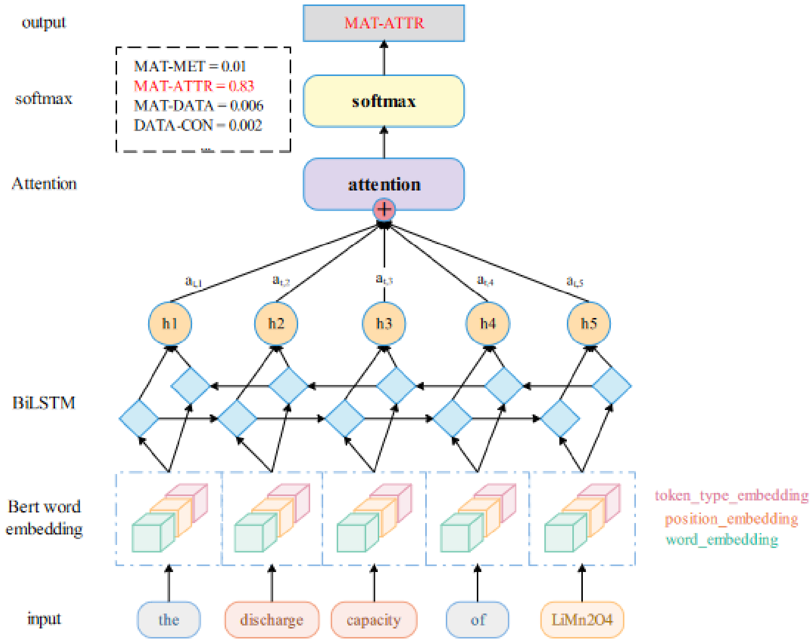


句子: <e1> Vanadium pentoxide </e1> (V205) offers high capacity and energy density as a <e2> cathode </e2> candidate for lithium-ion batteries (LIBs).

关系: MAT-APL(e1, e2)

This article constructs a material domain relation extraction model based on the Bert-BiLSTM-Attention method, and the model structure is shown in Figure 7.

Figure 7. Bert-BiLSTM-attention model architecture.



The model introduces the attention mechanism, which is used to selectively focus on and weigh information at different positions in the sequence data. By calculating the weight of each position, the model can more effectively concentrate on the important parts. In the relation extraction model, the attention mechanism can help the model focus on the most meaningful parts for relation discrimination in the sentence. The output of the Attention layer is the weighted sum of each position in the input sequence. For each position, the corresponding weight is obtained by calculating the similarity with other positions, and then the weighted sum of the representation vectors of each position is obtained by multiplying and summing the weights of each position with its corresponding position.

Finally, the weighted vector is input into the softmax classifier to obtain the final predicted classification result. The softmax function is a commonly used activation function that converts a set of real numbers into a probability distribution. In deep learning, the softmax function is usually used for multi-class classification problems, transforming the model's output into the probability value for each category, and then selecting the category with the highest probability for output. It takes the exponential value of each element of the input vector, then sums all the exponential values, and divides each element's exponential value by the sum to obtain the probability of each element. The formula is as follows:

$$soft\max\left(\vec{x}_i\right)=\frac{e^{xi}}{\sum_{j=1}^{k}e^{xj}} \quad \text{................. (6)}$$

The input $\vec{x}_i$ a vector containing K elements, where K is the number of categories in the relationship. The term x_i refers to the ith element in the input vector, and the softmax value represents the probability that the input vector belongs to the ith category.

This study employs the cross-entropy loss function to facilitate parameter learning and optimization for the relation extraction model. The cross-entropy loss function measures the discrepancy between the probability distribution output by the model and the actual labels. The loss function reaches its minimum value of 0 when the model output perfectly matches the true labels. Conversely, as the difference between them increases, so does the cross-entropy loss. This function aids the model in learning correct classification decisions and in optimizing parameters to minimize classification errors. The formula for its calculation is as follows:

$$CrossEntropy\left(y,p\right)=-\sum_{i=1}^{n}y\left(x_i\right)\log\left(p\left(x_i\right)\right) \quad \text{.................. (7)}$$

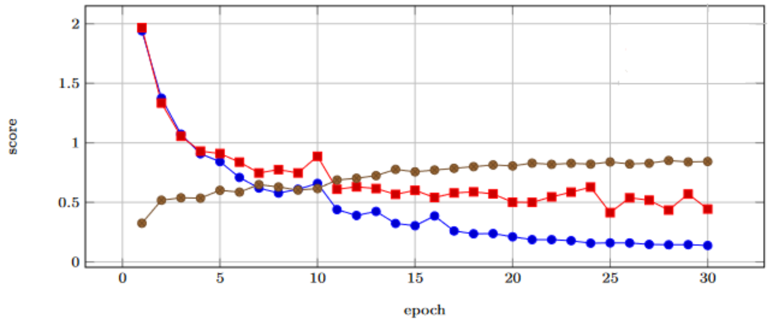
Where y represents the probability distribution of the true labels, and p represents the probability distribution output by the model. The terms y_i and p_i respectively denote the probability values of the ith category for the true labels and the model output. The settings for the model-related parameters are shown in Table 4.

Table 4. Bert-BiLSTM-attention model parameters.

Parameter	Definition	Values
Batch_size	batch size	10
max_len	the maximum length of the processed sequence data	80
stm_hidden	dimensions of the hidden state of the LSTM	768
lr	learning rate	1e-05
epoch	iterations	30
dropout	Probability of neuron dropout	0.1
weight decay	regularized coefficient of the weight decay	0.01

During the training process, the model that achieves a higher F1 score is continually saved by comparing changes in the loss function values for the training and validation sets, as well as changes in the F1 score on the validation set. During 30 epochs, the performance of the model is illustrated in Figure 8.

Figure 8. Loss and F1 score trends for relation extraction model. **Note:** —●— Train-loss; —■— Dev-loss; —●— Dev-F1



We observe that after ongoing training and tuning, the loss for both the training and validation sets has been steadily decreasing and tending towards convergence. Subsequently, the model’s F1 score on the validation set consistently exceeds 80%, ultimately converging around 84%. Finally, we evaluate the best-performing model on the test set, achieving a final loss of 0.435 and an F1 score of 85.15%. The experimental results indicate that our relation extraction model has achieved fairly good performance and is capable of effectively extracting relationships between entities.

Entity disambiguation

After extracting all the relational triples from the literature through relation extraction model, we perform disambiguation for entities with synonymous names and those with inconsistent character formatting. Initially, for abbreviations of specialized terms in the text, we construct a full name-abbreviation dictionary using regular expressions to replace all abbreviated entity terms in the triples. The pseudocode for building the full name-abbreviation dictionary is shown in Algorithm 1. Since most specialized terms in materials literature are represented in the format” full name (abbreviation)”, we extract full name-abbreviation pairs from all the literature by matching the initials of the full name, which starts with a capital letter, with the abbreviation found inside parentheses. Full names of different forms under the same abbreviation are separated by” |”. When performing disambiguation replacement, we replace the abbreviation with the first full name value under the corresponding key in the dictionary.

We manually collected and integrated similar entities with different forms to build a disambiguation dictionary. Then, we unified all entities in the triples into a single representation. We also manually supplemented and deleted redundant data, resulting in 186,473 retained relationships, which were imported into a Neo4j graph database.

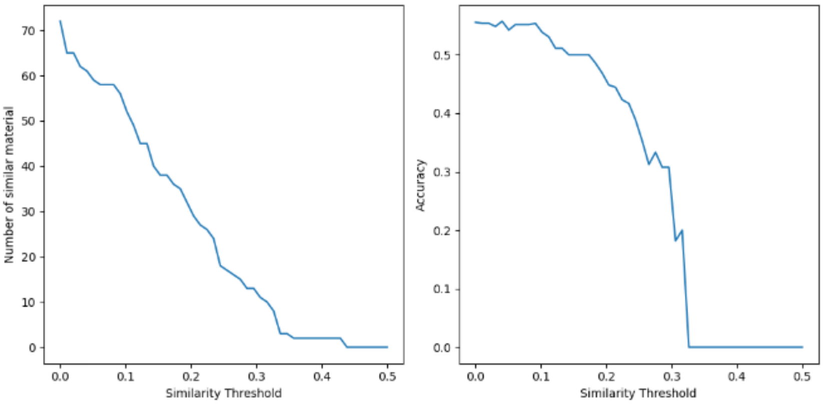
Algorithm 1: Build full name-abbreviation dictionary

```
Open the abstract file alltext.txt
Read the file content and store it in the variable named content
Create an empty dictionary dic = {}
for sentence in content do
    token word = word tokenize(sentence) ▷ tokenize the sentence
    Match the sentence using the regular expression r”([A − Z][A − Za − z0 −
    9 ()@]+)”, and store the results in the variable matches
    for item in matches do
        Count the number of uppercase letters in each item, referred to as count
        if item in token word then
            Find the index position p of the item in token word
            Search for the index k of the word in the preceding count token contents with the same initial letter as the item
            Concatenate the token word[k] to token word[p-1] into a string and store it in full
            Clear full, break
        end if
        if full not null then
            if item not in dic then
                dic [item] = full
            else
                Add full to dic[item], by '—'
            end if
        end if
        count = 0, clear full
    end for
end for
for each key-value pair in dictionary dic do
    Write the key and value in the format" key:value" into a new file
end for
```

Joint reasoning with llm and knowledge graphs

To assess the ability of our method to predict material properties, we manually collected some materials and their attribution information. We then constructed a set of 200 question-answer pairs based on predicting material attributions. Each question pertains to the association judgment between different materials and attributions, with the answer being either “yes” or “no”. To optimize predictive accuracy, we evaluated the impact of various parameters on our model, including the Jaccard threshold for filtering candidate similar materials and the scoring threshold to determine the correctness of predictions. Given the diversity and richness of entity relationships and feature attributes among different lithium battery materials in the knowledge graph, the number of common feature nodes between materials is generally limited, resulting in typically modest material similarity scores. Therefore, the selection of an appropriate similarity threshold is crucial for model performance. Overly low thresholds may cause the model to overemphasize weak similarities, leading to incorrect inferences, while excessively high thresholds might prevent the retrieval of relevant similar materials, thus constraining the model from inferring based solely on its limited inherent knowledge. We analyzed the number of similar materials retrieved and the predictive outcomes with thresholds ranging from 0% to 50%, as illustrated in Figure 9.

Figure 9. The impact of different similarity thresholds on the model.



With increasing thresholds, stricter criteria for similarity retrieval are applied, which correspondingly decreases the number of similar materials that meet these criteria, and the accuracy of material property prediction by similarity reasoning declines. It is observed that retrieving materials with a similarity exceeding 50% is challenging using the knowledge graph, highlighting the importance of selecting an optimal similarity threshold. The change in threshold and accuracy indicates a marked acceleration in the decline of accuracy around a similarity threshold of 0.2, where the quantity of similar materials falls to approximately half of the initial value. Consequently, a threshold of 0.2 is deemed suitable for the retrieval of similar materials.

The model will use common features between similar materials and combine them with its own knowledge to infer and predict whether the attribution of the target material can be predicted based on the similarity between the materials. For each material to be predicted, the model will provide a likelihood score (0-10) for the establishment of the property association. However, in reality, there are only two situations: association and non-association between the actual material and attribution. Therefore, it is necessary to map the model's prediction scores into two categories: Correct or incorrect. Traditional binary classification methods consider results greater than the mean (in this case, 5) as positive samples and those less than as negative samples. Considering the uniqueness of material data, we also analyzed and selected the threshold values for the scores. Setting different thresholds has the advantage of adjusting the sensitivity and specificity of the classification based on specific situations. By adjusting the threshold, the classifier can more flexibly adapt to different application scenarios and requirements. A higher threshold will increase specificity and reduce the false positive rate, while a lower threshold will increase sensitivity and reduce the false negative rate. Therefore, we compared the model's predictive performance at different thresholds between 4 and 8 to select the optimal threshold for the model to achieve the best performance. The accuracy, recall, and F1 values of the model at each threshold are shown in Table 5.

Table 5. Predictive performance of the model at different thresholds.

Threshold	Precision	Recall	F1-score
4	0.80	0.91	0.85
5	0.80	0.91	0.85
6	0.80	0.91	0.85
7	0.79	0.9	0.84
8	0.72	0.86	0.77

As the threshold increases, the model's accuracy gradually decreases. However, considering that in the field of materials, higher prediction scores indicate more reliable predictions, we comprehensively consider and choose a scoring threshold of 6 as the boundary for classifying the model's prediction results as correct or incorrect. In the end, we selected the model with a similarity threshold of 20% and a scoring threshold of 6 as the best model. To demonstrate the effectiveness of this model in predicting the attribution of lithium battery materials, we conducted experiments using traditional machine learning models, generative pre-trained language models, the constructed material knowledge graph, and the large language model integrated with the materials knowledge graph. As shown in Table 6, due to the limited amount of data and the scarcity of material feature information contained in the data, directly training models such as Support Vector Machines (SVM) and logistic regression makes it difficult for the model to capture meaningful features relevant to the prediction problem and accurately capture the relationship between materials and attributions. This leads to poor predictive performance, as the model tends to predict all data as a single type, resulting in a recall rate of 1 and a precision rate of less than 50%. Generative pre-trained models like GPT, which are pre-trained on large-scale data, can leverage domain-specific knowledge to answer different types of questions. However, such general domain models cannot encompass all the specialized knowledge in the materials domain. When directly questioning GPT about lithium battery materials without additional contextual background information, the model cannot fully achieve efficient and accurate answers. According to the experimental results, we found that the predictive performance of the GPT-3.5-turbo model is only about 34%. In comparison, although the GPT-4 model has more parameters and a larger training dataset, enabling it to conduct more extensive and detailed knowledge question-answering, its F1 score in predicting the properties of lithium battery materials is only around 62%. Similarly, we tested the predictive effectiveness of the constructed material knowledge graph on question-answering data. It is evident that only those material attribution associations included in the knowledge graph can be correctly predicted. Simply retrieving information from the knowledge graph cannot achieve inference for unknown material attributions, resulting in an effectiveness of only 51%. The effectiveness of the method proposed in this paper for inferring and predicting material properties was finally evaluated, and the results using different LLMs were compared. From Table 6, it can be seen that the use of LLM combined with the external material knowledge graph for joint reasoning can effectively improve the model's predictive performance.

Table 6. The effectiveness of different inference models.

Inference model	Precision	Recall	F1-score
Logistic regression	0.45	0.1	0.62
SVM	0.45	0.1	0.62
GPT-3.5-turbo	0.67	0.22	0.34
GPT-4	0.83	0.5	0.62
KG	0.61	0.43	0.51
KG+GPT-3.5-turbo	0.9	0.62	0.73
KG+GPT -4	0.8	0.91	0.85

LLM utilizes the material knowledge it has learned and understands the material associations in the knowledge graph, using the similar features between materials for reasoning and providing scores and reasoning basis, allowing researchers to better evaluate the accuracy and reliability of the predictive results. It can be observed that the combination of the material knowledge graph and the GPT-3.5-turbo model achieved an F1 score of around 73%, while our method combined with GPT-4 has a better predictive performance than GPT-3.5-turbo, with an F1 score of 85% To further analyze the advantages of our model compared to regular LLM-based question answering, we conducted a detailed comparison of different models' predictive performance on questions related to the attribution of a specific battery material. As shown in Table 7.

Table 7. Comparison of model response results.

Model	Does 3D-NGS have good cycling performance? (yes)
GPT-3.5-turbo	Yes, 3D-nitrogen-doped graphene sheets are known to have good cycling performance. The nitrogen doping enhances the electrical conductivity and stability of the graphene sheets, allowing them to maintain their performance over multiple cycles. This makes them suitable for various applications, including energy storage devices like batteries and super capacitors, where cycling performance is crucial.
GPT-4	Yes, 3D-nitrogen-doped graphene sheets generally exhibit good cycling performance. This makes them promising for use in energy storage devices like super capacitors and lithium batteries. The doping with nitrogen helps to enhance their electrochemical properties, including their capacity, energy density, and stability over repeated charging and discharging cycles. However, the actual performance can depend on many factors, including the specific method of doping and the conditions of use.
MatKG+GPT-4	The most similar candidate materials: multiwalled carbon nanotube$^{@}$S/N-doped porous carbon$^{@}$polyethylene glycol

	Highest similarity: 0.4285714285714286 Predicted score: 8 Inference reason: Both 3D-nitrogen-doped graphene sheets and multiwalled carbon nanotube@S/N-doped porous carbon@polyethylene glycol have similar structural morphologies. This similarity might lead to some similar properties between these two materials. Besides, due to the fact that nitrogen-doping generally improves the conductivity and stability of carbon materials, the 3D-nitrogen-doped graphene sheets could be inferred to also possess excellent cycling performance. Nonetheless, the different composites of the two materials might also bring about some difference that differs their performance details. Therefore, considering the commonalities and differences, the score of 8 has been given which represents a high possibility but not certain.

GPT-3.5 draws a conclusion about the good cycling performance of 3D-NGS based on its understanding of 3D-NGS and its nitrogen-doping characteristics. Similarly, GPT-4 provides a correct conclusion about good cycling performance based on its learned knowledge and explains the significant impact of nitrogen doping, further highlighting the favorable characteristics of 3D-NGS such as capacity and energy density. In contrast, our model leverages a knowledge graph to retrieve materials with up to 43% similarity to 3D-NGS. Through reasoning based on common properties and similarities, our model reasonably infers that 3D-NGS likely exhibits good cycling performance similar to multiwalled carbon nanotube@S/N-doped porous carbon@polyethylene glycol. The analysis, combined with the nitrogen-doping characteristics of the material itself, enhances the credibility of the results. The model takes into account both the similarities and differences between materials. Unlike GPT, which directly provides a binary answer, our model generates a likelihood score between 0 and 10 based on the inference results. A higher score indicates a higher probability that the target material possesses the predicted attribution, making the results more rational. With a final prediction score of 8, it is evident that there is a high likelihood of 3D-NGS exhibiting good cycling performance. After applying a threshold filter, we confirm that the model has provided the correct predictive answer. Since the knowledge graph cannot comprehensively cover all materials and their relational attributes, some materials exhibit relatively low similarity values. To demonstrate that our model does not solely rely on material similarity for predictions, but rather engages in joint reasoning by combining various material features and semantic information, we conducted a comparison specifically targeting material attribution predictions under conditions of lower similarity, as illustrated in Table 8.

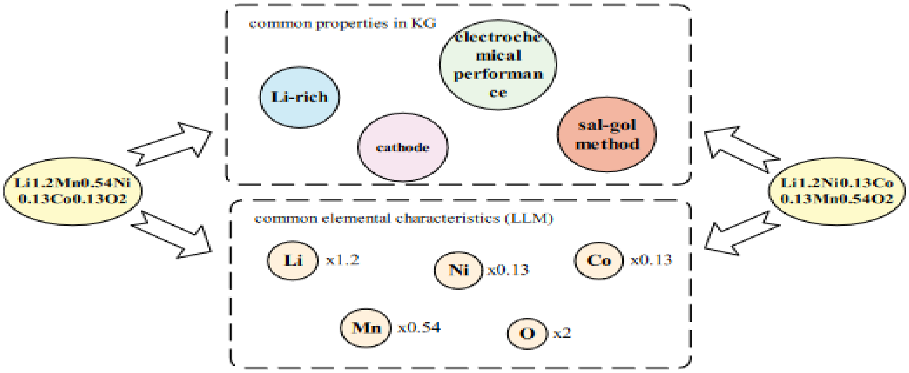
Table 8. Comparison of model response results.

Model	Does Li1.2Mn0.54Ni0.13Co0.13O2 have high capacity? (yes)
GPT-3.5-turbo	Yes, Li1.2Mn0.54Ni0.13Co0.13O2 is considered to have a high capacity as it is an advanced cathode material for lithium-ion batteries. This material has a high specific energy density, which means it can store a larger amount of energy per unit mass. The high capacity of this material makes it a promising candidate for use in high-performance batteries.
GPT-4	Yes, Li1.2Mn0.54Ni0.13Co0.13O2, also known as NMC 811 cathode material, is considered to have a high energy capacity. It is widely used in lithium-ion batteries because of its high energy density and capacity, thermal stability, and good cycling performance.
MatKG+GPT-4	The most similar candidate materials: Li1.2Ni0.13Co0.13Mn0.54O2 Highest similarity: 0.1378830083565464 Predicted score: 9 Inference reason: The two materials, Li1.2Mn0.54Ni0.13Co0.13O2 and Li1.2Ni0.13Co0.13Mn0.54O2, are very similar in chemical ratios and materials. They are both recognized as high-capacity cathode materials in the lithium-ion battery field, especially considered as promising candidates for next-generation high-energy-density batteries. An argument can be made that owing to the high degree of similarity (in terms of Li-rich structure, morphologies, and similar performance) between these two compounds, if one exhibits high capacity, it is highly likely that the other will exhibit it as well. However, slight differences in the structure due to the arrangement of ions - Mn, Ni, and Co-could potentially cause variations in their properties, which is why the score is not a perfect 10. Despite that, based on the given Jaccard similarity and common characteristics, the potential for Li1.2Mn0.54Ni0.13Co0.13O2

	to also demonstrate high capacity is very strong, therefore assigning it a score of 9.

It is observed that the GPT-3.5 and GPT-4 models provide straightforward answers, indicating the high-capacity nature of Li1.2Mn0.54Ni0.13Co0.13O2 and some other characteristics without specifying the underlying reasons. In contrast, using the approach presented in this paper, inference based on similar materials reveals common features between Li1.2Mn0.54Ni0.13Co0.13O2 and Li1.2Ni0.13Co0.13Mn0.54O2, as shown in Figure 10.

Figure 10. Common features from similar material extracted using JRLKG.


common properties in KG
Li-rich
electrochemical performance
cathode
sal-gol method
Li1.2Mn0.54Ni 0.13Co0.13O2
Li1.2Ni0.13Co 0.13Mn0.54O2
common elemental characteristics (LLM)
Li x1.2
Ni x0.13
Co x0.13
Mn x0.54
O x2

This inference considers the shared characteristics obtained from the knowledge graph regarding their lithium-rich features, research methods, and applications as cathode materials. Combining this information with GPT-4's analysis of the identical elemental composition and proportions of the two materials, despite their Jaccard similarity in the knowledge graph being only around 13%, GPT-4 can deduce a high degree of similarity between them. Therefore, it can be inferred that if the material Li1.2Ni0.13Co0.13Mn0.54O2 exhibits high capacity, then Li1.2Mn0.54Ni0.13Co0.13O2 should also possess similar attribution. However, considering the potential impact of minor structural differences on material performance, especially in ion arrangement, a perfect score of 10 is not assigned. Instead, a score of 9 is given, indicating a high likelihood that Li1.2Mn0.54Ni0.13Co0.13O2 has high capacity, without absolute certainty. This underscores the model's ability to accurately predict material attributions despite the relatively low material similarity calculated from the graph, demonstrating its robust semantic analysis capabilities.

CONCLUSION

This study constructed a knowledge graph in the field of lithium-ion battery materials and integrated it with the teacher-level large language model GPT-4. By employing prompt engineering techniques, the LLM was guided to combine its implicit knowledge with the explicit knowledge in the knowledge graph for joint reasoning, enabling the prediction of material attributions. We demonstrated the effectiveness of the model used to build the knowledge graph, ensuring that the graph has high-quality information for better subsequent reasoning. We compared the performance of different inference models in material attribution determination, and the experiments confirmed that our proposed joint reasoning method achieved significant results, exhibiting higher credibility compared to other approaches. Our work provides a novel approach to predicting material performance, and we hope that this method will contribute to the research and development of lithium-ion battery materials. Furthermore, we anticipate that it will foster the integration of knowledge graphs and large language models in the field.

ACKNOWLEDGEMENTS

This work was supported by the Natural Science Foundation of Jiangxi Province (Grant No. 20224BAB212015, 20224ACB202007), Jiangxi Provincial 03 Special Project and 5G Project (20224ABC03A13, 20232ABC03A26), the National Natural Science Foundation of China under grant No. 51962010.

REFERENCES

1. Xu YH, et al. First principles of lithium-ion battery cathode materials. Prog Chem. 2008;20:1827-1833.
2. Zhu XH, et al. First principle calculation of lithiation/delithiation voltage in Li-ion battery materials. Chin Sci Bull. 2011;56:3229-3232.
3. Shi XH, et al. Formation of oxygen vacancy clusters in lithium-rich manganese-based ternary materials for lithium-ion batteries: first principles calculations. Acta Phys Sin. 2023;72:396-405.
4. Zhu RS, et al. Research progress on first principles calculations of lithium-ion battery cathode materials. Dianchi Gongye. 2009;14:273-275.
5. Kumar S, et al. Learning techniques for designing solid-state lithium-ion batteries with high thermomechanical stability. Mater Lett. 2023;351:135049.
6. Houchins G, et al. An accurate machine-learning calculator for optimization of Li-ion battery cathodes. Chem Phys. 2020;153:054124.
7. Nie ZW, et al. Construction and application of materials knowledge graph based on author disambiguation: Revisiting the evolution of LiFePO4. Adv Energy Mater. 2021;11:2003580.
8. Nie ZW, et al. Automating materials exploration with asemantic knowledge graph for li-ion battery cathodes. Adv Energy Mater. 2022;32.
9. Xie T, et al. Large language models as master key: Unlocking the secrets of materials science with GPT. arXiv preprint arXiv. 2023;5:2304.
10. Kauwe S, et al. Data-driven studies of li-ion-battery materials. Crystals. 2019;9:54.
11. Takagishi Y, et al. Machine learning approaches for designing mesoscale structure of li-ion battery electrodes. Batteries. 2019;5:54.
12. Joshi RP, et al. Machine learning the voltage of electrode materials in metal-ion batteries. ACS Appl Mater Interfaces. 2019;11:18494-18503.
13. Liu K, et al. Interpretable machine learning for battery capacities prediction and coating parameters analysis. Control Eng Pract. 2022;124:105202.
14. Louis SY, et al. Accurate prediction of voltage of battery electrode materials using attention-based graph neural networks. ACS Appl Mater Interfaces. 2022;14:26587-26594.
15. Yang W, et al. Time-dependent deep learning predictions of 3D electrode particle-resolved microstructure effect on voltage discharge curves. J Power Sources. 2023;579:233087.
16. Bordes A, et al. Translating embeddings for modeling multi-relational data. Adv Neural Inf Process Syst, 2013;26.
17. Lin Y, et al. Learning entity and relation embeddings for knowledge graph completion. AAAI-24. 2015;29.
18. Ren H, et al. Query2box: Reasoning over knowledge graphs in vector space using box embeddings. arXiv preprint arXiv. 2020;2:2002.05969.

19. Teru K, et al. Inductive relation prediction by subgraph reasoning. ICML. 2020;9448-9457.
20. Zhang M, et al. Link prediction based on graph neural networks. Adv Neural Inf Process Syst. 2018;31.
21. Devlin J, et al. BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv. 2019;2:1810.04805.
22. Brown T, et al. Language models are few-shot learners. Adv Neural Inf Process Syst. 2020;33:1877–1901.
23. Yang L, et al. ChatGPT is not enough: Enhancing large language models with knowledge graphs for fact-aware language modeling. arXiv preprint arXiv. 2023;1:2306.11489.
24. Wang S, et al. Gpt-ner: Named entity recognition *via* large language models. arXiv preprint arXiv. 2023; 4:2304.10428.
25. Feng C, et al. Knowledge solver: Teaching LLMs to search for domain knowledge from knowledge graphs. arXiv preprint arXiv. 2023;1:2309.03118.
26. Wen Y, et al. Mindmap: Knowledge graph prompting sparks graph of thoughts in large language models. arXiv preprint arXiv. 2023;4:2308.09729.
27. Touvron H, et al. Llama: Open and efficient foundation language models. arXiv preprint arXiv. 2023;1:2302.13971.
28. Duomo. Transgpt. 2023.
29. Yu L, et al. Metamath: Bootstrap your own mathematical questions for large language models. arXiv preprint arXiv. 2023;3:2309.12284.
30. Nguyen HT. A brief report on lawGPT 1.0: A virtual legal assistant based on GPT-3. arXiv preprint arXiv. 2023;2:2302.05729.
31. Touvron H, et al. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv. 2023;2:2307.09288.
32. Taori R, et al. Stanford alpaca: Aninstruction-following LLaMA model [EB/OL]. 2023;05-30.
33. Yang J, et al. YEDDA: A lightweight collaborative text span annotation tool. arXiv preprint arXiv. 2018;3:1711.03759.