communications materials

Article

https://doi.org/10.1038/s43246-024-00708-9

Data extraction from polymer literature using large language models

Sonakshi Gupta[1,4], Akhlak Mahmood[2,4], Pranav Shetty[1], Aishat Adeboye[3] & Rampi Ramprasad[2]

Automated data extraction from materials science literature at scale using artificial intelligence and natural language processing techniques is critical to advance materials discovery. However, this process for large spans of text continues to be a challenge due to the specific nature and styles of scientific manuscripts. In this study, we present a framework to automatically extract polymer-property data from full-text journal articles using commercially available (GPT-3.5) and open-source (LlaMa 2) large language models (LLM), in tandem with the named entity recognition (NER)-based MaterialsBERT model. Leveraging a corpus of ~2.4 million full text articles, our method successfully identified and processed around 681,000 polymer-related articles, resulting in the extraction of over one million records corresponding to 24 properties of over 106,000 unique polymers. We additionally conducted an extensive evaluation of the performance and associated costs of the LLMs used for data extraction, compared to the NER model. We suggest methodologies to optimize costs, provide insights on effective inference via in-context few-shots learning, and illuminate gaps and opportunities for future studies utilizing LLMs for natural language processing in polymer science. The extracted polymer-property data has been made publicly available for the wider scientific community via the Polymer Scholar website.

The field of materials informatics[1,2] suffers from lack of data readiness and data accessibility. Although materials data can be systematically generated through computational and physical experiments, a substantial amount of historical data is trapped in published literature. An ever-growing volume of data is continually released in scientific journal articles, but this data frequently exists in unstructured natural language text formats, posing challenges for immediate utilization by modern informatics that rely on the availability of structured datasets. Natural language processing (NLP) techniques implemented in materials science seek to automatically extract materials insights, materials properties, and synthesis data from a corpus of text documents, and propose hypotheses and designs for new materials[3–5]. After acquiring the corpus, a series of complex NLP operations are performed which include turning texts into smaller units called tokens, recognizing key entities (such as materials, characterization methods, or properties) using named entity recognition (NER) methods, creating rule-based algorithms to identify relationships between the entities through dependency parsing, and finally, extracting information and organizing it into a structured format[6].

With the advent of modern machine learning (ML) and artificial intelligence techniques, deep learning models including recurrent neural networks and long short-term memory architectures have become valuable for NER tasks[7,8]. In recent years, the transformer-based BERT architecture, with its ability to capture contextual and semantic relationships within scientific texts[9], has especially demonstrated superior performance compared to traditional neural network models[10]. We have previously developed and published MaterialsBERT[11], a NER model derived from PubMedBERT[12]. This model demonstrated superior performance on publicly available datasets in comparison to other BERT models, including ChemBERT[13] and MatBERT[14], particularly for materials science-specific data extraction tasks. By employing a MaterialsBERT-based pipeline, we successfully extracted over 300,000 polymer-property records from approximately 130,000 abstracts, the largest such undertaking at that time, with the data made publicly available[15]. This data extraction approach demonstrated the effectiveness of the MaterialsBERT model in processing a substantial volume of abstracts to obtain polymer-property information. The potential for large-scale data extraction using MaterialsBERT from the full texts of journal articles presents a further new opportunity for materials data acquisition.

While NER models excel in identifying named entities within texts, discerning entity relationships across extended passages encompassing

[1]School of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, GA, USA. [2]School of Materials Science and Engineering, Georgia Institute of Technology, Atlanta, GA, USA. [3]School of Chemical and Biomolecular Engineering, Georgia Institute of Technology, Atlanta, GA, USA. [4]These authors contributed equally: Sonakshi Gupta, Akhlak Mahmood. e-mail: rampi.ramprasad@mse.gatech.edu

multiple sentences solely through recognized named entities continues to be a challenge[16]. This limitation is particularly pronounced in technical and scientific documents, where critical information is often expressed in a non-standard and complex manner. In the domain of polymer science, NER-based extraction methods encounter additional specific challenges stemming from the expansive chemical design space of the materials and the utilization of non-standard nomenclature, including commonly used names, acronyms, synonyms, and historical terms[17].

Recently, large language models (LLMs) such as Generative Pretrained Transformer (GPT), Large Language Model Meta AI (LlaMa), Pathways Language Model, etc., have gained significant attention in the field of natural language processing[18,19]. These models have shown remarkable performance in handling various NLP tasks, showcasing their robustness and versatility[20], especially in high-performance text classification, NER, and extractive question answering with limited datasets[21]. A key factor contributing to the success of the LLMs is the vast amount of 'knowledge' these models gain during semi-supervised pre-training (e.g., using masked language modeling to predict the next token given a set of preceding tokens for context)[22]. In the pre-training phase, LLMs acquire a foundational comprehension of language semantics and contextual understanding through exposure to training datasets, which typically comprise texts from general science and scientific literature[23]. Subsequently, the pre-trained LLMs, also referred to as base models, undergo supervised fine-tuning to produce desired text outputs in response to specific prompts or instructions. Examples include OpenAI Codex and Code LlaMa, both of which are fine-tuned to generate code snippets based on a given natural language input[24]. Similarly, ChatGPT and LlaMa Chat models are language models fine-tuned to respond to user prompts or instructions conversationally while maintaining a history of previous interactions for added context for the conversation. A human-like understanding of the language semantics and subsequent instruction tuning thus enable the LLMs to perform in-domain tasks such as information extraction about a specific material class with no (zero-shot) to only a few task-specific examples (few-shots). Such ability offers excellent performance and eliminates the efforts needed to create a labeled dataset of significant volume and train or fine-tune a new model[25].

Despite the potential for many use cases including data extraction, the improved capabilities of the LLMs depend on access to significant computational resources. Using LLMs for inference incurs significant monetary costs, due to high demands of energy consumption, hardware or cloud computing time, and in terms of the environment, due to the carbon footprint of powering a number of modern tensor processing units[26,27]. Therefore, a data extraction pipeline aiming to efficiently utilize LLMs should extract the maximum amount of high-quality information and at the same time reduce the unnecessary prompting of the LLMs during the processing of millions of full-text scientific articles.

Limited prior works exist on the application of LLMs for data extraction in materials science. Dagdelen et al. fine-tuned GPT-3.5 and LlaMa 2 models to extract useful records of linking dopants and host metal-organic frameworks[28]. Zheng et al. developed a workflow utilizing ChatGPT as a collaborator for human chemists, extracting 26,257 distinct synthesis parameters of approximately 800 metal-organic frameworks from 228 articles[29]. Polak and Morgan proposed a similar workflow for metallic glasses and high entropy alloys, employing follow-up questions to GPT-4 to ensure correctness and address the issues of hallucinations with LLMs[30]. Similarly, Yang et al. used a repeated questioning strategy with GPT-4 for bandgap values, demonstrating reduced error rates and a more extensive dataset than human-curated databases[31]. GPT-based approach offered high-performance text classification, NER, and extractive question answering with limited datasets, and could reduce researcher workload by producing initial labelling sets and verifying human-annotations.

In this contribution, we present an approach to employing LLM- and NER-based pipelines, specifically designed to automate the extraction of property data of polymers from the full-text contents of journal articles. Our data extraction workflow, depicted in Fig. 1, processes a corpus of 2.4 million materials science journal articles published in the last two decades, from which, we identify and concentrate on 681,000 polymer-related articles. Subsequently, the paragraphs of the articles are processed through a dual-stage filtering scheme consisting of a 'heuristic filter' and a 'NER filter' to identify the most relevant paragraphs that contain extractable property data. The materials and properties are identified, relationships are established, and the information is extracted in a structured format using MaterialsBERT and GPT-3.5 models independently. Our pipelines extracted more than one million values of 24 selected properties from the full texts of the polymer-related articles. We have made the extracted data publicly available at polymerscholar.org (henceforth referred to as Polymer Scholar) where researchers can explore the distribution and relationships within the properties of polymers[15]. To identify the most efficient model, with a special focus on optimizing quality and costs, we evaluate three models – Materi-

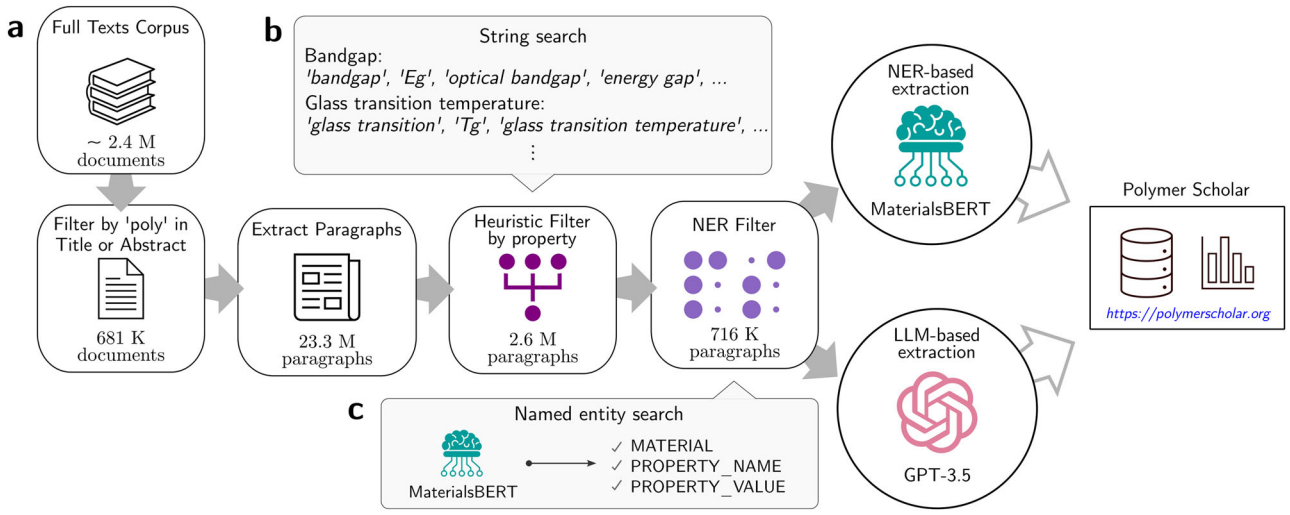

Fig. 1 | Overall workflow to extract polymer property data. a Polymer-specific documents are selected from a corpus of 2.4 million materials science journal articles. Multiple stages of filtering select the most relevant documents and paragraphs of the documents before performing data extraction by MaterialsBERT and GPT-3.5. Extracted data are finally deposited to a relational database of the Polymer Scholar web interface. **b** Property-specific paragraphs are selected by a heuristic filter based on string matching and dictionary lookup. **c** The NER filter identifies paragraphs with extractable named entities. The LlaMa-2 large language model was also evaluated, but was not used in the final data extraction pipeline due to comparatively low performance and long inference time, as described later in the text.

alsBERT, GPT-3.5, and LlaMa 2, across four critical performance categories: quantity, quality, time, and cost of data extraction. Our study undertakes a thorough examination of the capabilities of the LLMs, juxtaposing their performance against MaterialsBERT. We also present results offering insights into optimizing the performance and costs associated with using LLMs for data extraction via in-context few-shots learning and analyzing the general trends, characteristics, and distributions of the extracted full text data. We conclude by addressing the remaining challenges and looking ahead at the future potential of utilizing LLMs for informatics tasks specific to polymer science.

Results and discussion

Overview of the data extraction pipelines

We assembled a corpus comprising more than 2.4 million materials science journal articles published over the last two decades. The articles were initially indexed through the Crossref database, followed by authorized downloads from 11 publishers, including Elsevier, Wiley, Springer Nature, American Chemical Society, and the Royal Society of Chemistry. Further details regarding the articles can be found in the Methods section and in ref. 32. Specifically focusing on polymer-related content, we identified 681,000 documents by searching for the term 'poly' in the title and abstract of the articles. Extracting information from these polymer-related documents involved treating individual paragraphs as text units, resulting in a total of 23.3 million paragraphs. To extract data from the selected paragraphs, we targeted 24 properties of polymers based on their significance and downstream usage. Commonly reported thermal and optical properties were selected for their efficacy in training multi-task ML models, using highly correlated properties as substitutes for less prevalent ones. Additionally, properties that are beneficial for various polymer application areas were included. For instance, the bandgap and refractive index are vital for dielectric aging and breakdown, gas permeability properties are crucial for filtering and distillation applications, and mechanical properties are significant for thermosets and recyclable polymers. A list of the polymer properties selected for extraction can be found in Table 1.

A two-step filtering system was used to avoid unnecessary prompting of LLMs by ignoring texts that do not have extractable and complete data. First, as illustrated in Fig. 1b, each paragraph was passed through property-specific heuristic filters to detect paragraphs that mention a target polymer property or its co-referents manually curated via literature review. Approximately 2.6 million paragraphs (~11%) successfully passed the property-specific heuristic filters, indicating relevance to the selected 24 properties of polymers. Subsequently, an additional NER filter was applied to identify paragraphs containing all necessary named entities such as material name, property name, and property value (Fig. 1c) to confirm the existence of a complete extractable record. This refined filtering stage yielded about 716,000 paragraphs (~3%) containing texts relevant to the selected 24 properties. Regardless of the final data extraction model, the NER filter is utilized to verify the presence of 'material', 'property', 'value', and 'unit' entities in the given paragraph because the absence of any of these entities would preclude the extraction of a complete data point by the models. This filter thus assists LLMs in accurately identifying relationships without which they may generate placeholder values such as 'not mentioned', 'n/a', or '-', or even hallucinate false data if an entity is not present in the text.

The texts of the filtered paragraphs are then passed to either MaterialsBERT for NER-based data extraction, or to the OpenAI API for GPT-3.5-based data extraction. During the extraction process, the relationship extraction module of MaterialsBERT processes the identified entities to determine and establish correct relationships using heuristic rules. GPT-3.5, on the other hand, automatically identifies relationships between the entities by itself. Finally, the extracted data undergo post-processing, validation, and deposition into a relational database and the data stored in the database is made publicly accessible for visualization via a user-friendly web interface of Polymer Scholar. In total, the pipelines extracted over one million polymer-property records from the full texts. Data extracted by GPT-3.5 (MaterialsBERT) from the full text is approximately 21 times (12 times) than what was collected purely from the abstracts in our previous work for the selected 24 properties. Additional details about the different stages of the pipelines are discussed in the Methods section, with specific details on data extraction using MaterialsBERT provided in ref. 11.

Table 1 | List of the selected 24 property names and the corresponding number of property values extracted using the GPT-3.5 and MaterialsBERT pipelines

Property	GPT-3.5 Full text	MaterialsBERT Full text	MaterialsBERT Abstract
Glass transition temperature	125,585	75,722	6155
Melting temperature	76,577	41,766	1615
Thermal decomposition temperature	70,285	19,817	1479
Lower critical solution temperature	20,115	11,658	712
Crystallization temperature	12,863	4045	605
Thermal conductivity	5574	10,300	1429
Upper critical solution temperature	1486	581	50
Bandgap	63,361	30,732	2245
Ion exchange capacity	3118	4656	1034
Refractive index	18,982	9785	576
Tensile strength	63,014	38,773	4382
Young's modulus	40,148	32,207	1904
Elongation at break	30,754	15,072	1499
Compressive strength	12,343	6879	814
Flexural strength	7201	3543	313
Hardness	5271	1984	244
Water contact angle	84,601	63,685	3932
Water uptake	15,991	6019	330
Limiting oxygen index	7893	6606	1146
CO_2 permeability	2943	2561	685
Swelling degree	1880	1995	71
Methanol permeability	1228	852	174
O_2 permeability	735	503	99
H_2 permeability	501	1072	46
Total	672,449	390,813	31,539

The number of data records extracted from the abstracts alone using MaterialsBERT are taken from ref. 11 for comparison against the full text data extraction.

Data extraction using large language models

Polymer-related property data from a journal article can be extracted by leveraging the ability of LLMs to understand the specialized semantics of materials entities discussed in the text. However, obtaining the desired output from an LLM poses a challenging task and is presently a subject of active research[33,34]. Even with the same prompt or instruction and text generation parameters, the LLMs can produce varying responses[35]. The effectiveness of simple natural language prompts in eliciting desired results from LLMs is not always straightforward, due to the models' interpretation methods often being non-intuitive[36]. Hence, it is crucial to employ techniques and parameters that can minimize the variability in the generated responses.

One of the intriguing capabilities of LLMs, referred to as in-context few-shot learning, is their ability to learn from examples (often termed as 'shots') prepended to the prompt[25]. The response generation can be sufficiently influenced by the shots and the prompt given to the model as inputs. For few-shot learning, we used a pool of examples containing manually curated 595 glass transition temperature (Tg) values and 356 bandgap values from 630 abstracts. Given the necessity for the LLM's response to be presented in a structured format for seamless programmatic extraction of material names and property values, we experimented with different prompts. We finally selected one that directs the model to identify entities of interest, reading: "Extract all `<property>` values in JSONL format with 'material', 'property', 'value', 'condition' columns." The placeholder `<property>` is replaced with the desired polymer property names we chose to extract. A sample shot, shown in Fig. 2a, displays glass transition temperature data manually curated from the abstract of ref. 37 and the corresponding prompt. The formatted response in the example adheres to the JSONL structure, serving as a demonstration to the model that its generated response should precisely follow the same format. The shot is followed by the actual prompt containing the input text from which data needs to be extracted. Fig. 2b provides an illustrative example of such a prompt, comprising a paragraph taken from the full text of ref. 38, along with specific instructions given to the LLM for data extraction. The resulting response from GPT-3.5 is shown in Fig. 2c, revealing three data points of glass transition temperature values extracted by the LLM.

Our similarity-based shot selection method, illustrated in Fig. 2e, provides a way to determine the most suitable example from the pool of examples for inclusion as a shot along with the LLM prompt. We first performed k-means clustering (with $k = 10$) on the word embeddings of the examples. The embeddings of the examples and input text were determined using the MaterialsBERT text encoder. Subsequently, we selected the example corresponding to the centroid of the cluster closest to the input text which is sent to the LLM for data extraction. This method, as opposed to random selection, allows us to choose an example that closely resembles the text from which data needs to be extracted.

To assess the monetary costs associated with text generation using GPT-3.5, we counted the number of OpenAI tokens in the input prompts and shots while extracting property data from the 630 manually curated abstracts. The tokenization process employed by the LLMs depends on the linguistic characteristics and contextual nuances of the words, numbers, punctuations, and symbols present in a given text. Our initial evaluations nevertheless reveal a direct correlation between the number of tokens and the word count within the selected abstracts, as depicted in Fig. 2f, which provides an approximate but simpler way to understand the effects of the text lengths on computational expenses associated with using GPT-3.5. Costs during text generation directly increase with the number of words being processed. Furthermore, the introduction of multiple shots as an additional component in the input requires the LLM to consider extended textual inputs during response generation. A linear increase in both token

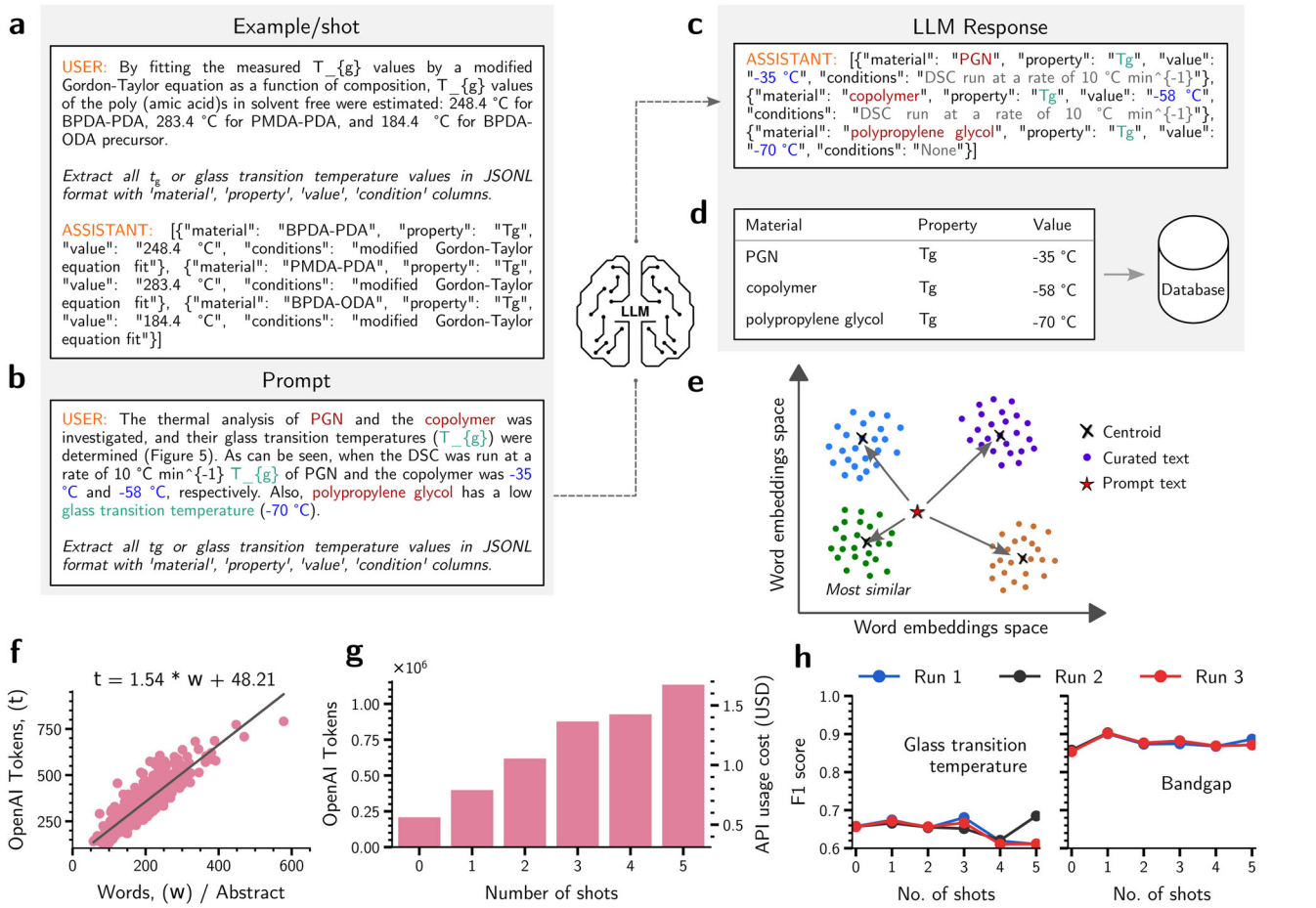

Fig. 2 | Data extraction using large language models. a Example of a shot generated from the text of a manually curated dataset containing glass transition temperature and bandgap values. **b** Prompt used to extract data using LLM. **c** Response generated by GPT-3.5 and **d** Tg data extracted from the response text. **e** Schematic illustration of clusters formed by word embeddings of the texts of 630 manually curated abstracts and choice of the shot most similar to the text from which data is to be extracted. **f** Correlation between token count and number of words in 630 abstracts related to polymers. The positive y-intercept is attributable to the average expected number of punctuation marks and symbols typically found in the texts. **g** Barchart depicting an increase in the number of tokens and the ultimate cost for OpenAI API usage for multiple shots. **h** Effects of multiple shots on the accuracy of the extracted Tg and bandgap data from the manually curated 630 abstracts.

utilization and corresponding API usage can be observed in Fig. 2g with an increasing number of shots used to extract data from the 630 abstracts.

Despite the expectation that the model's performance would improve with an increased number of shots, we consistently observed optimal results when providing only a single shot to GPT-3.5 while prompting for extraction of Tg and bandgap values (see Fig. 2h). A plausible explanation for this phenomenon could be that the model learns the structure of the anticipated output immediately from a single shot and experiences disorientation when additional shots are added to the prompt. Based on these observations, we proceeded to use one shot while prompting the LLM to extract data from a given text.

Performance benchmarking for a labeled subset of the full corpus

We employ NER- and LLM-based extraction methods to comparatively assess the validity and reliability of different data extraction pipelines. The primary objective of this assessment is to identify the most effective extraction methods while emphasizing the optimization of computational and monetary costs. Consequently, the evaluation involves a subset of 1000 articles from the larger pool of the 681,000 polymer-related papers. In this subset, we have manually curated data from the abstracts of 630 articles, which reported one or more Tg and bandgap values in their abstracts and were selected randomly. The rest of the 370 articles were randomly chosen from the polymer papers. Bar charts containing the distribution of the selected articles compared to the full corpus are shown in Fig. S1.

The assessment pipeline, depicted in Fig. 3a, involves parsing the full texts of the selected papers into paragraphs, resulting in a total of 37,434 paragraphs. As discussed in the previous section, two filtering stages are employed to select the most relevant paragraphs for a target property and paragraphs containing extractable data. In the first stage, the property-specific heuristic filter is applied, resulting in a reduction of the paragraph count to 12,817. Subsequently, the second stage utilizes the MaterialsBERT-based NER filter, further narrowing the selection to 6179 paragraphs. The final data extraction process involves three models: MaterialsBERT, incorporating NER and rule-based entity recognition and relationship extraction; the open-source 70 B LlaMa-2 model developed by Meta AI; and the commercially available GPT-3.5 model hosted by OpenAI. To determine the optimal shot for each prompt in the LLM-based data extraction process, we compared our strategy based on similarity to a random selection of shots from the curated data pool. We termed these approaches as "Similar" and "Random" shot selection methods, respectively.

In terms of quantity, (Fig. 3c, d) GPT-3.5 demonstrates significant superiority over the other models. It extracted the largest amount of data, comprising 4706 material-property value pairs using Random-shot and 4589 pairs using Similar-shot selection from the selected 6179 paragraphs.

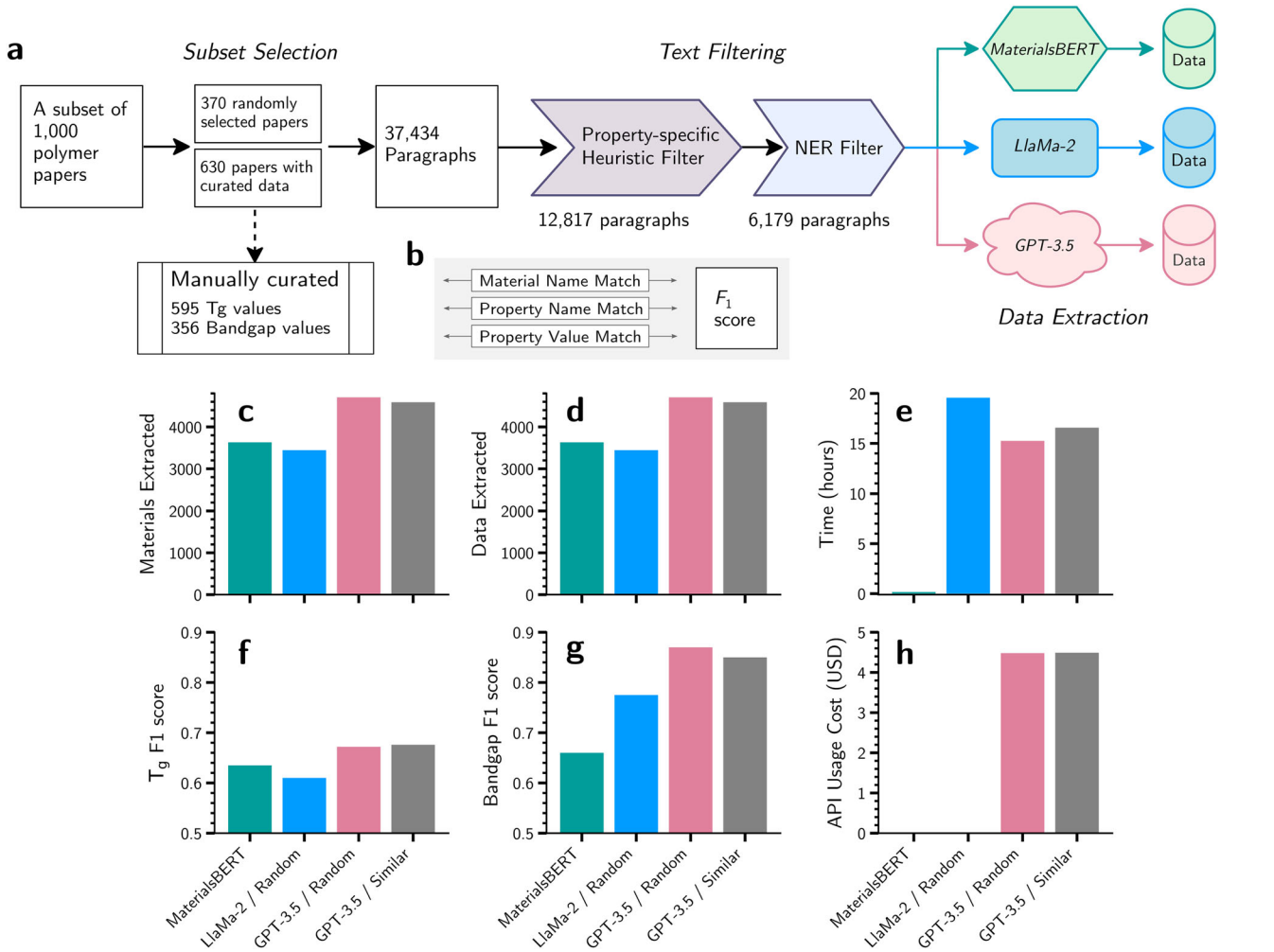

Fig. 3 | Performance evaluation of NER and LLM pipelines. a Overview of the pipelines used to measure the performance of MaterialsBERT, LlaMa-2, and GPT-3.5 for data extraction from 1000 polymer documents. **b** Representation of F_1 score measurement using manually curated data. **c**, **d** Total number of materials and property data extracted using the pipelines, **e** time spent running the pipelines, **f**, **g** calculated F_1 scores for Tg and bandgap respectively and **h** incurred cost due to API usage by the pipelines.

The NER-based MaterialsBERT pipeline extracted 3631 material-property pairs, slightly outperforming LLaMa, which extracted 3441 data pairs.

To assess extracted data quality, we checked if the extracted material name, property name, and property value (including unit) match the manually curated records (Fig. 3b) to calculate the F_1 scores. Our F_1 score computation methodology necessitates that the extracted data be completely present within the provided text. Thus, to ensure precise data extraction, it is imperative to identify all entities, specifically 'material', 'property', 'value', and 'unit' within the specified paragraph. Furthermore, to verify the correct relationships among these entities, they must correspond accurately with their respective entities in the curated dataset to account for hallucination by the LLMs. The F_1 score calculation method not only verifies the accuracy of the completely extracted data but also confirms their origin from the appropriate source paragraph, because any semantically accurate but fabricated data points produced by LLMs would be absent in the curated ground truth data extracted from the provided text. Additional details about the calculation of the F_1 score are discussed in the Supplementary Discussion.

We found that the performance of the models is contingent on the property being extracted. Both LLMs exhibit superior performance relative to the NER-based MaterialsBERT pipeline in extracting data for bandgap, while accuracy declines when extracting data for Tg (Fig. 3f, g). GPT-3.5 achieved the highest F_1 score of 0.67 for Tg, with the Similar-shot selection method slightly outperforming the Random selection method. MaterialsBERT and LlaMa obtained F_1 scores of 0.63 and 0.64, respectively. LlaMa 2 outperformed the comparatively lower F_1 score of 0.66 achieved using MaterialsBERT, particularly during the extraction of bandgap, where it achieved an accuracy of 0.77. GPT-3.5 once again secured the highest F_1 scores of 0.87 and 0.85 for the Random and Similar-shot selection methods, respectively. We previously demonstrated the superior performance of MaterialsBERT compared to other existing NER models[11]. The higher F_1 scores compared to the MaterialsBERT obtained in this work thus underscores the advantage of the LLMs over the existing NER models.

Concerning computational efficiency and monetary costs, MaterialsBERT emerges as the most advantageous choice (Fig. 3e). Operated in-house, MaterialsBERT processed the 6179 paragraphs in under half hour without incurring any financial costs. LlaMa-2, also hosted locally, imposed no direct monetary costs but demonstrated the longest inference time, attributable to its substantial model size of 70 billion parameters running on four Nvidia Quadro GP 100 GPU cards. In contrast, the commercial LLM, GPT-3.5, required API calls to OpenAI's servers for inference, introducing a direct financial cost of $4.48 (for ~2.9 million tokens) for each of the shot selection methods (Fig. 3h).

This thorough evaluation allowed us to identify the best models for data extraction from the full corpus. Given GPT-3.5's superior performance in both quantity and quality and MaterialsBERT's optimal cost efficiency, we chose to incorporate GPT-3.5 with Similar-shot selection and MaterialsBERT in our final pipeline to extract data for all the selected properties from the entire corpus of polymer articles.

Data extraction from full texts

Having selected the best-performing models, we extracted data for the 24 selected properties from the full texts of 681, 000 polymer-related journal articles using the NER-based MaterialsBERT and the LLM-based GPT-3.5 pipelines. Given that neither pipeline can achieve perfect accuracy, and manually curating data sets for all 24 properties requires a significant effort, we conducted additional validation of the extracted data in a post-processing step. After programmatically obtaining the data from the MaterialsBERT pipeline and JSONL responses generated by GPT-3.5, we verified if the property name matched one of the selected 24 property names or their known variations. We standardized the extracted values and units within the extraction pipelines, such as converting kPa or GPa to MPa, K to °C, etc. Subsequently, in the post-processing stage, we checked if the unit of the extracted data matched the unit corresponding to the selected property. Additionally, we assessed if the extracted value for each property fell within a specified minimum and maximum range, that was manually assigned based on literature review. We ignored any extracted data that does not satisfy these post-processing validation criteria. In addition to the polymers, the pipelines also extracted property data for other classes of materials. To identify the polymers, we checked if the extracted material is a valid polymer name by cross-referencing it with a comprehensive, albeit non-exhaustive, list of polymer names manually collected from the literature.

From the GPT-3.5 pipeline, we extracted 672,449 polymer-property records, and from the MaterialsBERT pipeline, we obtained 390,813 records for the 24 selected properties that passed the validation stage (see Table 1). In assessing the number of data points extracted for the selected properties, the GPT-3.5 pipeline demonstrated superior performance compared to the MaterialsBERT pipeline. Specifically, with the full text, the LLM extracted data volume was 72% greater than the data extracted by the NER pipeline, and 21 times the data extracted from abstracts (using the NER pipeline) in ref. 11. Among the extracted data, thermal and mechanical properties were more commonly found in the literature, while data on gas permeability was comparatively sparse.

Figure 4a, b illustrates the distribution of extracted Tg and bandgap data, respectively. For comparative purposes, we have also presented the distributions of Tg and bandgap data extracted from abstracts using MaterialsBERT which show a significantly higher amount of data extracted from the full texts. Though not obvious in the distribution of extracted data from abstracts, a bimodal distribution of Tg values and an elongated tail can be observed for all pipelines, demonstrating the presence of extreme property values. A comparative analysis of full-text extraction using MaterialsBERT reveals 75,722 valid Tg records, which is about 12 times the data extracted from abstracts earlier. This corresponds to 20,511 unique materials. Further, it provides 30,732 valid bandgap records, indicating 13 times the abstract extracted data and corresponds to 10,627 unique materials. In contrast, the GPT-3.5 pipeline provides a higher volume of valid records, amassing 125,585 Tg records, a 65% increase over MaterialsBERT and 20 times the data extracted from abstracts. This pertains to 69,740 unique materials. Similarly, GPT-3.5 yielded 106% more bandgap data than MaterialsBERT (63,361 records), which is 28 times the data obtained from abstracts, for a total of 31,337 unique materials.

Upon comparing the valid data points of Tg and bandgap that passed the post-processing criteria, we observed that each pipeline retrieved data from the source paragraphs where the other pipeline encountered difficulties (Fig. 4c). This supports the previously reported F_1 scores, demonstrating that GPT-3.5 is capable of understanding more intricate relationships and extracting more data. Notably, there is a significant number of paragraphs where MaterialsBERT fails ($F_1 = 0.66$), in contrast to the success of GPT-3.5 ($F_1 = 0.85$). Specifically, GPT-3.5 extracted data from an additional 47,966 paragraphs, from which MaterialsBERT failed to extract any valid data. Conversely, MaterialsBERT successfully extracted data from 7311 paragraphs where the GPT-3.5 pipeline did not retrieve any valid data.

The combined data extraction efforts of both pipelines resulted in a total of 113,099 unique materials for Tg and bandgap. Notably, only 11,042 material names exactly matched between the pipelines, as illustrated in Fig. 4d, and parity plots showing overlap between the pipelines in Fig. S2. GPT-3.5 demonstrated its proficiency by extracting an additional 88,128 material names that were not captured by MaterialsBERT. This significant increase in the identification of new material names can be attributed to GPT-3.5's inclination towards detailed extraction of composition and types while extracting material names (see Table S3). It is important to mention that variations in naming and co-referents of polymers were considered correct during the calculation of the F_1 score. Consequently, although there is a low overlap of the names exactly extracted by the methods as depicted in Fig. 4d, a high F_1 score is still maintained. For comparison, similar Venn diagrams for Tg and bandgap data extracted from the 630 abstracts are shown in Fig. S1. Upon further investigation, we found that MaterialsBERT in some cases faced challenges in correctly associating property and material names in texts filled with numerous numbers and values. This was

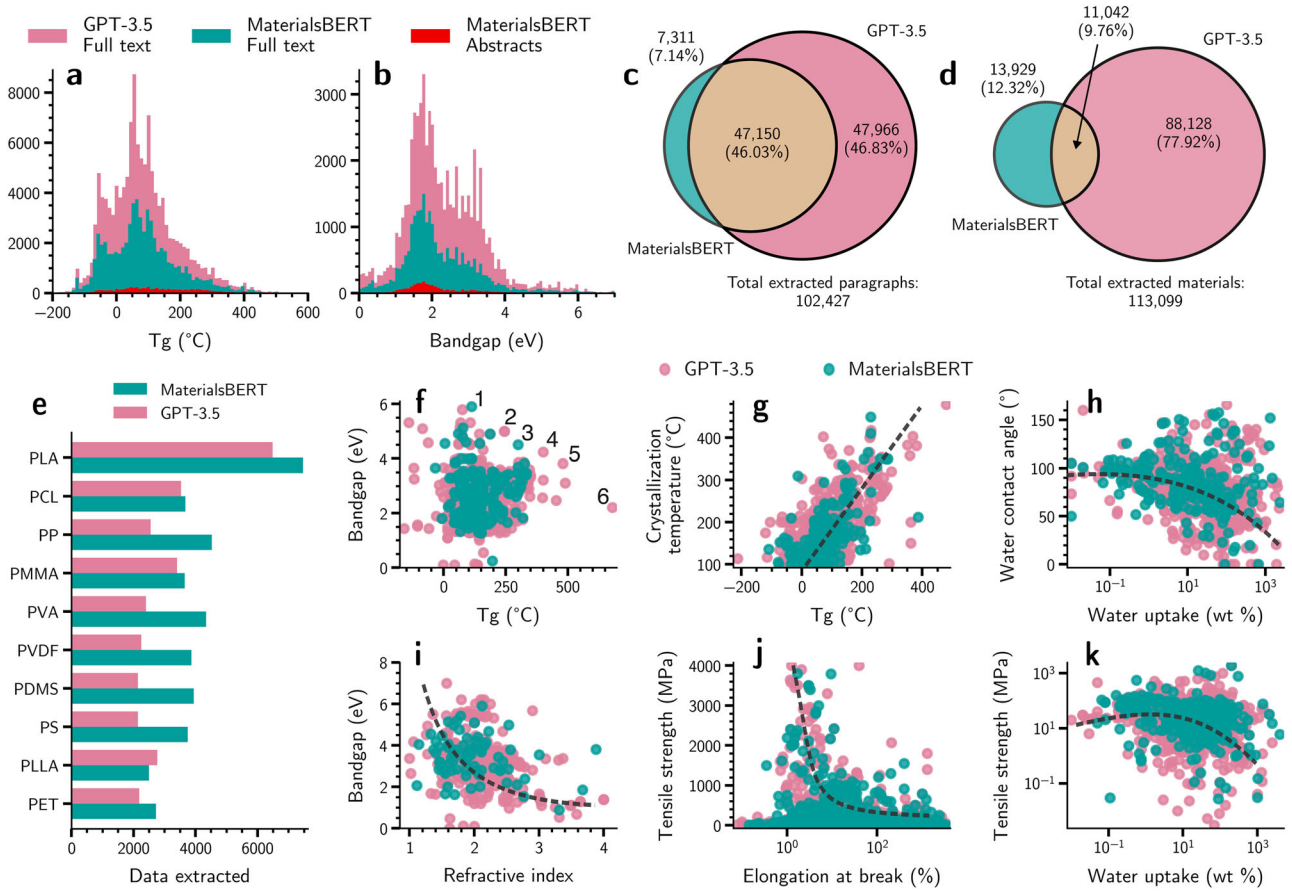

Fig. 4 | Comparison of GPT and MaterialsBERT extracted data. a, **b** Distribution of extracted Tg and bandgap data from the abstracts using MaterialsBERT and full texts of polymer articles using both GPT-3.5 and MaterialsBERT pipelines. **c**, **d** Overlap between the source paragraphs and extracted material names in GPT-3.5 and MaterialsBERT extracted Tg and bandgap data. **e** Barchart showing the total number of data extracted from the full texts for the top 10 polymers using the two pipelines. **f** Pairwise plot of bandgap and Tg values for the materials extracted from the same source articles, marked points are discussed in the text. **g**–**k** Representative pairwise plots showing relationships between selected pairs of properties. The dashed lines are guides for the eye.

particularly prevalent in cases involving polymer blends or composites, where MaterialsBERT often extracted redundant values for each polymer. As a result, the total amount of data extracted by the MaterialsBERT pipeline is often erroneously inflated for many ordinary polymers, as shown in Fig. 4e. GPT-3.5 exhibited significantly enhanced efficiency in accounting for composite and blend compositions in the sentences. Examples elucidating the F-1 score calulcation method and anomalies of MaterialsBERT are discussed in the Supplementary Discussion.

We plotted the pairwise distributions where values of two properties were available for the same material name and were extracted from the same source article. Fig. 4f depicts the relationship between bandgap and Tg data extracted by the pipelines. GPT-3.5 demonstrated the ability to capture more data including numerous extreme property values, while MaterialsBERT successfully captured values towards the center of the distribution. Additionally, both methods illustrate the scarcity of materials exhibiting high bandgaps and simultaneously high Tg values. Six data points with high bandgap or Tg values, as determined by the pipelines, are identified and numbered in Fig. 4f. The pipelines correctly extracted points 1, 2, and 3. In the case of point 1, GPT-3.5 managed to extract more detailed information, identifying both 'non-oriented PMMA layer' and 'oriented PMMA layer' instead of merely labeling it as 'PMMA'. Neither GPT-3.5 nor MaterialsBERT were successful in extracting any bandgap data for points 2 and 3, respectively. With regard to point 4, GPT-3.5 incorrectly identified the thermal decomposition temperature as the Tg value. Meanwhile, MaterialsBERT was unable to extract the high bandgap value. Points 5 and 6 were identified as a composite and blend, respectively. However, MaterialsBERT only managed to extract the polymer names, failing to recognize the presence and modifications by other materials. Despite the observation that GPT-3.5 typically extracts the compositions of polymer composites and blends, it did not identify the presence of other materials in this instance and only extracted the polymer names due to the complexity of the sentences in the source text. Out of the total 12 values for the designated property pairs, 10 were accurately extracted. Supplementary Information, including references and actual values of the marked points, can be found in Tables S2 and S3.

Correlations between extracted properties

Knowledge of the relationships between distinct material properties can be gained by examining the pairwise distributions among other property pairs. Representative pairwise plots indicate diverse trends, with Fig. 4g highlighting a discernible positive correlation between crystallization temperature and Tg. The extracted data highlights an inverse correlation between water contact angle and water uptake (Fig. 4h), confirming that hydrophilic materials with smaller contact angles tend to absorb greater amounts of water. Additionally, the bandgap determines the energy above which a material remains transparent. As the light wavelength decreases towards the bandgap, there is a corresponding increase in the refractive index. When the light wavelength is held constant, materials with a larger bandgap generally exhibit a smaller refractive index. The inverse trend depicted in Fig. 4i illustrates this relationship between the optical properties of the materials extracted. Another fundamental observation for polymers is the inherent trade-off between mechanical properties. Fig. 4j elucidates the negative

trade-off between tensile strength and elongation at break, emphasizing the capacity of GPT-3.5 to capture additional values. The phenomenon of water absorption induces polymer chain swelling, thereby instigating plasticizing effects that can alter the mechanical properties of the material. Although the specific nature of the relation depends on the material in question, Fig. 4k suggests a pervasive negative trend wherein water uptake causes a reduction in the tensile properties of materials. Overall, the extracted data largely follow expected trends and agree with domain knowledge as demonstrated by the pairwise distributions.

The datasets we extracted from literature containing the 24 properties are integral for training downstream ML models. In a previous study, we employed similar datasets to optimize the material system and develop robust predictive models to optimize power conversion efficiency of polymer solar cells and demonstrated a significant reduction in new polymer discovery time[39]. In another study, we assembled a comparable dataset to predict the retrosynthesis pathways for a target polymer[40].

Outlook

LLMs, such as GPT-3.5 with likely over 200 billion parameters, show a marked advantage in data extraction quality and ease due to their pre-training on extensive text corpora, even without fine-tuning on domain-specific datasets. The efficacy of pre-training is highlighted by GPT-3.5's proficiency in recognizing material and chemical entities. The LlaMa-2 model, with 70 billion parameters, demonstrates comparatively limited capability in recognizing chemical entities and establishing correct entity relationships, hinting at consideration for potential improvement through fine-tuning on labeled datasets. The challenges specific to polymer literature for NER-based models are marked by the absence of a standardized naming convention for polymers and the requirement for manual efforts to identify entity relationships.

Despite the promising performance of GPT-3.5, various limitations still exist for the extraction of data from polymer literature and their applications in polymer informatics. We discuss some specific issues and our goals for improvement below.

- Manual conversion is necessary to transform the extracted material names into machine-readable formats such as Simplified Molecular Input Line Entry System (SMILES) strings to make the datasets informatics-ready. Despite the incorporation of LLMs into the data extraction pipeline, the parsing of chemical structures from figures remains a significant challenge, particularly in polymer-related studies. This is due largely to the fact that polymer structures are often exclusively presented in figures, which obstructs the direct extraction and conversion of polymer chemistry into machine-readable SMILES strings from the text. In the future, integration of large-scale computer vision models with LLMs to efficiently identify and extract polymer molecules depicted in figures will enable immediate use of the extracted data for training ML models without the need for additional manual processing.
- The intricate nature of scientific texts, particularly in introducing material names across different sections and using abbreviations, makes establishing correct relationships between entities mentioned in different paragraphs or even sentences a difficult task. Our current pipelines extract data that is described completely in a specific paragraph by looking for all the required named entities (i.e., 'material', 'property', 'value' and 'unit') to establish correct relationships. However, the properties of polymers often rely on further information, such as molecular weights, temperature, synthesis and processing conditions, and morphology. This additional data also needs to be extracted from multiple paragraphs, while ensuring the preservation of valid relationships. Using a specific example of Fig. 2d, where one of the extracted materials is simply labeled as a 'copolymer,' it becomes challenging to fully extract the actual name or chemistry of the material without inputting the full text of the article into the LLM or correctly identifying the first occurrence of the term using other means. However, feeding the entirety of text contents into larger context lengths of the latest LLMs, even if possible, is fundamentally inefficient and a squandering of computational resources. In addition, the outputs derived from conversational LLMs often exhibit inconsistency, necessitating manual effort for conversion into structured formats. Formulation of a robust chemical entity relationship extraction strategy that leverages both NER and LLM, could markedly augment the quality and application of the extracted data.
- Extraction of materials data present in tabular formats can further enhance the utility of comprehensive data extraction tasks. Another significant source of data often originates from the supplementary information published alongside articles. These documents are typically available in portable document format (PDF), which poses a challenge for parsing due to the lack of standardization in document creation[41]. While values with heightened scientific significance are usually mentioned in the main text, the presence of a substantial amount of extractable and relevant data in tabular format and supporting documents has the potential to greatly improve the performance of downstream data-hungry ML models. However, tables are frequently arbitrarily structured, necessitating meticulous filtering, classification, and pre-processing for correct relationship establishment[42,43].
- Extracting property data from literature represents a specific application of NLP in polymer research. Accurate predictions of step-by-step tasks and procedures, such as synthesis recipes, characterization data, measurement conditions, etc., could guide the development of superior polymers through inverse design and suggest specific conditions that researchers could maintain to produce a target material. Synthesis recipes, for example, present a unique challenge due to the need to extract a diverse set of information, including monomers, catalysts, temperature, reaction conditions, and more. Additionally, the chemical reactions must be predicted algorithmically, maintaining proper order of the procedure. Despite these challenges, the capacity of LLMs to comprehend complex procedures offers a promising avenue for systematically extracting such information.

Nevertheless, it is evident that the introduction of LLMs such as GPT-3.5 is a significant leap forward in the field of data extraction, particularly in complex domains like polymer literature. Future work may involve fine-tuning the model to concurrently handle searching, filtering, NER and data extraction tasks. As advancements in smaller open-source models persist, we anticipate the potential to substitute our entire workflow with a single, fine-tuned, open-source LLM. Such a system could democratize the process of extracting accurate data from literature. However, this transition would require comprehensive analysis and validation by the materials science community. Our focus will not only be on refining data extraction workflows but also on ensuring the availability of the extracted data for inspection through resources such as Polymer Scholar.

Conclusion

In conclusion, this study presents a framework for automated extraction of polymer property data from full-text scientific literature, utilizing a combination of NER-based MaterialsBERT and GPT-3.5 LLMs. The approach has demonstrated a significant improvement in data extraction capabilities, yielding 21 times more data than previously extracted from just the abstracts of journal articles. A comparative analysis of the performance and costs associated with both extraction models was also conducted. The GPT-3.5 and MaterialsBERT models achieved F_1 scores of 0.67 and 0.63 respectively for Tg, and 0.85 and 0.66 respectively for bandgap. While traditional NER models offer speed, the LLMs, although more costly, provide ease of use and require less manual effort, making them an attractive alternative. GPT-3.5 also shows a marked improvement in recognizing materials entities and correct entity relationships, particularly in the context of polymer composites and blends. However, both models encountered difficulties in extracting correct values from texts containing complex discussions about materials. The potential of LLMs lie in their ability to produce not just

structured outputs such as property data extraction, but also assistance with property predictions, material design and synthesis recipe recommendations.

Methods

The literature corpus

Our literature corpus consists of ~2.4 million documents downloaded from publishers including Elsevier, Wiley, Springer Nature, American Chemical Society, and Royal Society of Chemistry which covers articles published up to the year 2021. Only HTML and XML versions of the documents were processed in this work as formats such as PDF are difficult to parse[44]. Literature published before the year 2000 is often found in PDF format while XML and HTML versions are available from most publishers after 2000. The details of the workflow is discussed in ref. 32.

Full-text extraction

Plain text paragraphs embedded inside the `p`, `span`, or similar tags of the HTML and XML documents were extracted using the LXML Python package. Both the abstracts and the full texts available in the body of the documents were collected in the process. Subscripts and superscripts in the text were encoded by the underscore and the caret symbols, respectively. No additional pre-processing was performed since the LLMs are generally able to 'understand' chemical entities, units, and literature references.

Property-specific heuristic filter

The paragraphs extracted from full texts of the journal articles underwent a two-stage filtering process before being sent to the data-extraction pipeline. The heuristic filter assessed the text for the occurrence of the property name or any of its known variations, employing string-matching and dictionary lookup techniques. The property name variations were carefully curated through an extensive literature search.

NER filter

NER-based filtering was used to identify the paragraphs that contain data suitable for extraction. First, the named entities present in the paragraph are predicted using MaterialsBERT. The filter then selects the paragraphs that have (1) at least one of the material-related named entities, (i.e., "POLYMER", "MONOMER", "POLYMER_FAMILY", "ORGANIC", "INORGANIC") (2) the "PROPERTY_NAME" entity and (3) the "PROPERTY_VALUE" entity present in the text. If the text does not meet all three conditions, the paragraph fails the filter. This allows the filter to select paragraphs with data for all possible materials and properties known by MaterialsBERT and are suitable for extraction.

LlaMa model

An instruct-tuned 70B LlaMa-2-chat model was used in this work to extract data from texts. The architecture of the LlaMa model was obtained from the HuggingFace hub using the transformers Python package. The corresponding LlaMa 2 weights were requested and obtained from the official website of Meta AI. A 4-bit GPTQ quantized (group size 32, with act order) version of the model was run on four 235W 16GB Nvidia Quadro GP100 GPUs hosted in our in-house computing servers. For text generation parameters, the temperature was set to 0.001, top_p to 0.95, min_p to 0, frequency_penalty to 1.1, and top_k to 1. The maximum output length was automatically computed each time before inference, so the total number of tokens for the prompt and the generated output remains less than the context length of the model, i.e., 4096. The use of a conversational model did not involve incorporating history from prior interactions during text generation. Consequently, the model treated each prompt as a distinct text generation request.

GPT model

The GPT-3.5-turbo-0613 model hosted by OpenAI was used to extract data from text. The OpenAI Python package was used to access the OpenAI API. The temperature parameter was set to 0.001 for text generation by the model, with all other parameters remaining at their default values. Similar to the LlaMa 2 model, a history of previous interactions was not maintained between the API requests. The manually curated Tg and bandgap data were used as a shot to the LLM regardless of the property to be extracted. The extraction process from the full texts of the 716,000 paragraphs took approximately a month (respecting the guidelines for fair usage of the API server), incurring approximately 1200 US dollars. The API usage costs were calculated assuming 0.0015 and 0.0020 US dollars per one thousand prompt tokens and completion tokens respectively.

Data availability

The journal articles used to extract material property data were downloaded through licensing arrangements that the Georgia Institute of Technology has with Elsevier, Wiley, Royal Society of Chemistry, American Chemical Society, Springer Nature, Taylor & Francis, and the American Institute of Physics. The pre-trained language model MaterialsBERT is available in the HuggingFace hub at https://huggingface.co/pranav-s/MaterialsBERT. The material property data extracted in this work can be freely explored through https://polymerscholar.org.

Code availability

The code used in this work can be found at https://github.com/Ramprasad-Group/PromptDataExtraction.

Received: 25 April 2024; Accepted: 30 November 2024;
Published online: 19 December 2024

References

1. Doan Tran, H. et al. Machine-learning predictions of polymer properties with Polymer Genome. *J. Appl. Phys.* **128**, 171104 (2020).
2. Wu, C. et al. Dielectric polymers tolerant to electric field and temperature extremes: integration of phenomenology, informatics, and experimental validation. *ACS Appl. Mater. Interfaces* **13**, 53416–53424 (2021).
3. Kononova, O. et al. Opportunities and challenges of text mining in materials research. *iScience* **24**, 102155 (2021).
4. Foppiano, L. et al. Automatic extraction of materials and properties from superconductors scientific literature. *Sci. Technol. Adv. Mater.: Methods* **3**, 2153633 (2023).
5. Cheung, J. J. et al. PolyIE: a dataset of information extraction from polymer material scientific literature http://arxiv.org/abs/2311.07715 (2023).
6. Batra, R., Song, L. & Ramprasad, R. Emerging materials intelligence ecosystems propelled by machine learning. *Nat. Rev. Mater.* **6**, 655–678 (2021).
7. Oka, H., Yoshizawa, A., Shindo, H., Matsumoto, Y. & Ishii, M. Machine extraction of polymer data from tables using XML versions of scientific articles. *Sci. Technol. Adv. Mater.: Methods* **1**, 12–23 (2021).
8. Kononova, O. et al. Text-mined dataset of inorganic materials synthesis recipes. *Sci. Data* **6**, 203 (2019).
9. Choi, J. et al. Deep learning of electrochemical CO2 conversion literature reveals research trends and directions. *J. Mater. Chem. A* **11**, 17628–17643 (2023).
10. Beltagy, I., Lo, K. & Cohan, A. SciBERT: a pretrained language model for scientific text http://arxiv.org/abs/1903.10676 (2019).
11. Shetty, P. et al. A general-purpose material property data extraction pipeline from large polymer corpora using natural language processing. *npj Comput Mater.* **9**, 1–12 (2023).
12. Gu, Y. et al. Domain-specific language model pretraining for biomedical natural language processing. *ACM Trans. Comput. Healthc.* **3**, 1–23 (2022).
13. Chithrananda, S., Grand, G. & Ramsundar, B. ChemBERTa: large-scale self-supervised pretraining for molecular property prediction http://arxiv.org/abs/2010.09885 (2020).

14. Trewartha, A. et al. Quantifying the advantage of domain-specific pre-training on named entity recognition tasks in materials science. *Patterns* **3**, 100488 (2022).
15. Mahmood, A., Sonakshi, G. & Shetty, P. Polymer scholar https://polymerscholar.org (2024).
16. Olivetti, E. A. et al. Data-driven materials research enabled by natural language processing and information extraction. *Appl. Phys. Rev.* **7**, 041317 (2020).
17. Shetty, P. & Ramprasad, R. Machine-guided polymer knowledge extraction using natural language processing: the example of named entity normalization. *J. Chem. Inf. Model.* **61**, 5377–5385 (2021).
18. Touvron, H. et al. Llama 2: open foundation and fine-tuned chat models http://arxiv.org/abs/2307.09288 (2023).
19. Chowdhery, A. et al. PaLM: scaling language modeling with pathways http://arxiv.org/abs/2204.02311 (2022).
20. Lappin, S. Assessing the strengths and weaknesses of large language models. *J. Log. Lang. Inf.* https://doi.org/10.1007/s10849-023-09409-x (2023).
21. Choi, J. & Lee, B. Accelerating materials language processing with large language models. *Commun. Mater.* **5**, 1–11 (2024).
22. Wang, H., Li, J., Wu, H., Hovy, E. & Sun, Y. Pre-trained language models and their applications. *Engineering* **25**, 51–65 (2023).
23. Liu, Y., Cao, J., Liu, C., Ding, K. & Jin, L. Datasets for large language models: a comprehensive survey https://arxiv.org/abs/2402.18041v1 (2024).
24. Rozière, B. et al. Code Llama: open foundation models for code https://arxiv.org/abs/2308.12950v2 (2023).
25. Brown, T. B. et al. Language models are few-shot learners http://arxiv.org/abs/2005.14165 (2020).
26. Strubell, E., Ganesh, A. & McCallum, A. Energy and policy considerations for deep learning in NLP http://arxiv.org/abs/1906.02243 (2019).
27. Luccioni, A. S., Viguier, S. & Ligozat, A.-L. Estimating the carbon footprint of BLOOM, a 176B parameter language model http://arxiv.org/abs/2211.02001 (2022).
28. Dagdelen, J. et al. Structured information extraction from scientific text with large language models. *Nat. Commun.* **15**, 1418 (2024).
29. 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).
30. Polak, M. P. & Morgan, D. Extracting accurate materials data from research papers with conversational language models and prompt engineering. *Nat. Commun.* **15**, 1569 (2024).
31. Yang, S. J. et al. Accurate prediction of experimental band gaps from large language model-based data extraction http://arxiv.org/abs/2311.13778 (2023).
32. Shetty, P. & Ramprasad, R. Automated knowledge extraction from polymer literature using natural language processing. *iScience* **24**, 101922 (2021).
33. Meyer, J. G. et al. ChatGPT and large language models in academia: opportunities and challenges. *BioData Min.* **16**, 20 (2023).
34. Li, Y., Ramprasad, R. & Zhang, C. A simple but effective approach to improve structured language model output for information extraction http://arxiv.org/abs/2402.13364 (2024).
35. Dengel, A. et al. Qualitative research methods for large language models: conducting semi-structured interviews with ChatGPT and BARD on computer science education. *Informatics* **10**, 78 (2023).
36. Zhou, Y. et al. Large language models are human-level prompt engineers http://arxiv.org/abs/2211.01910 (2023).
37. Kim, S. I., Pyo, S. M., Kim, K. & Ree, M. Investigation of glass transition behaviours in aromatic poly(amic acid) precursors with various chain rigidities by oscillating differential scanning calorimetry. *Polymer* **39**, 6489–6500 (1998).
38. Khanlari, T., Bayat, Y. & Bayat, M. Synthesis, thermal stability and kinetic decomposition of triblock copolymer polypropylene glycol–poly glycidyl nitrate–polypropylene glycol (PPG–PGN–PPG). *Polym. Bull.* **77**, 5859–5878 (2020).
39. Shetty, P., Adeboye, A., Gupta, S., Zhang, C. & Ramprasad, R. Accelerating materials discovery for polymer solar cells: data-driven insights enabled by natural language processing. *Chem. Mater.* **36**, 7676–7689 (2024).
40. Chen, L., Kern, J., Lightstone, J. P. & Ramprasad, R. Data-assisted polymer retrosynthesis planning. *Appl. Phys. Rev.* **8**, 031405 (2021).
41. Zhu, M. & Cole, J. M. PDFDataExtractor: a tool for reading scientific text and interpreting metadata from the typeset literature in the portable document format. *J. Chem. Inf. Model.* **62**, 1633–1643 (2022).
42. Hira, K., Zaki, M., Sheth, D. & Anoop Krishnan, N. M. Reconstructing the materials tetrahedron: challenges in materials information extraction. *Digital Discov.* **3**, 1021–1037 (2024).
43. Gupta, T. et al. DiSCoMaT: distantly supervised composition extraction from tables in materials science articles. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (eds. Rogers, A., Boyd-Graber, J. & Okazaki, N.) 13465–13483 (Association for Computational Linguistics, Toronto, Canada, https://aclanthology.org/2023.acl-long.753 2023).
44. Smith, A., Bhat, V., Ai, Q. & Risko, C. Challenges in information-mining the materials literature: a case study and perspective. *Chem. Mater.* **34**, 4821–4827 (2022).

Acknowledgements

This work was supported by the Office of Naval Research through grants N00014-19-1-2103 and N00014-20-1-2175. Pranav Shetty was partially funded by a fellowship by JPMorgan Chase & Co. that helped to support this research. Any views or opinions expressed herein are solely those of the authors listed, and may differ from the views and opinions expressed by JPMorgan Chase & Co. or its affiliates.

Author contributions

Conceptualization: R.R. Data curation: A.A., P.S., and S.G. MaterialsBERT pipeline: P.S. LLM pipelines: A.M., P.S. Database design: A.M., S.G. Polymer Scholar: A.M., P.S. Visualization: A.M. Original draft: A.M., S.G. Review & editing: S.G., A.M., P.S., and R.R.

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/s43246-024-00708-9.

Correspondence and requests for materials should be addressed to Rampi Ramprasad.

Peer review information *Communications materials* thanks Byungju Lee, Zhiling Zheng and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Primary Handling Editors: Milica Todorović and Aldo Isidori. A peer review file is available.