JOM, Vol. 73, No. 11, 2021
https://doi.org/10.1007/s11837-021-04902-9

Challenges and Advances in Information Extraction from Scientific Literature: a Review

ZHI HONG,[1,3] LOGAN WARD,[2,4] KYLE CHARD,[1,2] BEN BLAISZIK,[1,2] and IAN FOSTER[1,2]

1.—University of Chicago, Chicago, IL, USA. 2.—Argonne National Laboratory, Lemont, IL, USA. 3.—e-mail: hongzhi@uchicago.edu. 4.—e-mail: lward@anl.gov

Scientific articles have long been the primary means of disseminating scientific discoveries. Over the centuries, valuable data and potentially groundbreaking insights have been collected and buried deep in the mountain of publications. In materials engineering, such data are spread across technical handbooks specification sheets, journal articles, and laboratory notebooks in myriad formats. Extracting information from papers on a large scale has been a tedious and time-consuming job to which few researchers have wanted to devote their limited time and effort, yet is an activity that is essential for modern data-driven design practices. However, in recent years, significant progress has been made by the computer science community on techniques for automated information extraction from free text. Yet, transformative application of these techniques to scientific literature remains elusive—due not to a lack of interest or effort but to technical and logistical challenges. Using the challenges in the materials science literature as a driving motivation, we review the gaps between state-of-the-art information extraction methods and the practical application of such methods to scientific texts, and offer a comprehensive overview of work that can be undertaken to close these gaps.

Key words: Information extraction, Text mining, Scientific data

INTRODUCTION

"There is an information overload in scientific literature",[1] according to *Nature*. A bibliometrics study shows that approximately 2.5 million new papers are published each year.[2] Such enormous volumes of new information are well beyond any human's ability to read, let alone digest and absorb. Inevitably, valuable knowledge remains buried deep in this publication mountain. As research techniques that require large quantities of scientific data gain popularity, automating the process of retrieving pertinent information from free (i.e., natural language and unstructured) texts to feed said techniques is expected to become crucial to many research domains.

The need for well-organized and thoroughly vetted data resources is particularly evident in materials engineering. On one level, identification of appropriate materials for new technologies is accomplished by searching through reams of certification and testing data—a process made better only by making more data available to engineers. The design of new materials is also intrinsically limited by available data. Data are foundational to devising and validating the core tools by which materials are understood and engineered: structure–property relationships. Many computational modeling tools, such as CALPHAD and phase-field models, are parameterized using large amounts of experimental data.[3] In recent years, the Materials Genome Initiative has further increased the prominence of data-driven materials research.[4] Overall, high-quality resource of materials data have been

(Received June 11, 2021; accepted September 1, 2021; published online October 5, 2021)

critical to the development of materials engineering and promise to become even more important in the future.

Despite the central need for high-quality data, building new databases remains a resource-intensive task. Curated data repositories are only available as collections published after significant effort (e.g., *Polymer Handbook*[5] and the *ASM Handbooks*), community-driven resources from specific research communities (e.g., the Crystallographic Open Database[6] and CALPHAD databases[3]), and web-accessible databases created by individual research groups (e.g., the Open Quantum Materials Database,[7] Polymer Genome,[8] and Materials Project[9]). Such databases are the ideal case for research data: vetted data presented in well-documented formats with predictable structure. However, the majority of important material property data remain strewn throughout decades of journal articles in fields spanning from fundamental materials physics to myriad specialized industrial applications. Consequently, maintaining the status quo with respect to materials data curation would result in the depths of historical data remaining uncatalogued and much new data slipping into obscurity after publication. New approaches are needed.

Automated information extraction (IE) techniques offer a route for accelerating the curation of data contained in scientific literature. IE, a process for the automated extraction of structured information, including entities and relations between entities, from text, is a highly developed topic in the natural language processing (NLP) community. Since the 1970s, a wide variety of techniques have been proposed to tackle this problem.[10] Traditional methods often require considerable target-domain knowledge and the development of sets of domain-specific rules defined manually from experience. More recently, with rapid growth in both data volumes and computing power, statistical models using machine learning (ML) or deep learning (DL) have taken center stage.

Adoption of automated IE methods in science and engineering varies greatly across disciplines. For example, use in the life sciences is advanced due to work on online biomedical bibliographic systems, biomedical knowledge representation, and text mining dating back to the 1960s,[11–14] while work in some other fields has barely started. Automated IE in materials science and engineering is only in its initial stages, with promising work in the ceramics and polymers community[15–17] and significant opportunities in other types of materials.

In this paper, we present an overview of the scientific IE (SciIE) process with a particular emphasis on the challenges and opportunities for materials science and engineering. Our goal is to examine the specific challenges and relevant advances in applying state-of-the-art methods developed by computer scientists to real-world materials SciIE problems. Finally, we identify important open research areas that should be explored to advance the application of SciIE.

SCIENTIFIC INFORMATION EXTRACTION WORKFLOW

Before diving into the specific barriers faced by SciIE, we provide an overview of the common steps involved in a SciIE pipeline (Fig. 1): data preprocessing, curation and annotation, and learning.

The first step (preprocessing) is to break down scientific articles into chunks of clean text for later steps. In addition to text in the body of an article, scientific documents may also include figures, tables, and publisher embellishments (e.g., logos, running titles, and page numbers). The first step in preprocessing is thus to *parse the document* and extract the body text. This is particularly complicated in science due to the complex formats of scientific articles, which we discuss in Challenge 1: The Computer-(Un)friendly Format of Scientific Texts section. While tables and figures may also contain valuable facts, extracting information from them relies on a completely different set of technologies, such as computer vision, which lie beyond the scope of this review. Interested readers may refer to[19–22] for relevant research. After document parsing, texts are then split into smaller units. This step is called *tokenization*. Sentence tokenization splits passages into sentences, which is the input format expected for many later steps. Word tokenization further splits sentences into tokens to reduce the entropy of the vocabulary (e.g., "is," "does," "isn't," and "doesn't" can be tokenized into three tokens: "is," "does," and "n't.")

Once text has been extracted and cleaned, the next step is to create the tools that allow for so-called intelligent processing of the language. The entire process of extracting information from text is accomplished by a pipeline of complementary tools rather than a single program that produces data directly from tokenized text. The steps in these pipelines often include (Fig. 2):

1. *Vocabulary generation* to assign a vector to each unique token in the corpus. The vectors are referred to as *word embeddings* and are a key prerequisite for other NLP tasks. Meanings are often inferred by measuring similarity in the contexts in which words appear, or the substructure of words. Embeddings can even be studied to predict materials properties.[23]
2. *Text classification* to assign a label or score to an entire block of text. For example, they could determine whether a block of text is an abstract or whether it contains the desired data.
3. *Named entity recognition* (NER) to classify whether a word or phrase belongs to a specific category. Categories may be broad (e.g., noun) or specific (e.g., place name or polymer).
4. *Relationship extraction* to produce pairs of

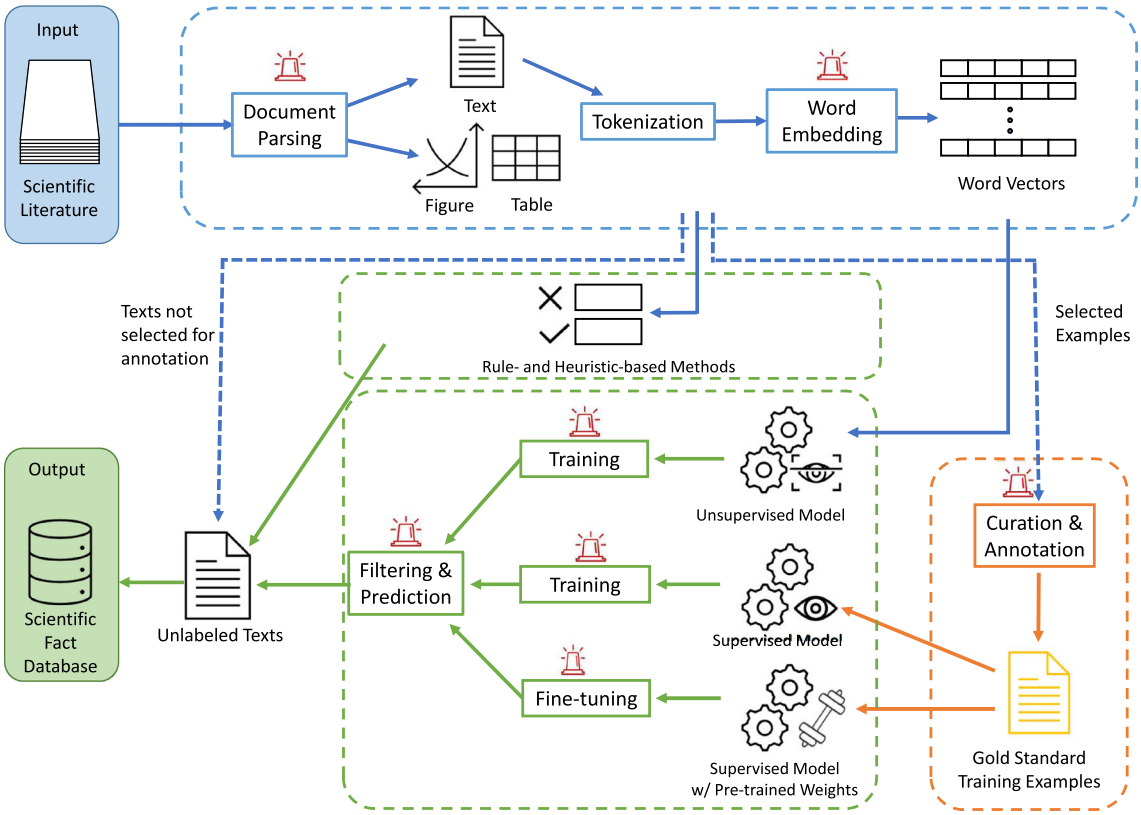
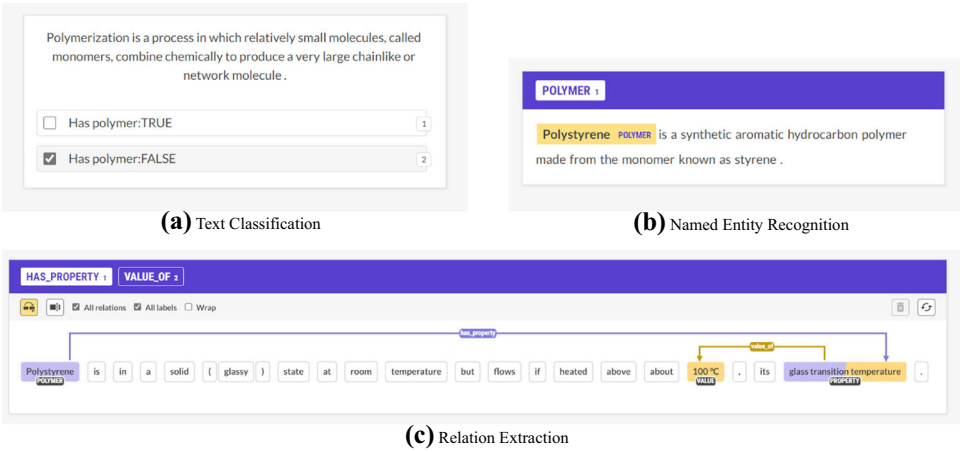


Fig. 1. The three steps in a scientific information extraction (SciIE) pipeline: preprocessing (blue), data curation (orange), and learning (green). Components marked with a red alarm symbol are particularly challenging.



(a) Text Classification **(b)** Named Entity Recognition **(c)** Relation Extraction

Fig. 2. Example tasks in SciIE (visualized with Prodi.gy[18]): (a) classifying sentences based on whether they mention polymers, (b) recognizing named entities such as polymers, and (c) identifying relations between named entities (e.g., polymers, properties, and property values).

named entities that are connected by a certain relationship. A common example of IE in materials science and engineering is to associate words that are materials with those that are property names. Complex relationships can then be built from multiple pair relationships, such as the material "iron" *has property* "density" *with value* "7.9 g/cc."

In principle, it should be possible to define rules manually to perform each of the tasks just listed. For example, units can be identified by matching words that optionally begin with "k," "m," or "M" and end with a character from a known list (e.g., "m," "s," or "Ω"). The complexity of human language, however, is too high to allow the enumeration of a complete set of rules except in trivial cases

(e.g., units). Rather, the modern approach is to use ML techniques to *learn* such rules automatically from many examples.

The second major step in performing IE is *curating and annotating* enough data to train ML models for each SciIE subtask. Tokenized texts are first selected and annotated to form a so-called *gold-standard* training set for supervised models. Labels could be per example (sentence) or per token, depending on the task. A sentence could be given the label "True" or "False" if we simply want to classify whether it mentions any polymer. If we want to find out which polymers are mentioned in a sentence, then each word will need a separate label indicating whether it is a polymer or not. The annotation process is usually costly, and time-consuming, and is especially difficult for scientific data due to the expertise required, the limited bandwidth of the people who do have the expertise (Challenge 2: The Need for (and Lack of) Training Data section), and the rarity of desired data across the scientific literature (Challenge 3: The Sparsity of Information of Interest in Literature section).

The final step in the pipeline is *model learning*. ML models for NLP are classified into two categories: supervised and unsupervised models. Most models used in IE are supervised, meaning labels are required to accompany the training data. For example, common NER approaches use the embeddings and other features of a word (e.g., length and whether it contains digits) and those of its context (i.e., words ahead of or behind it) as input into a simple ML model such as a decision tree or support vector machine (SVM) to predict whether that word belongs to a category. State-of-the-art techniques use neural networks that automatically account for the context of a word (e.g., recurrent or convolutional networks) and are flexible enough to express the exquisitely complicated model forms required for expressing language.[24,25]

Modern NLP research has focused on reducing the amount of data required to train supervised learning models. Fine-tuning methods take pretrained data from a previous NLP problem and (re)train part of the model on annotated data specific to the task at hand. Google's Bidirectional Encoder Representations from Transformers (BERT) language model, which is trained on the BooksCorpus[26] (800M words) and English Wikipedia (2500M words), when fine-tuned with just 2432 relevant paper abstracts, achieved an F-1 score of 74.7% on extracting and classifying chemical–protein interactions (e.g., "Regulator," "Agonist," "Antagonist," etc.) from the CHEMPROT corpus.[27] The F-1 score, or F-score, is the harmonic mean of a model's precision (the fraction of correctly predicted positive examples among all the examples that the model predicted as positive) and recall (the fraction of correctly predicted positive examples among all the positive examples in the dataset). It is often used to measure a model's prediction accuracy. Unsupervised models such as open information extraction (openIE) systems have also been gaining traction because they do not require labeled data. Such systems have been demonstrated to outperform other traditional IE methods in multiple studies.[28–30] However promising, the use of these small-dataset learning techniques for scientific tasks is complicated by the esoteric language often used in science as well as the variation in languages used to describe the same concept (Challenge 4: The Difficulties of Applying Models Trained on General Corpora to Scientific Text section).

Finally, the difficulty in accurately extracting information combined with the high standards for success mandate that NLP tools must be used with care. The key challenge in applying models is often the sparsity of the desired information in literature, which can lead to many false positives when the pipeline is run on too many irrelevant papers. It is difficult to know ahead of time which papers will contain the target data and where to find them in the paper. We detail approaches for preselecting texts and postprocessing outputs to improve extracted data in Challenge 3: The Sparsity of Information of Interest in Literature section.

In the following sections, we carefully examine the major challenges faced by SciIE, including the scarcity of labeled data, the sparsity of interested information in publications, and the difficulties of applying ML or DL models trained on general corpora to scientific texts. Relevant advances are discussed as potential solutions to, or methods for alleviating, these challenges.

CHALLENGE 1: THE COMPUTER-(UN)FRIENDLY FORMAT OF SCIENTIFIC TEXTS

The data needed to train or use NLP models are texts, ideally clean, well-formed texts that can be easily ingested by computers (and humans). Many generic NLP datasets exist, such as the CoNLL-2003 dataset for training NER models[31] and the TACRED dataset for relation extraction models.[32] One common feature of these NLP datasets is that they are distributed as plain text and have well-defined, homogeneous internal formats. For example, in the CoNLL dataset, each word is placed on a separate line with an empty line at the end of each sentence, and on each line, the word is followed by three tags: a part-of-speech tag, a syntactic tag, and a named entity tag. Such structured datasets can be fed into a model with only a few lines of code and without any extra preprocessing. Feeding scientific literature into models, unfortunately, is quite a different story.

Useful documents in materials engineering are stored in a few different kinds of documents, each presenting different processing challenges. Specification sheets and handbooks often express data in tabular formats instead of natural language and are

best treated with special-purpose software tailored to their specific formats given the predictable form in which data are expressed. Information from journal articles, conference proceedings, and technical reports are held in text in a variety of document formats. Older articles are often only available as scanned images, whereas more modern articles are expressed in a variety of digital formats. As we detail here, this multitude of formats for engineering text presents a major barrier to extracting knowledge.

Historical Relic: Portable Document Format

Most papers are shared digitally in the portable document format (PDF). To strictly maintain document typesetting, PDF stores a fixed layout of the content, including texts, figures, and tables. However, it does not record structural information (Fig. 3). While humans can tell whether a number is part of the body, a page number, or a line number based on visual clues, it is difficult to do so programmatically. In the worst case (common for papers published prior to 2000), a PDF file may be a scanned image of the printed copy. In such cases, optical character recognition (OCR) must be performed to recognize the letters and numbers in the paper, prior to proceeding with the rest of the pipeline.

The complex layouts of scientific papers makes it difficult to extract the actual narrative from PDF files. The widely adopted double-column format can confuse automated methods for identifying the flow of text blocks. Pages are often decorated with publisher names, running titles, page numbers, etc. General-purpose toolkits such as PDF2Text[33] are not equipped to handle such complexities and thus often produce outputs in which useful body text is mixed with typesetting embellishments.

Systems specifically designed to extract from scientific articles commonly employ a layout-aware (e.g., two-column versus one-column) approach. Both manually defined rules and statistical models may be used to estimate the structural information missing in PDFs. For example, LA-PDFText[34] detects words belonging to the same block based on their font, height, and horizontal and vertical distance from the nearest words. Then, a rule-based classifier categorizes each block into sections (e.g., Abstract, Introduction, Methods, and Discussion), and finally texts classified into the same sections are stitched together to form the final clean output. Although experimental evaluations have shown that LA-PDFText can reach an F-1 score of 91% in identifying and classifying text in scientific articles, it is still not perfect and requires the manual definition of rules according to the typesetting style, which varies from journal to journal or even over time for the same journal. Thus, applying LA-PDFText to a large collection of heterogeneous PDF papers is impractical for many purposes.

Rising Star: Markup Language Formats

With the development of the Internet and web-based technologies, scientific papers are increasingly available in hypertext markup language (HTML) or extensible markup language (XML) formats. HTML is by far the most popular format on the web since it adapts to the browser screen regardless of its form factor. Most importantly, HTML tags unambiguously identify the different elements on a page (e.g., <*img*> for images, <*table*> for tables, and <*p*> for paragraphs) (Fig. 3). These tags simplify the process of identifying sections in a paper. Body texts can be extracted without any of the "guesswork" involved with other formats.

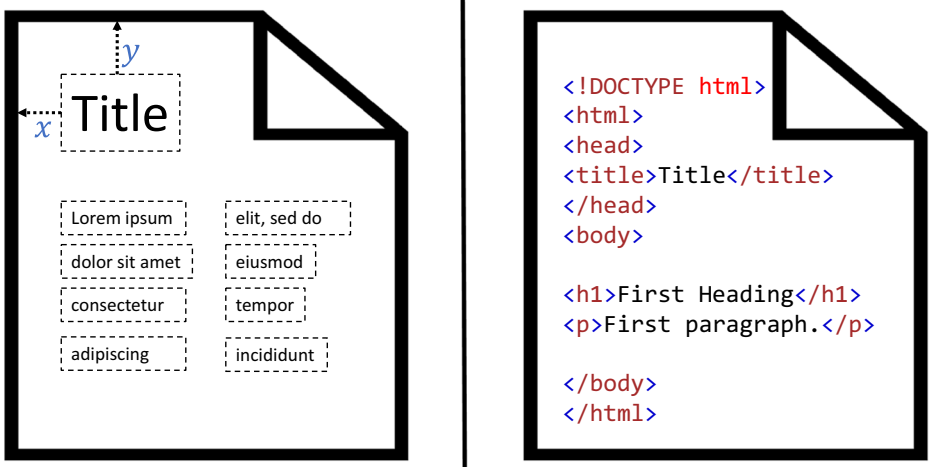


Fig. 3. PDF (left) files are designed to capture page layout, while HTML (right) files are designed to save logical structure. It is thus much easier to identify which parts of a file are the desired body texts with the help of HTML tags.

HTML-formatted papers are commonly used as the source in many studies. BigGrams is a semisupervised IE system designed to work with HTML inputs.[35] A series of research focusing on human–machine hybrid IE pipelines uses HTML papers from the journal *Macromolecules* to extract glass-transition temperatures of polymers[36,37] and create a training dataset for ML models.[38] Another study uses HTML papers from Elsevier journals to develop an NER model to extract and analyze datasets used in sociology studies.[39]

CHALLENGE 2: THE NEED FOR (AND LACK OF) TRAINING DATA

Modern artificial intelligence (AI) methods derive their "intelligence" from the data on which they are trained. No model architecture, regardless of its sophistication, can do better than random guessing without training data. Moreover, having the data itself is often not sufficient. Many models are created for predictive purposes, i.e., to respond to a query: "Given **x**, predict **y**." Having the data (**x**) alone is of limited utility without the corresponding labels (**y**). Such models are called *supervised* models, and modern ML tools can require thousands of examples of (x, y) pairs.

The dearth of training data in materials engineering can be simply attributed to IE in materials being in its beginnings, though there are ways to rapidly accelerate the development of training sets. In this section, we first examine methods for curation of labeled training data, discussing the difficulties involved and presenting progress towards solving the difficulties. Then, we discuss models that can train on data without labels (*unsupervised* models) and analyze their strengths and limitations. Figure 4 shows a summary of the pros and cons of different solutions to the data collection and annotation problem.

Harvesting Existing Data

In many scientific fields, there are efforts to compile useful databases from publications. For example, the Polymer Property Predictor and Database includes proprieties such as the polymer interaction parameter (χ) and glass-transition temperature (T_g) extracted semiautomatically from literature.[40] In sociology, the Inter-university Consortium for Political and Social Research (ICPSR) maintains a database of sociology datasets and related publications via manual curation.[49] These databases usually store the extracted data in a structured format, but they often have an identifier, e.g., a digital object identifier (DOI), pointing to the source of the each entry, with which discovering the source text is feasible programmatically. Combining the curated data in the databases and their source text provides a good gold-standard dataset for training ML or DL models to extract similar data automatically from literature.

Another potential source of existing data comes from outside of academia. Scientific publications have long been ignored by NLP research in industry. Yet, in recent years, some companies have developed models and datasets for scientific literature. For example, AllenAI developed SciBert[50] and published the data used to train the model, which includes gold-standard annotations on papers from semanticscholar.org for many common NLP tasks, such as labeled entities in sentences to develop NER models, syntactic tags for words in texts to train syntax parsing models, and classification labels with accompanying sentences for text classification.

Crowdsourcing

Traditionally, datasets have been curated via slow and expensive manual processes. For example, from 1989 to 1992, a team at the University of Pennsylvania spent three years annotating a corpus of over 4.5 million English words with part-of-speech (POS) and sentence skeletal parsing information.[51] The resulting Penn Treebank dataset is still widely used to train NLP models for POS tagging and sentence parsing today, almost 30 years later, because few can afford the high cost of building a corpus of such scale.

With the increasing penetration rate of Internet-connected devices among the population, crowdsourcing has become a more viable approach to a number of labor-intensive tasks, corpus annotation included. Online services such as Amazon Mechanical Turk (AMT) and CrowdFlower (CF) offer platforms to post crowdsourcing jobs and to engage the public to contribute for monetary rewards. There are also tools that help streamline the crowdsourcing process. The GATE crowdsourcing plugin automates the mapping of documents to crowdsourcing units and generates user interfaces for common NLP crowdsourcing tasks.[52] Crowdsourcing has been shown to be effective at solving the problem of training data annotation. Granted, the quality of annotations produced by an untrained crowd could vary, so it is common to assign the same tasks to multiple workers and apply a majority voting system to improve annotation quality. For example, 145 AMT participants annotated a corpus of 593 biological abstracts for disease mentions, achieving an overall F-1 score of 87.2% with a cost of $0.066 per abstract per worker.[53] Another AMT project adapted the reCAPTCHA idea[41] to crowdsource the digitization of satellite images and demonstrated that an untrained population could achieve an accuracy within 10% of that of geoinformatics experts.[42] MIT and Amazon conducted a research on 10 widely used ML datasets including text, image, and audio datasets, and found that the labeling error rate ranges from 0.15% to 10.12% with an average error rate of 3.4%.[54]

Crowdsourcing, nevertheless, is not without its drawbacks, and applying it to annotate a scientific

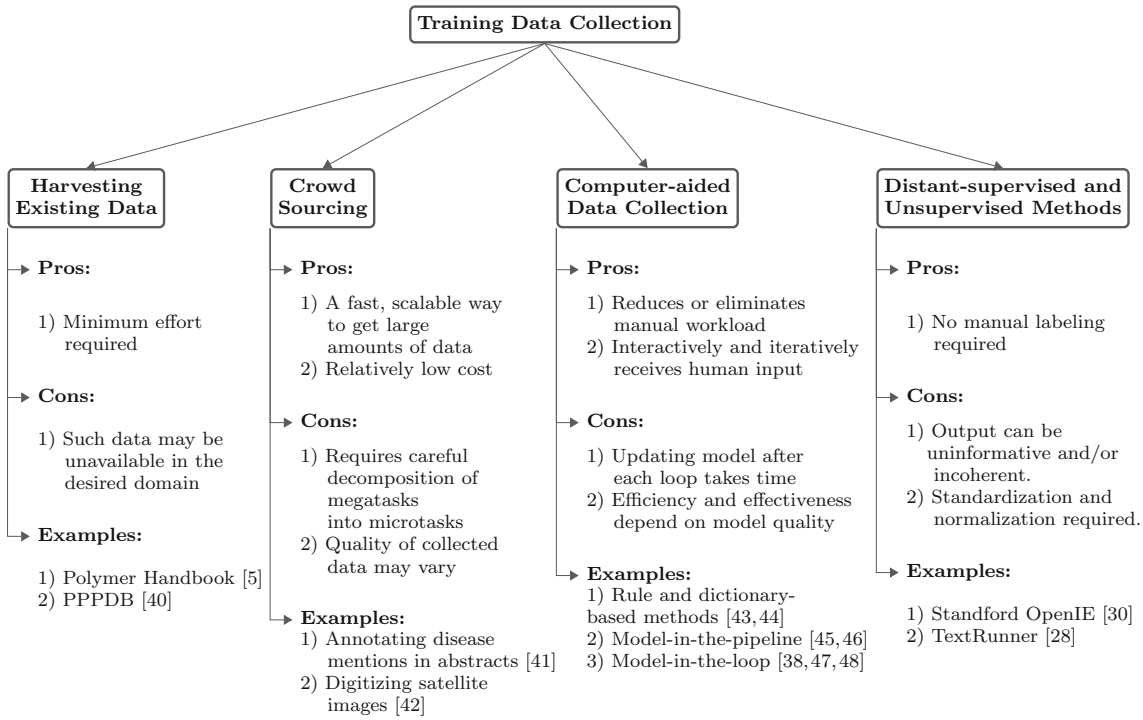


Fig. 4. Comparison of data collection and annotation solutions: Harvesting existing data takes the least effort if such data are available; crowdsourcing is the most straightforward, but requires careful task decomposition; computer-aided methods reduce manual workload, but quality may vary; distant-supervised and unsupervised methods do not require manual labeling but may produce ambiguous or incoherent labels.

corpus requires careful planning. Crowdsourcing is best suited to tasks that are *microtasks*, i.e., tasks that are both relatively simple and modest in scale. In contrast, scientific research often aims to solve *megatasks*, such as building a database of all material properties from all published literature. Even extracting all polymer proprieties mentioned in one paper is a megatask for crowdsourcing. One solution to this mismatch is by crowdsourcing only to people with enough scientific background. One such effort engaged undergraduate students to extract the Flory–Huggins (χ) parameter of polymer blends from the journal *Macromolecules*.[55] CHEMDNER is another chemistry corpus that contains 10,000 PubMed abstracts annotated by about 50 scientists from around 30 institutions.[56] The Synthesis Project from Massachusetts Institute of Technology (MIT) offers annotations for 230 material science papers.[57] The well-known *Polymer Handbook* listed 96 contributors from universities and research institutions worldwide.[5] The National Institute of Standards and Technology (NIST) Thermodynamic Research Center (TRC) maintains a large database of available thermophysical property data extracted from articles manually selected for relevant content.[58]

Crowdsourcing to people with relevant domain expertise seems like a sure way to guarantee the quality of the results, but it is not without its own shortcomings. In addition to being drastically more expensive, it also restricts the eligible "crowd" to a small group, and such people typically have limited bandwidth for such tasks. In our experience, it took three materials scientists over 2 months to annotate just 150 paragraphs for glass-transition temperatures (T_g) of polymers, since it was difficult to find time to work on it with their busy schedules.

To take full advantage of crowdsourcing, megatasks must be divided into smaller and simpler microtasks. One way of doing so is to create a *game with a purpose* (GWAP).[59] Carefully designed GWAPs have been used to address many complex scientific problems that would otherwise be incomprehensible to an untrained person. In biology, the multiple gene sequence alignment problem has been presented as a color-matching game to crowdsourced workers.[60] GWAPs have also been used in annotating complex language resources.[61] When designing GWAPs for complex scientific text annotation, the key is to decompose the megatask. For example, instead of assigning a worker a full paper, one assigns them a paragraph or even a sentence; instead of asking them to label every material property, one assigns only the tasks pertaining to a single property, so that they have less information to remember. Partitioning megatasks into microtasks also enables the mapping of microtasks to different levels in a GWAP based on difficulty, giving the workers a sense of achievement as they

increase their skill, which can contribute to keeping them engaged.

Computer-Aided Training Data Collection

Rule-based and dictionary-based methods. Annotating a corpus does not have to be done entirely by humans. For certain tasks, the help of machines can greatly reduce the manual effort required to annotate a corpus. In many scientific domains, systematic nomenclatures and unique identifiers are commonly used. In chemistry, there is the International Union of Pure and Applied Chemistry (IUPAC) nomenclature[62] and Chemical Abstracts Service (CAS) registry numbers,[63] both of which are used to refer to chemicals in literature. In biology, standardized nomenclature has been defined for human gene mutations.[44] Rule-based approaches (e.g., regular expressions) are a great fit to automatically annotate their mentions in texts.[43,44] Rules can also be constructed with formal grammars,[43] which works best when there are commonly adopted languages in a domain for presenting certain types of information in literature.

Rule-based methods are not sufficient if the information we want to annotate presents no obvious pattern. However, there are dictionaries available in many domains. For example, DBpedia—a structured database built from Wikipedia that includes categories such as chemical compound, mineral, gene, and protein[64]—can be leveraged to label such entities automatically in free text. Granted, rules and dictionaries may not be 100% accurate or comprehensive, so manual review is often necessary, but reviewing is still much more efficient than manual labeling.

Model-in-the-pipeline methods: Researchers have recently explored the use of ML/DL models to create datasets that are then used to train bigger and better models. Their workflows begin with the manual annotation of a small subset of the corpus. The annotated texts are used to train a model, which may be a simplified or a smaller version of the full model to be trained on the fully annotated corpus. The trained model is then applied to the rest of the unlabeled corpus. For each sample in the unlabeled corpus, the model's prediction is used to compute a metric that is then used to decide what human annotators should do with that sample.[45] For a classification model, for instance, the metric could be the classifier output probabilities. Samples with probabilities below a threshold should undergo a thorough manual annotation process, while those with higher probabilities could be assigned to a different group of (perhaps less experienced) annotators for a quick review.[46]

Model-in-the-loop methods. Model-in-the-pipeline methods for corpus annotation have a major drawback: the quality of the model in the pipeline is solely dependent on the choice of the initial training examples. If the initially selected examples are not representative of the overall distribution in the corpus, which is not unlikely for large corpora with hundreds of thousands of articles, the model will be less effective, and the human annotators would spend precious time correcting the same mistake repeatedly. In contrast, *model-in-the-loop* methods avoid this problem by adding a feedback loop between humans and machine.

With the feedback loop, the initial selection of training examples becomes less significant. The annotation process is done in batches of m examples. The model runs prediction on a batch of examples, and after human review of the results, the m gold standards are added to the training set and the model is retrained with the newly added data. This process is sometimes also called *active learning*, since the model actively requests new inputs from humans in order to update itself.[65] Model-in-the-loop methods have been used to create training sets for detecting mentions of polymers[38] and drug-like molecules,[47] extract health determinants,[48] and detect named entities in financial data.[66]

Distant-Supervised and Unsupervised Methods

While the aforementioned methods focus on reducing the cost of manual annotation through crowdsourcing, others seek to eliminate manual labor from the annotation process entirely. *Distant-supervised methods* still have supervised models at the core, but the labels are automatically generated from an external source of knowledge such as Wikipedia or Freebase. If two entities in the same sentence match to an entry in the knowledge base, then the sentence is labeled as having that relation.[67]

Distant-supervised methods nevertheless suffer from the noisy label problem. For example, if say "(Bill Gates, Microsoft)" is described via a "FounderOf" relation in an external knowledge base, then the sentence "Bill Gates steps down from Microsoft board to focus on philanthropy" might be mislabeled as expressing that relation since it mentions both entities. *Multi-instance learning* (MIL) is designed to mitigate this problem. Instead of assigning a label to each training example, examples that would get the same labels are put in a bag, and labels are assigned to the bag rather than the examples. The intuition is that *some*, but not necessarily *all*, examples in the bag have that label.[68] *Multi-instance multilabel learning* takes this process one step further, assigning multiple possible labels to the same bag.[69] However, neither approach fully addresses the noisy label problem. In both methods, labels are hard-coded to the bags and are immutable after the distant-supervised labeling process.

More recently, distant-supervised learning with soft labels has been used to correct labels

dynamically, instead of trying only to minimize the negative impact of mislabeled examples on model training.[70] Empirical data show that distant-supervised labeling assigns correct labels to a majority (94.4%) of the examples in a benchmark.[71] With that assumption, the soft label of an example is updated based on its syntax pattern similarity to the examples both in the same bag and in other bags.

Distant-supervised learning has been applied to SciIE tasks. In one materials NLP task, the application of distant supervision to a corpus of 3400 publicly accessible articles on ScienceDirect resulted in the automatic labeling of about 5000 sentences as expressing process–structure or structure–property relations. The sentences were then used to train several models that aim to generate processing–structure–property–performance (PSPP) design charts for desired properties from text.[72] In biology, a dataset consisting of 1728 examples produced from PubMed abstracts by using the Protein Data Bank (PDB) as the distant supervision knowledge base was used to train a model for mining protein–residue associations from literature.[73] Another effort produced 450 examples of intrasentence gene–drug relations from literature using the Gene Drug Knowledge Database (GDKD) as the knowledge base.[74]

A notable deficiency of distant-supervised learning is that relations not present in the knowledge base cannot be recognized, no matter how prevalent they are in the corpus. Open information extraction (openIE) systems such as TextRunner[28] and the Wikipedia-based Open Extractor (WOE)[29] represent one approach to overcoming this deficiency. These systems are not bound by a predefined vocabulary because they extract entities *and* the relation term, all from the text based on syntactic dependency. OpenIE is especially advantageous in annotating scientific articles describing cutting-edge or rapidly evolving research, where structured knowledge bases have not been compiled. In a study on a collection of 12,007 abstracts, Stanford OpenIE[30] extracted 3116 relations, 65% of which were missed by other extraction tools.[75] This result shows that incorporating OpenIE systems into the automated annotation pipeline can greatly expand the scope of the resulting training set.

OpenIE can help with the noisy label problem that plagues distant supervision. Because OpenIE only gets the relation term from text, we can be more confident that the relation extracted is actually expressed in that text. In the Bill Gates example above, it is impossible for OpenIE to output a "(Bill Gates, founder, Microsoft)" relation since "founder" does not exist in that sentence. Therefore, OpenIE can be used to remove, and potentially correct, the noisy labels derived from knowledge bases in distant-supervised learning.

OpenIE systems also have shortcomings. They may extract words too broad to be meaningful as relations, such as "is", "has", and "got", rather than the actual relations "is the author of," "has a population of," and "got funding from." Incoherent words may be extracted as a relation, such as "was central torpedo" from the sentence "The Mark 14 was central to the torpedo scandal of the fleet."[76] Furthermore, the extracted relations can be difficult for downstream models to use, such as when the semantically equivalent phrases "melting point of" and "melting point is" are captured as distinct relations. One solution to such problems is to combine domain-independent OpenIE with domain-specific knowledge. Domain-specific relation mapping rules, class recognizers, and SciIE models can be used to cluster and categorize relations detected by OpenIE systems.[77–79]

CHALLENGE 3: THE SPARSITY OF INFORMATION OF INTEREST IN LITERATURE

SciIE must address a challenging range of scales. At one extreme, according to a bibliographical study on global publications from 2000 to 2018 by the National Science Foundation (NSF),[80] scientific literature published in that time span encompasses some 35.5M articles, which ultimately we would like to analyze in their entirety. At the other extreme, the literature associated with individual disciplines and subdisciplines, each characterized by distinct vocabularies and conventions for communicating information, are much smaller. For example, the same study found that the materials science literature encompasses 1.1M articles during that period: 3.1% of the total. A particular subdiscipline, such as polymer science, accounts for just a portion of those 1.1M articles, of which a yet smaller subset contain information relevant to any specific question.

Thus, even if we can identify the relevant publications accurately and efficiently, the sparsity of interesting information in text can be yet another obstacle to efficacious IE from scientific literature.[81,82] One study on extracting glass-transition temperatures (T_g) showed that only 64 (0.67%) of 9518 sentences in 31 papers from *Macromolecules* contain both a polymer and its T_g value.[83] The remaining 99.33% of sentences are just noise for this task. Such imbalance is rare in standard NLP datasets that most state-of-the-art models are designed for and trained on, but will be common when extracting data from other materials engineering literature. For comparison, the widely used SemEval 2010 Task 8 relation extraction dataset has only 17.63% and 16.71% of "Other" sentences (i.e., not belonging to known relations in the dataset) in the training and test set, respectively.[84] Extracting such sparse information will be difficult for ML/DL models. The high percentage of noise in texts will lead to more false positives and thus dilute the extraction results. To avoid this problem, it is

advisable to apply a filtering step during preprocessing to remove as much noise as possible before feeding the texts into an IE model.

In this section, we provide a review of the techniques developed to filter noisy texts. We start with traditional intuitive heuristic-based methods and expand to modern statistical model-based methods (Fig. 5).

Heuristic-Based Filtering

Article structure-based filtering. Scientific papers usually follow a structured format made up of sections and subsections. Sections such as "Introduction," "Related Work," and "Experimental Results" are widely used in many domains. Some publishers even have standardized section headings that every manuscript must have. With some background knowledge, we can tell with confidence that some sections will not have the information we want to extract: the "References" section is probably not the best place to look if we want to extract the synthesis process of a novel polymer. Therefore, a filter can be applied to remove texts in irrelevant sections to reduce potential noise in subsequent IE tasks.

Papers in markup languages (HTML or XML) can be easily dissected into sections because they have rich metadata describing the document structure. PDF files, however, cannot (Historical Relic: Portable Document Format section). A multipass sieve approach has been proposed to classify text in PDF files into proper sections. It demonstrated better performance (measured by F-scores) than many ML classifiers, including SVM, naive Bayes, J48, and logistic regression.[85] Using this method, structure-based filtering can also be applied to articles in PDF format.

Sentence-level filtering. Section filtering alone is not always sufficient, since noisy text exists in every section, even potentially useful ones. Sentence filtering offers more fine-grained control. With manually defined rules or patterns, it can achieve high precision, but often at the cost of recall due to the small number of rules defined.[86,87] Statistical methods can automatically learn a large number of rules and thus increase recall, but precision is lost as a result. Many efforts have been made to improve the quality of rule-based filters. Dropping rules whose keywords triggers more false positives than true positives, limiting the maximum length of sentences that can be matched, and tuning the threshold of what is considered as a match are simple yet effective tricks that enabled a rule-based filter to outperform a more sophisticated method based on minimum description length from information theory.[88]

Filtering with Statistical Models

Unlike heuristic-based filtering, statistical models do not rely on explicit rules. They learn what is (and is not) interesting information from the context based on word embeddings. Traditional classifiers are the most intuitive choice for this task, and more sophisticated methods such as subjectivity analysis and data programming have also been used recently.

Classification models. Text fragment filtering is a type of classification task, so intuitively ML classifiers have been applied to this task. Popular traditional classifiers include decision tree, SVM, *k*-nearest neighbors, naive Bayes, importance value index (IVI), and C4.5.[89,90] Over the years, amendments have been made to such algorithms to improve their performance, specifically on text classification, and each method has its own advantages. The naive Bayes classifier has been shown to achieve the best performance without any feature selection, whereas when features (words) were selected by its capacity to independently characterize a class, the IVI classifier came out top.[91] SVM works well on two-class classification problems, and decision trees do not require independent features to be effective.[92] In short, no single classifier has significant advantages over all others, and the choice of which classifier to use should be made based on the features used and the task at hand.

Subjectivity analysis. Another common source of error for IE systems is subjective language such as opinions, arguments, and speculations, which should be excluded when aiming to extract scientific facts. Removing subjective sentences with a naive Bayes classifier increased the precision of an IE system by 10% while losing less than 2% of recall.[93] Another series of study demonstrated that IE and subjectivity analysis are mutually beneficial because a subjectivity classifier can be bootstrapped from clues learnt by IE techniques, and used to improve the precision of IE systems.[93–96]

Data programming. This alternative ML-based approach to filtering text fragments for IE is used in Snorkel, a system for rapid collection of training data.[97,98] Snorkel is a weakly supervised method since it does not require any manual labeling. Instead, users write *labeling functions* (LFs). An LF assigns labels to inputs. It can be based on arbitrary rules or heuristics. Different LFs can have unknown accuracy and may not be independent of each other. For each input, Snorkel aggregates the labels from all LFs, learning about the performance of the different LFs in the process, and eventually outputs a high-quality label for the input.

The ensemble labeling towards scientific information extraction (ELSIE) system builds on Snorkel. Its goal is not to extract relations from sentences, but to determine whether a particular sentence exists in a text fragment. Instead of having each LF independently classify a sentence and then denoising their outputs, it groups its LFs into buckets, with each bucket responsible for determining whether a part of the target relation exists in the sentence. All buckets collectively aim to determine

whether the target relation is expressed in the text. ELSIE can successfully filter sentences with 94% recall and 87% F-1.[83]

CHALLENGE 4: THE DIFFICULTIES OF APPLYING MODELS TRAINED ON GENERAL CORPORA TO SCIENTIFIC TEXT

It is standard practice in most applications of NLP to reuse ML models trained on plentifully available text (e.g., websites or newspapers), with only *retraining* performed to adapt the models to new domains. For general NLP, there are many datasets (e.g., Penn Treebank,[51] CoNLL,[99] and OntoNotes[100]), pretrained word vectors (e.g., GloVe[101] and FastText[102]), and pretrained models (e.g., BERT[103] and Turing-NLG[104]) available online. However, their training corpora are usually taken from general sources such as newspapers,[51] Twitter posts,[105] online reviews,[106] and Wikipedia pages[107] rather than scientific publications. In addition to common obstacles such as long-distance dependencies[108] and polysemant disambiguation,[109] the unique terminologies and language styles used in scientific publications pose unique challenges to the IE problem and make such model reuse problematic. Elsevier has conducted an evaluation of openIE systems on two datasets: one consisting of 200 sentences randomly selected from Wikipedia, and the second made up of 220 sentences from the scientific literature for the ten most published disciplines. Extractions were checked manually by five humans to guarantee evaluation accuracy, and the results showed that the extractors performed better on encyclopedic sentences (54% precision) than on scientific sentences (34% precision).[110]

Many IE challenges observed in other fields of science will certainly also be challenges in materials engineering. There is plenty of terminology that is not specific to materials engineering and yet is critical to understanding the content of materials text. In previous work, we have also encountered many challenges in that the same materials can be referred to by different names (e.g., trade names, specification number, and composition), which complicates both learning language models and organizing extracted data. Long-distance dependencies are also common in materials article, such as when the processing path for a material is described in a separate section from where its properties are discussed. Journal articles require significant training for humans to understand well, and it is similar for machines.

In this section, we examine the many gaps that prevent state-of-the-art NLP models from reaching their full potential on scientific literature, as well as techniques proposed to bridge the gaps (Fig. 6).

Standardized Datasets in NLP

As discussed in Challenge 2: The Need for (and Lack of) Training Data section, assembling an NLP dataset often requires considerable time and effort, which makes many researchers turn to precompiled datasets. Using these datasets saves NLP researchers valuable time and resources and provides a level playing field for comparing model performance, so it is no surprise to find that many models are solely developed and tested on standardized datasets. However, an often-overlooked problem is that these carefully curated datasets are not representative of text found "in the wild." Understandably, curators want their datasets to be of high quality, i.e., comprised of rich, balanced, and clean training examples. However, this curation process can lead to datasets and thus models that present a distorted view of how language and text are used.

As an example, CoNLL-2003 is a widely used dataset compiled from Reuters news stories to train NER models.[31] There are over 31,000 entities (location, organization, person, and misc) among its 22,000 sentences, averaging to about 1.5 entities per sentence. In contrast, when developing a polymer NER model, we found that around 84% of 12,000 sentences randomly selected from publications in the journal *Macromolecules* did not mention any polymer names,[39] resulting in a qualitatively different predictive task. Besides, natural language is rarely cut and dry. For example, in the polymer NER example, should "polymer a" and "polymer 1" be classified as polymers or not? While meaningless in isolation, these terms can be helpful for downstream tasks such as relation extraction with reference resolution. Such grey areas are usually excluded from standardized datasets to keep the dataset "clean." Consequently, models developed on such datasets expect all input to be as clean and noise-free, often leading to performance degradation. Perhaps, the definition of dataset quality should be reconsidered to include how well datasets reflect languages as they are used naturally.

Transfer Learning

Unique domain-specific languages are at the root of many incompatibilities between state-of-the-art NLP models and scientific literature. As discussed in the previous section, widely used datasets in the NLP community are usually compiled from nonscientific corpora. Yet the meaning of a word can vary drastically in different domains; for example, "PS" could be polystyrene, PostScript, or PhotoShop, depending on who you ask. Due to this variation, models trained on a generic corpora cannot be applied directly to extract information from scientific literature.

Transfer learning is an ML method that repurposes a model to a different task by reusing (transferring) knowledge from the task on which it was trained. Transfer learning has been applied to a

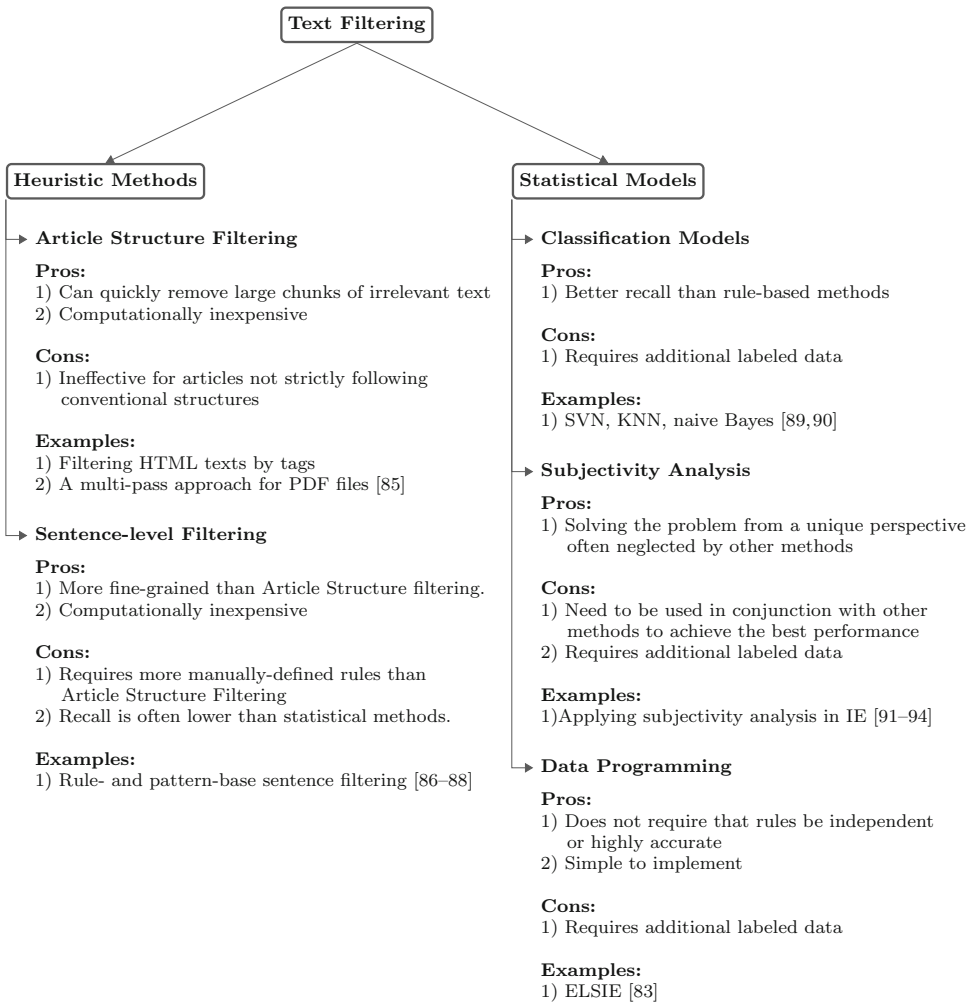


Fig. 5. Comparison of text filtering techniques. Heuristic methods are simple to implement and often achieve higher precision but lower recall. Statistical models are more complex and commonly have higher recall but lower precision.

variety of tasks, including image recognition,[113] machine translation,[114] and text classification.[115] It has also shown promising results in scientific research, such as medical imaging,[116] material property prediction,[117] and material defect detection.[118] Roughly speaking, transfer learning strategies can be divided into two classes, *inductive* and *transductive*. Inductive learning reuses a model for a different task in the same domain, whereas in transductive learning, the model is applied to a different domain but the task is similar to the one it was trained on. Transductive learning is particularly helpful in SciIE, since there is a plethora of models designed for various tasks but trained on generic texts. Common methods for transductive learning in NLP include leveraging pretrained word embeddings and fine-tuning pretrained models.

Word Embedding

Word embedding models map human-friendly words to computer-friendly vectors, in which the senses of words are embedded. With training, the word embedding vectors are fit to capture the meaning(s) of a word from the contexts it is used in. Due to the difficulty of judging directly whether a numeric vector accurately represents a word's semantic and syntactic information, most word embedding models are trained for specifically designed tasks that are closely related to how languages are used. When vectors can achieve satisfactory performance on the training task, they are presumed to be accurate representations of the original words. For example, Word2Vec, one of the most popular traditional word embedding models, is trained to predict a missing word based on its context.[119,120] After training, word vectors are derived from the weights of the hidden layers in the model and may be used in transfer learning applications.

Since word vectors are learned from context, the more often and accurately they are represented in the training corpus, the higher quality they will be. Although word embedding models can be trained using unsupervised methods, training on a large

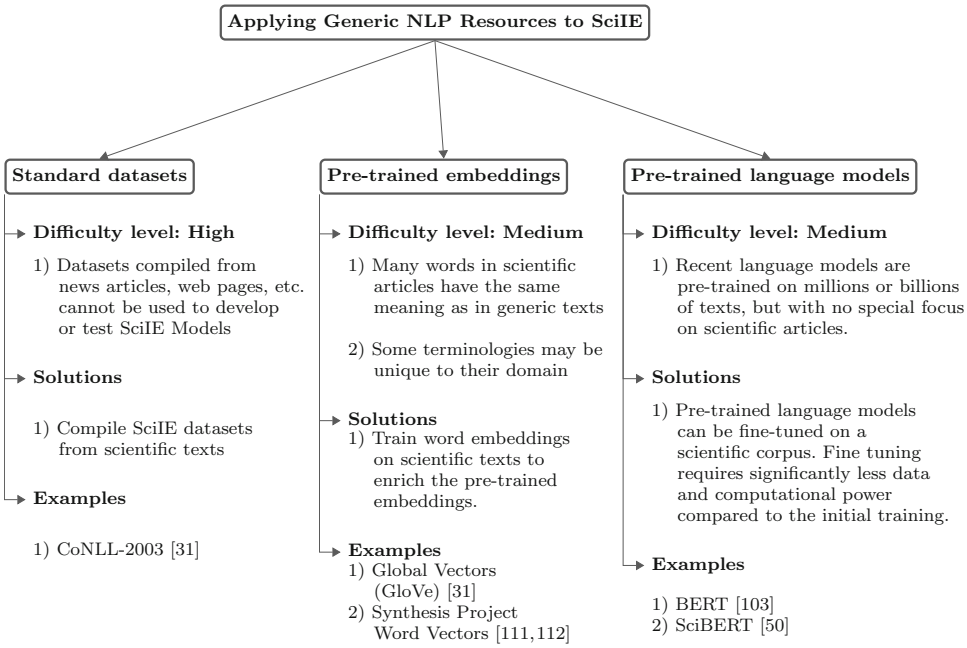


Fig. 6. Challenges with applying generic NLP methods to SciIE: datasets compiled from nonscientific sources cannot be applied to train SciIE models; word embeddings pretrained on generic texts need to be enriched with embeddings trained on scientific texts to include terminologies; pretrained language models require fine-tuning on scientific texts to achieve better understanding of domain-specific language.

corpus is often infeasible because of the associated computational requirements.[121] Therefore, word vectors are the primary form of transfer learning in NLP, and a number of word embedding models pretrained on large quantities of texts are available. The Google News vectors are 300-dimensional Word2Vec vectors pretrained on the Google News dataset of over 100 billion words.[122] The Global Vectors (GloVe) model of word representations has been used to produce word embeddings pretrained on over 900 billion words from the Web.[101] FastText extends the Word2Vec model with character n-gram embeddings and offers pretrained word vectors for 157 languages.[123] In materials science specifically, the Synthesis Project offers pretrained embeddings of the Word2Vec, FastText, and ELMo models.[111,112]

Transfer learning with word embeddings is based on the assumption that the meaning of a word is invariant across corpora, so that the vector for a word learned from one text or corporate can be used to represent the same word in a different text. Also, unlike human dictionaries, where one word can have multiple definitions, dictionaries generated by word embedding models only have one vector (value) for each word (key) in the vocabulary. However, multisense words are inherent in natural languages, and discipline-specific jargon makes the problem yet more complex. Sense2Vec, proposed to disambiguate word vectors,[109] creates multiple vectors corresponding to the different senses of a word. To this end, it requires the training corpus to have disambiguation labels. Although some disambiguation can be inferred automatically from POS labels or sentence parsing information, other cases still require manual annotation, sacrificing the most valuable trait of word embedding—unsupervised training.

Another, more widely adopted, solution to the multisense word problem in transfer learning is simply to increase the size of the word vectors, hoping that the extra dimensions will allow a single vector to embed multiple senses. This solution is more widely used than the sense vector solution because it does not affect the unsupervised training process. Instead, responsibility for learning the proper weights for the dimensions of the word vector space is handed off to the downstream models. The downside of this method is that it inflates downstream models, but with the rapidly increasing computational power offered by hardware accelerators such as graphical processing units (GPUs) and tensor processing units (TPUs), this trade-off is getting easier to bear.

Language Model Fine-Tuning

Many potentially valuable data are lost during transfer learning with word embeddings. Specifically, only the vector outputs of the word embedding model are kept and reused, while the parameters in other hidden layers are discarded after training. Also, word vectors are only used to initialize the first layer of the downstream model and kept frozen during training, thus knowledge gained from training with one downstream task cannot be transferred to another with word vectors. Therefore, the fine-tuning of pretrained language models has been gaining more traction as an alternative transfer learning method in NLP.

State-of-the-art language models such as BERT[103] and SciBERT[50] are transformer models based on the encoder–decoder architecture.[124] Like word embedding models, they are pretrained for generic language tasks (such as predicting a missing word). However, unlike word embedding models, they keep all the learnt weights after pretraining, with their model architecture allowing the same model to be applied to various tasks (e.g., sentence classification and NER) via *fine-tuning*. Fine-tuning is the process of retraining a model with new data specific to the task at hand. Since the model has been pretrained, fine-tuning only takes a fraction of the data and time needed to train the model from scratch. Fine-tuning can be done on all the layers, or, for encoder–decoder models, only on the top few layers near the output while all other layers are kept frozen. In this way, more valuable information is transferred from pretraining to specific downstream tasks. This method is especially helpful for using large models effectively. BERT is trained from scratch on a corpus of over 3.3 billion words on 16 TPUs in four days. Not every downstream task has a sufficiently large corpus or enough computation power to train such large models from scratch. Fine-tuning BERT, on the other hand, takes just a couple of hours on one GPU; thus, fine-tuning provides an efficient way for other tasks to reap the benefits of large, complex models.

ENABLING BETTER IE IN MATERIALS RESEARCH

As described above, significant challenges must be addressed before the data published over decades in various scientific literatures can be made readily accessible. Many challenges require improvements in NLP techniques, which are not necessarily the domain of researchers working in disciplines such as materials science and engineering. However, we note a few actions that the materials community can perform to make the most of the current state of the art and to ensure steady innovation in the future.

Access to Well-Formatted Versions of Articles

Access to machine-readable full-text articles (i.e., in HTML, XML, or JSON formats) is a prerequisite for robust SciIE, but obtaining such access, when possible at all, often involves lengthy negotiations with publishers. Such barriers limit growth of this field. Moves towards open publication of articles in ways that enable unfettered access, whether by traditional publishers or via technical-society-sponsored preprint repositories (e.g., ChemRxiv), will have major benefits for the NLP community and for science and engineering more broadly. Where full open access is not possible, streamlined licensing agreements or the establishment of repositories where researchers can perform analyses would make a big difference.

Open software practices can also contribute to lowering the cost of entry for scientific NLP research. Many NLP efforts begin by creating tools to streamline access to publisher application programming interfaces (APIs) or to handle the idiosyncratic formats in which articles are provided. Sharing such tools in a community-wide repository—which has been done with atomic-scale simulation outputs[125]—would remove the need for individual research groups repeatedly to recreate and maintain such tools.

Open Science Practices in Materials NLP Research

Building an IE pipeline to solve a specific problem involves much work that can be repurposed for other tasks. Tools to process articles in the formats used by different journals, word embeddings created from text from a specific literature, and ML models trained on relevant data can all be useful to others for new tasks. Of course, some elements of the pipeline (e.g., typeset versions of journal articles) are protected intellectual property (IP), but most are not. We have created a checklist of components that should be released for a paper to best ensure progress by the community:

1. *Preprocessing codes* to render journal articles or other text into a form ready to be used in IE pipelines.
2. *Training sets*, employed to develop NER and association models, are not just needed for reproducibility but are also critical to speeding development of NLP tools. Labeled datasets are often the most resource-intensive step (Sect. 4) in NLP, which makes them particularly valuable to share.
3. *Word embeddings*, which underpin most NLP models, are useful both to analyze for new materials[23] and to bootstrap new NLP efforts.
4. *Trained models*, the final product of NLP development, should be published to ensure that access to data does not backslide as tool developers move on to new projects.

Fortunately, the materials IE community has already established a culture of sharing such tools. The CHEMDNER datasets that were crucial in launching the field of NLP for chemical data extraction are open.[56] Swain and Cole also released their SciIE pipeline as an open-source tool, ChemDataExtractor,[126] that many teams have found useful[36] and that has been used to build a battery chemicals and properties database with over 290,000 records.[17] The Synthesis Project has also produced useful training datasets and tool releases.[57,111,112] Kononova et al. published a dataset of nearly 20,000 inorganic materials synthesis recipes extracted from text.[16] We hope that these

early, excellent examples of open science will set the stage for establishing a vibrant research community in NLP for materials science and engineering.

Publishing Structured Data

Ultimately, human language is an imperfect way to communicate materials data. Automated processing of large quantities of data works best when data are in identical formats and co-located with *all* the metadata needed to understand them. Homogeneity and conciseness are not features of human languages. Accordingly, better access to materials data may ultimately be achieved best not by processing text documents but by encouraging humans to record data in computer languages.

We already see adoption of such processes within certain subfields and projects in materials science and engineering. For example, inorganic crystal structures must be reported in the Crystallographic Information File (CIF)[127] format when published in journals of the International Union of Crystallography (IUCr). Density functional theory (DFT) computations are starting to be shared in their native formats via domain-specific databases that provide such data in searchable formats, such as NOMAD[128] and the Materials Data Facility.[129,130] Further, many individuals are beginning to publish their data as machine-accessible resources. Such activities focused on publishing data in computer-readable formats are important[131,132] but lie beyond the scope of this review. We do note that many subdisciplines within materials science, such as thermochemistry[133] and atomic-scale modeling,[134] are establishing community standards needed to bring order to a larger proportion of materials engineering data.

In short, scientists can reduce the need for SciIE by publishing information in computer-accessible forms. Ideally, all data would be published in a format that includes descriptions of how they were collected, presented according to the standards of a science community, as in the IUCr and NOMAD examples above. Publishing a large fraction of materials data in such structured formats will be a formidable challenge, but one that can be performed thoughtfully and in parallel with advancements in NLP. Electronic laboratory notebooks, laboratory information management systems, and robotics are particularly promising technologies to make digital data and metadata more prevalent in materials engineering.[135]

CONCLUSIONS AND OUTLOOK

Data are taking center stage in materials engineering and an increasing number of scientific fields, yet vast amounts of data remain and continue to be buried in written papers, inaccessible to humans and machines. In this paper, we have examined the major factors obstructing practical applications of computer-aided information extraction on scientific corpora, including the file formats of scientific publications, the lack of domain-specific training data, the sparsity of interesting information in papers, as well as the difficulties inherent in transferring a model trained on generic texts to scientific literature. We reviewed potential solutions or remedies for the problems, and discussed their strengths and weaknesses. We intend that this paper provide a clear overview of the current landscape of scientific information extraction and shed light on the many obstacles that future research efforts can take on.

Information extraction represents only an initial step towards using NLP to aid research in science and engineering by taking on tasks currently performed by human scientists. Outside science, modern NLP technologies are being used for increasingly complex tasks, such as answering questions in prose (e.g., Google's question-answering search engine[136]). A future NLP tool could use facts and qualitative relationships extracted from papers to enable autonomous reasoning engines capable of producing and testing hypotheses from the literature, or identifying anomalies worthy of further investigation. Solving the challenges currently inhibiting our ability to extract information from papers would unlock such a potential future for AI in materials science and engineering.

ACKNOWLEDGEMENTS

This work was performed under financial assistance award 70NANB19H005 from the US Department of Commerce, National Institute of Standards and Technology, as part of the Center for Hierarchical Materials Design (CHiMaD), and was also supported in part by the US Department of Energy, Office of Science, Advanced Scientific Computing Research, under Contract DE-AC02-06CH11357, and by the Joint Center for Energy Storage Research (JCESR), an Energy Innovation Hub funded by the US Department of Energy (DOE), Office of Science, Office of Basic Energy Sciences.

CONFLICT OF INTEREST

On behalf of all authors, the corresponding author states that there are no conflicts of interest.

REFERENCES

1. E. Landhuis, *Nature* **535**(7612), 457 (2016).
2. M. Ware, M. Mabe, *The STM Report: An Overview of Scientific and Scholarly Journal Publishing* (International Association of Scientific, Technical and Medical Publishers, Oxford, 2015).
3. G. Olson, *Scr. Mater.* **70**, 1 (2014).
4. J.J. de Pablo, N.E. Jackson, M.A. Webb, L.Q. Chen, J.E. Moore, D. Morgan, R. Jacobs, T. Pollock, D.G. Schlom, E.S. Toberer, J. Analytis, I. Dabo, D.M. DeLongchamp, G.A. Fiete, G.M. Grason, G. Hautier, Y. Mo, K. Rajan, E.J. Reed, E. Rodriguez, V. Stevanovic, J. Suntivich, K. Thornton, J.C. Zhao, *NPJ Comput. Mater.* **5**, 1 (2019).
5. J. Brandrup, E.H. Immergut, E.A. Grulke (eds.), *Polymer Handbook*, 4th edn. (Wiley, Hoboken, 2004).

6. S. Gražulis, D. Chateigner, R.T. Downs, A.F.T. Yokochi, M. Quirós, L. Lutterotti, E. Manakova, J. Butkus, P. Moeck, A.L. Bail, *J. Appl. Crystallogr.* **42**(4), 726 (2009).
7. S. Kirklin, J.E. Saal, B. Meredig, A. Thompson, J.W. Doak, M. Aykol, S. Rühl, C. Wolverton, *NPJ Comput. Mater.* **1**(1), 1 (2015).
8. C. Kim, A. Chandrasekaran, T.D. Huan, D. Das, R. Ramprasad, *J. Phys. Chem. C* **122**(31), 17575 (2018).
9. A. Jain, S.P. Ong, G. Hautier, W. Chen, W.D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder et al., *APL Mater.* **1**(1), 011002 (2013).
10. C. Borkowski, J. Sperling Martin, *J. Am. Soc. Inform. Sci.* **26**(2), 94 (1975).
11. F.B. Rogers, *Bull. Med. Libr. Assoc.* **52**(1), 150 (1964).
12. R.J. Roberts, *Proc. Natl. Acad. Sci.* 98(2), 381 (2001). https://doi.org/10.1073/pnas.98.2.381.https://www.pnas.org/content/98/2/381.
13. D.R. Swanson, N.R. Smalheiser, *Artif. Intell.* **91**(2), 183 (1997).
14. L. Tanabe, U. Scherf, L. Smith, J. Lee, L. Hunter, J. Weinstein, *Biotechniques* **27**(6), 1210 (1999).
15. E.A. Olivetti, J.M. Cole, E. Kim, O. Kononova, G. Ceder, T.Y.J. Han, A.M. Hiszpanski, *Appl. Phys. Rev.* **7**(4), 041317 (2020).
16. O. Kononova, H. Huo, T. He, Z. Rong, T. Botari, W. Sun, V. Tshitoyan, G. Ceder, *Sci. Data* **6**(1), 1 (2019).
17. S. Huang, J.M. Cole, *Sci. Data* **7**(1), 1 (2020).
18. Prodi.gy. Prodi.gy: An annotation tool for AI, Machine Learning, and NLP. https://prodi.gy (2021). Accessed on 02 May 2021.
19. C.A. Clark, S.K. Divvala, in *AAAI Workshop: Scholarly Big Data*, vol. 6 (2015).
20. Y. Liu, K. Bai, P. Mitra, C.L. Giles, in *Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries* (2007), p. 91.
21. B. Gatos, D. Danatsas, I. Pratikakis, S.J. Perantonis, *International Conference on Pattern Recognition and Image Analysis* (Springer, New York, 2005), p. 609.
22. I. Kavasidis, C. Pino, S. Palazzo, F. Rundo, D. Giordano, P. Messina, C. Spampinato, *International Conference on Image Analysis and Processing* (Springer, New York, 2019), p. 292.
23. V. Tshitoyan, J. Dagdelen, L. Weston, A. Dunn, Z. Rong, O. Kononova, K.A. Persson, G. Ceder, A. Jain, *Nature* **571**(7763), 95 (2019).
24. D. Nadeau, S. Sekine, *Lingvist. Invest.* **30**(1), 3 (2007).
25. J. Li, A. Sun, J. Han, C. Li, *IEEE Trans. Knowl. Data Eng.* (2020).
26. Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, S. Fidler, *IEEE Int. Conf. Comput. Vis.* (2015), p. 19.
27. C. Sun, Z. Yang, L. Wang, Y. Zhang, H. Lin, J. Wang, *J. Biomed. Inform.* **103**, 103392 (2020).
28. A. Yates, M. Banko, M. Broadhead, M.J. Cafarella, O. Etzioni, S. Soderland, *Annual Conference of the North American Chapter of the Association for Computational Linguistics* (2007), p. 25.
29. F. Wu, D.S. Weld, in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (2010), p. 118.
30. G. Angeli, M.J.J. Premkumar, C.D. Manning, in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and 7th International Joint Conference on Natural Language Processing* (2015), p. 344.
31. E.F. Tjong Kim Sang, F. De Meulder, in *Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL 2003* (2003), p. 142.
32. Y. Zhang, V. Zhong, D. Chen, G. Angeli, C.D. Manning, in *Conference on Empirical Methods in Natural Language Processing* (2017), p. 35.
33. PDFTron. PDF2Text. https://www.pdftron.com/documentation/cli/guides/pdf2text/ (2021). Accessed on 15 Feb 2021.
34. C. Ramakrishnan, A. Patnia, E. Hovy, G.A. Burns, *Source Code Biol. Med.* **7**(1), 1 (2012).
35. M.M. Mirończuk, *Knowl. Inf. Syst.* **54**(3), 711 (2018).
36. R.B. Tchoua, K. Chard, D. Audus, J. Qin, J. de Pablo, I. Foster, *Proc. Comput. Sci.* **80**, 386 (2016).
37. R.B. Tchoua, K. Chard, D.J. Audus, L.T. Ward, J. Lequieu, J.J. De Pablo, I.T. Foster, in *IEEE 13th International Conference on e-Science* (IEEE, 2017), p. 109.
38. R. Tchoua, A. Ajith, Z. Hong, L. Ward, K. Chard, D. Audus, S. Patel, J. de Pablo, I. Foster, in *Proceedings of the 15th International Conference on eScience* (IEEE, 2019), p. 126.
39. Z. Hong, R. Tchoua, K. Chard, I. Foster, in *International Conference on Computational Science* (Springer, 2020), p. 308.
40. R. Tchoua, Z. Hong, D. Audus, S. Patel, L. Ward, K. Chard, J. De Pablo, I. Foster, *Bull. Am. Phys. Soc.* 65 (2020).
41. L. Von Ahn, B. Maurer, C. McMillen, D. Abraham, M. Blum, *Science* **321**(5895), 1465 (2008).
42. F. Hillen, B. Höfle, *Int. J. Appl. Earth Obs. Geoinf.* **40**, 29 (2015).
43. S. Yan, W.S. Spangler, Y. Chen, *IEEE/ACM Trans. Comput. Biol. Bioinf.* **10**(5), 1218 (2013).
44. A.J. Yepes, A. MacKinlay, N. Gunn, C. Schieber, N. Faux, M. Downton, B. Goudey, R.L. Martin, in *AMIA Annual Symposium Proceedings*, vol. 2018 (American Medical Informatics Association, 2018), vol. 2018, p. 616.
45. K. Ganchev, F. Pereira, M. Mandel, S. Carroll, P. White, in *Proceedings of the linguistic annotation workshop* (2007), p. 53.
46. Y. Jo, E. Mayfield, C. Reed, E. Hovy, in *Proceedings of the 12th Language Resources and Evaluation Conference* (2020), p. 1008.
47. Z. Hong, J.G. Pauloski, L. Ward, K. Chard, B. Blaiszik, I. Foster, arXiv preprint arXiv:2101.04617 (2021).
48. K. Lybarger, M. Ostendorf, M. Yetisgen, *J. Biomed. Inform.* **113**, 103631 (2021).
49. S.M. Swanberg, *J. Med. Libr. Assoc.* **105**(1), 106 (2017).
50. I. Beltagy, K. Lo, A. Cohan, in *Conference on Empirical Methods in Natural Language Processing* (2019).
51. M. Marcus, B. Santorini, M.A. Marcinkiewicz, Building a large annotated corpus of English: The Penn Treebank. Technical Report MS-CIS-93-8, University of Pennsylvania, Department of Computer and Information Science (1993).
52. K. Bontcheva, I. Roberts, L. Derczynski, S. Alexander-Eames, in *Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics* (2014), p. 9.
53. B.M. Good, M. Nanis, C. Wu, A.I. Su, *Pacific Symposium on Biocomputing* (World Scientific, Singapore, 2014), p. 282.
54. C.G. Northcutt, A. Athalye, J. Mueller, arXiv preprint arXiv:2103.14749 (2021).
55. R.B. Tchoua, J. Qin, D.J. Audus, K. Chard, I.T. Foster, J. de Pablo, *J. Chem. Edu.* **93**(9), 1561 (2016).
56. M. Krallinger, O. Rabal, F. Leitner, M. Vazquez, D. Salgado, Z. Lu, R. Leaman, Y. Lu, D. Ji, D.M. Lowe, R.A. Sayle, R.T. Batista-Navarro, R. Rak, T. Huber, T. Rocktäschel, S. Matos, D. Campos, B. Tang, H. Xu, T. Munkhdalai, K.H. Ryu, S. Ramanan, S. Nathan, S. Žitnik, M. Bajec, L. Weber, M. Irmer, S.A. Akhondi, J.A. Kors, S. Xu, X. An, U.K. Sikdar, A. Ekbal, M. Yoshioka, T.M. Dieb, M. Choi, K. Verspoor, M. Khabsa, C.L. Giles, H. Liu, K.E. Ravikumar, A. Lamurias, F.M. Couto, H.J. Dai, R.T.H. Tsai, C. Ata, T. Can, A. Usié, R. Alves, I. Segura-Bedmar, P. Martínez, J. Oyarzabal, A. Valencia, *J. Cheminform.* **7**(1), 1 (2015).
57. S. Mysore, Z. Jensen, E. Kim, K. Huang, H.S. Chang, E. Strubell, J. Flanigan, A. McCallum, E. Olivetti, in *Proceedings of the 13th Linguistic Annotation Workshop* (Association for Computational Linguistics, 2019), p. 56.
58. A. Peskin, A. Dima, *Integ. Mater. Manuf. Innov.* **6**(2), 187 (2017).
59. L. Von Ahn, *Computer* **39**(6), 92 (2006).
60. A. Kawrykow, G. Roumanis, A. Kam, D. Kwak, C. Leung, C. Wu, E. Zarour, L. Sarmenta, M. Blanchette, J. Waldispühl, *PLoS ONE* **7**(3), e31362 (2012).

61. B. Guillaume, K. Fort, N. Lefebvre, in *International Conference on Computational Linguistics* (2016).
62. H.A. Favre, W.H. Powell, *Nomenclature of Organic Chemistry: IUPAC Recommendations and Preferred Names 2013* (Royal Society of Chemistry, London, 2013).
63. H.L. Morgan, *J. Chem. Doc.* **5**(2), 107 (1965).
64. C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, S. Hellmann, *J. Web Sem.* **7**(3), 154 (2009).
65. B. Settles, *Synth. Lect. Artif. Intell. Mach. Learn.* **6**(1), 1 (2012).
66. A.R. Camacho, in *Proceedings of the 14th IAPR International Workshop on Document Analysis Systems*, vol. 12116 (Springer, 2020), p. 324.
67. M. Mintz, S. Bills, R. Snow, D. Jurafsky, in *Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP* (2009), p. 1003.
68. S. Riedel, L. Yao, A. McCallum, in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (Springer, 2010), p. 148.
69. M. Surdeanu, J. Tibshirani, R. Nallapati, C.D. Manning, in *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (2012), p. 455.
70. T. Liu, K. Wang, B. Chang, Z. Sui, in *Conference on Empirical Methods in Natural Language Processing* (2017), p. 1790.
71. W. Xu, R. Hoffmann, L. Zhao, R. Grishman, in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (2013), p. 665.
72. T. Onishi, T. Kadohira, I. Watanabe, *Sci. Technol. Adv. Mater.* **19**(1), 649 (2018).
73. K. Ravikumar, H. Liu, J.D. Cohn, M.E. Wall, K. Verspoor, *J. Biomed. Sem.* **3**(3), 1 (2012).
74. C. Quirk, H. Poon, in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics* (2017), p. 1171.
75. D. Buscaldi, D. Dessì, E. Motta, F. Osborne, D.R. Recupero, in *European Semantic Web Conference* (Springer, 2019), p. 8.
76. A. Fader, S. Soderland, O. Etzioni, in *Conference on Empirical Methods in Natural Language Processing* (2011), p. 1535.
77. S. Soderland, B. Roof, B. Qin, S. Xu, O. Etzioni, *AI Mag.* **31**(3), 93 (2010).
78. Y. Luan, L. He, M. Ostendorf, H. Hajishirzi, in *Conference on Empirical Methods in Natural Language Processing* (2018), p. 3219.
79. R. Kruiper, J.F. Vincent, J. Chen-Burger, M.P. Desmulliez, I. Konstas, arXiv preprint arXiv:2005.07751 (2020).
80. K. White, Publications output: US trends and international comparisons. Technical report, National Science Foundation (2019). https://ncses.nsf.gov/pubs/nsb20206/.
81. E. Riloff, in *Proceedings of the 11th National Conference on Artificial Intelligence* (1993), p. 811.
82. S. Soderland, *Mach. Learn.* **34**(1), 233 (1999).
83. E. Murphy, Ensemble labeling towards scientific information extraction (ELSIE). Ph.D. thesis, College of Computing and Digital Media (2020).
84. I. Hendrickx, S.N. Kim, Z. Kozareva, P. Nakov, D. Ó Séaghdha, S. Padó, M. Pennacchiotti, L. Romano, S. Szpakowicz, in *Proceedings of the 5th International Workshop on Semantic Evaluation* (Association for Computational Linguistics, 2010), p. 33.
85. D.D.A. Bui, G. Del Fiol, S. Jonnalagadda, *J. Biomed. Inform.* **61**, 141 (2016).
86. C. Blaschke, L. Hirschman, A. Valencia, *Brief. Bioinform.* **3**(2), 154 (2002).
87. K.B. Cohen, K. Verspoor, H.L. Johnson, C. Roeder, P. Ogren, W.A. Baumgartner, E. White, L. Hunter, in *BioNLP 2009 Workshop Companion Volume for Shared Task* (2009), p. 50.
88. Q.L. Nguyen, D. Tikk, U. Leser, *J. Biomed. Sem.* **1**(1), 1 (2010).
89. V. Pillet, *Méthodologie d'extraction automatique d'information à partir de la littérature scientifique en vue d'alimenter un nouveau système d'information: application à la génétique moléculaire pour l'extraction d'information sur les interactions*. Ph.D. thesis, Univ. d'Aix-Marseille 3 (2000).
90. J.R. Quinlan, *C4.5: Programs for Machine Learning* (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993).
91. C. Nédellec, M.O.A. Vetah, P. Bessieres, in *European Conference on Principles of Data Mining and Knowledge Discovery* (Springer, 2001), p. 326.
92. A.H. Aliwy, E.A. Ameer, *Int. J. Appl. Eng. Res.* **12**(14), 4309 (2017).
93. E. Riloff, J. Wiebe, W. Phillips, in *AAAI* (2005), p. 1106.
94. E. Riloff, J. Wiebe, T. Wilson, in *Proceedings of the 7th Conference on Natural Language Learning* (2003), p. 25.
95. J. Wiebe, E. Riloff, in *International Conference on Intelligent Text Processing and Computational Linguistics* (Springer, 2005), p. 486.
96. J. Wiebe, E. Riloff, *IEEE Trans. Affect. Comput.* **2**(4), 175 (2011).
97. A. Ratner, S.H. Bach, H. Ehrenberg, J. Fries, S. Wu, C. Ré, *Int. Conf. Very Large Data Bases* **11**(3), 269 (2017).
98. A.J. Ratner, S.H. Bach, H.R. Ehrenberg, C. Ré, in *ACM International Conference on Management of Data* (2017), p. 1683.
99. E.F. Sang, F. De Meulder, arXiv preprint cs/0306050 (2003).
100. R. Weischedel, S. Pradhan, L. Ramshaw, M. Palmer, N. Xue, M. Marcus, A. Taylor, C. Greenberg, E. Hovy, R. Belvin, A. Houston, OntoNotes Release 5.0. Web download, Linguistic Data Consortium (2013). https://doi.org/10.35111/xmhb-2b84. https://catalog.ldc.upenn.edu/LDC2013T19.
101. J. Pennington, R. Socher, C.D. Manning, in *Conference on Empirical Methods in Natural Language Processing* (2014), p. 1532.
102. T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, A. Joulin, in *International Conference on Language Resources and Evaluation* (2018).
103. J. Devlin, M.W. Chang, K. Lee, K. Toutanova, in *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (Association for Computational Linguistics, 2019), p. 4171.
104. C. Rosset, Microsoft Research Blog (2020). https://bit.ly/3eF1coS.
105. H. Saif, M. Fernandez, Y. He, H. Alani, in *Proceedings of the 1st International Workshop on Emotion and Sentiment in Social and Expressive Media: Approaches and Perspectives from AI* (2013).
106. A.L. Maas, R.E. Daly, P.T. Pham, D. Huang, A.Y. Ng, C. Potts, in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (Association for Computational Linguistics, 2011), p. 142.
107. H. Elsahar, P. Vougiouklis, A. Remaci, C. Gravier, J. Hare, F. Laforest, E. Simperl, in *Proceedings of the 11th International Conference on Language Resources and Evaluation* (European Language Resources Association, 2018).
108. W. Sun, X. Peng, X. Wan, in *Proceedings of the Sixth International Joint Conference on Natural Language Processing* (2013), p. 180.
109. A. Trask, P. Michalak, J. Liu, arXiv preprint arXiv:1511.06388 (2015).
110. P. Groth, M. Lauruhn, A. Scerri, R. Daniel, arXiv preprint arXiv:1802.05574 (2018).
111. E. Kim, K. Huang, A. Tomala, S. Matthews, E. Strubell, A. Saunders, A. McCallum, E. Olivetti, *Sci. Data* **4**(1), 1 (2017).

112. E. Kim, Z. Jensen, A. van Grootel, K. Huang, M. Staib, S. Mysore, H.S. Chang, E. Strubell, A. McCallum, S. Jegelka, E. Olivetti, *J. Chem. Inf. Model.* **60**(3), 1194 (2020).
113. D.S. Maitra, U. Bhattacharya, S.K. Parui, in *Proceedings of the 13th International Conference on Document Analysis and Recognition* (IEEE, 2015), p. 1021.
114. Y. Wu, M. Schuster, Z. Chen, Q.V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, J. Dean, arXiv preprint arXiv:1609.08144 (2016).
115. C.B. Do, A.Y. Ng, *Adv. Neural. Inf. Process. Syst.* **18**, 299 (2005).
116. M. Raghu, C. Zhang, J. Kleinberg, S. Bengio, in *Proceedings of the 33rd Conference on Neural Information Processing Systems* (2019).
117. H. Yamada, C. Liu, S. Wu, Y. Koyama, S. Ju, J. Shiomi, J. Morikawa, R. Yoshida, *ACS Cent. Sci.* **5**(10), 1717 (2019).
118. Y. Gong, H. Shao, J. Luo, Z. Li, *Compos. Struct.* **252**, 112681 (2020).
119. T. Mikolov, K. Chen, G. Corrado, J. Dean, arXiv preprint arXiv:1301.3781 (2013).
120. T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, in *Proceedings of the 26th International Conference on Neural Information Processing Systems* (2013), p. 3111.
121. T.B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., arXiv preprint arXiv:2005.14165 (2020).
122. Google. Google News Word2Vec. https://code.google.com/archive/p/word2vec/ (2021). Accessed 07 Apr 2021.
123. É. Grave, P. Bojanowski, P. Gupta, A. Joulin, T. Mikolov, in *Proceedings of the 11th International Conference on Language Resources and Evaluation* (2018).
124. I. Sutskever, O. Vinyals, Q.V. Le, arXiv preprint arXiv:1409.3215 (2014).
125. A.H. Larsen, J.J. Mortensen, J. Blomqvist, I.E. Castelli, R. Christensen, M. Dułak, J. Friis, M.N. Groves, B. Hammer, C. Hargus, E.D. Hermes, P.C. Jennings, P.B. Jensen, J. Kermode, J.R. Kitchin, E.L. Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J.B. Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schütt, M. Strange, K.S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, K.W. Jacobsen, *J. Phys. Condens. Matter* **29**(27), 273002 (2017). https://doi.org/10.1088/1361-648x/aa680e.
126. M.C. Swain, J.M. Cole, *J. Chem. Inf. Model.* **56**(10), 1894 (2016).
127. S.R. Hall, F.H. Allen, I.D. Brown, *Acta Crystallogr. A* **47**(6), 655 (1991).
128. C. Draxl, M. Scheffler, *MRS Bull.* **43**(9), 676 (2018).
129. B. Blaiszik, K. Chard, J. Pruyne, R. Ananthakrishnan, S. Tuecke, I. Foster, *J. Mater.* (2016).
130. B. Blaiszik, L. Ward, M. Schwarting, J. Gaff, R. Chard, D. Pike, K. Chard, I. Foster, *MRS Commun.* **9**(4), 1125 (2019).
131. M.R. Seringhaus, M.B. Gerstein, *BMC Bioinform.* **8**(1), 1 (2007).
132. B. Mons, H. van Haagen, C. Chichester, J.T. den Dunnen, G. van Ommen, E. van Mulligen, B. Singh, R. Hooft, M. Roos, J. Hammond et al., *Nat. Genet.* **43**(4), 281 (2011).
133. M. Frenkel, R.D. Chiroco, V. Diky, Q. Dong, K.N. Marsh, J.H. Dymond, W.A. Wakeham, S.E. Stein, E. Königsberger, A.R.H. Goodwin, *Pure Appl. Chem.* **78**(3), 541 (2006). https://doi.org/10.1351/pac200678030541.
134. C.W. Andersen, R. Armiento, E. Blokhin, G.J. Conduit, S. Dwaraknath, M.L. Evans, Á. Fekete, A. Gopakumar, S. Gražulis, A. Merkys, F. Mohamed, C. Oses, G. Pizzi, G.M. Rignanese, M. Scheidgen, L. Talirz, C. Toher, D. Winston, R. Aversa, K. Choudhary, P. Colinet, S. Curtarolo, D.D. Stefano, C. Draxl, S. Er, M. Esters, M. Fornari, M. Giantomassi, M. Govoni, G. Hautier, V. Hegde, M.K. Horton, P. Huck, G. Huhs, J. Hummelshøj, A. Kariryaa, B. Kozinsky, S. Kumbhar, M. Liu, N. Marzari, A.J. Morris, A.A. Mostofi, K.A. Persson, G. Petretto, T. Purcell, F. Ricci, F. Rose, M. Scheffler, D. Speckhard, M. Uhrin, A. Vaitkus, P. Villars, D. Waroquiers, C. Wolverton, M. Wu, X. Yang, *Sci. Data* **8**, 1 (2021). https://doi.org/10.1038/s41597-021-00974-z.
135. L. Ward, M. Aykol, B. Blaiszik, I. Foster, B. Meredig, J. Saal, S. Suram, *MRS Bull.* **43**(9), 683 (2018). https://doi.org/10.1557/mrs.2018.204.
136. D. Metzler, Y. Tay, D. Bahri, M. Najork, arXiv preprint arXiv:2105.02274 (2021).