Towards a Hybrid Human-Computer Scientific Information Extraction Pipeline

Roselyne B. Tchoua*, Kyle Chard†, Debra J. Audus‡, Logan T. Ward†,
Joshua Lequieu§, Juan J. de Pablo§ and Ian T. Foster*†¶
*Department of Computer Science, University of Chicago, Chicago, IL, USA
Email: roselyne@uchicago.edu
†The Computation Institute, University of Chicago and Argonne, Chicago, IL, USA
‡Materials Science and Engineering Division, National Institute of Standards and Technology, Gaithersburg, MD, USA
§Institute for Molecular Engineering, University of Chicago, Chicago, IL, USA
¶Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, USA

Abstract—**The emerging field of materials informatics has the potential to greatly reduce time-to-market and development costs for new materials. The success of such efforts hinges on access to large, high-quality databases of material properties. However, many such data are only to be found encoded in text within esoteric scientific articles, a situation that makes automated extraction difficult and manual extraction time-consuming and error-prone. To address this challenge, we present a hybrid Information Extraction (IE) pipeline to improve the machine-human partnership with respect to extraction quality and person-hours, through a combination of rule-based, machine learning, and crowdsourcing approaches. Our goal is to leverage computer and human strengths to alleviate the burden on human curators by automating initial extraction tasks before prioritizing and assigning specialized curation tasks to humans with different levels of training: using non-experts for straightforward tasks such as validation of higher accuracy results (e.g., completing partial facts) and domain experts for low-certainty results (e.g., reviewing specialized compound labels). To validate our approaches, we focus on the task of extracting the glass transition temperature of polymers from published articles. Applying our approaches to 6 090 articles, we have so far extracted 259 refined data values. We project that this number will grow considerably as we tune our methods and process more articles, to exceed that found in standard, expert-curated polymer data handbooks while also being easier to keep up-to-date. The freely available data can be found on our Polymer Properties Predictor and Database website at http://pppdb.uchicago.edu.**

Index Terms—**Information Extraction, Crowdsourcing, Machine Learning, Polymers, Glass transition**

I. Introduction

Materials informatics [1–3], often referred to as the fourth paradigm of materials discovery [4, 5], combines large datasets and computational models to identify candidates for new materials, with the goal of reducing both time-to-market and development costs. As such methods rely on access to large, machine-readable databases, the traditional text-based physical handbooks will not suffice. However, there are few examples of these scientific digital databases and constructing new ones is a monumental task requiring years of expert labor, as the data that populate these databases must often be extracted manually from free-text publications. One excellent example of a digital database is PolyInfo [6], which contains the records for over 200 000 properties of polymers extracted from more than 12 000 articles—a process that required years of expert curation effort. Achieving databases as large and useful as PolyInfo for different material properties at a rate commensurate with the time-to-market goals of modern materials engineering is a daunting task. It might appear that automated methods of extraction could solve this problem; however, despite considerable progress in natural language processing (NLP) and machine learning [7–13], fully automated extraction is not yet possible due to the complexity by which such properties are encoded in publications. Instead, human effort is needed to develop rules, define training sets, and validate results [14–16].

In response, we propose a hybrid Information Extraction (IE) pipeline that combines automation and crowdsourcing in ways that leverage the complementary strengths of computational modules and humans. This pipeline first extracts candidate properties automatically and subsequently assigns various curation tasks to humans with the goal of maximizing throughput and accuracy while minimizing the burden on human curators. We applied a preliminary version of this concept to extract 263 values for the Flory-Huggins interaction parameter, a measure of miscibility between two entities—typically a polymer and either another polymer or a solvent [17]. In that case, we automatically browsed and searched a relevant journal in polymer science for this property to identify candidate articles and trained student reviewers to extract data: an effective but still relatively costly approach [18]. Here, we extend that work, increasing the automation to develop an integrated IE pipeline that combines a general-purpose NLP toolkit to parse text and perform preliminary recognition; specialized domain-specific models to identify entities and relationships; a ranking system to prioritize crowdsourced tasks; and a crowdsourcing framework to review candidate relationships. We apply this system to extract the *glass transition temperature* (T_g) of polymers. This important property in the design of new polymeric materials quantifies the temperature at which polymers transition from a glassy state into a rubbery state. Values for this parameter are often found in the text of scientific articles.

We used this new IE pipeline to process 6 090 articles

978-1-5386-2686-3/17 $31.00 © 2017 IEEE
DOI 10.1109/eScience.2017.23

published over the last decade in *Macromolecules*, a prominent journal in polymer science. In the first pipeline step, an NLP-based extraction process identified 1 442 ***T_g* candidates** in these articles—text fragments with characteristics suggestive of a T_g value, but often with various irregularities. Subsequent automated and crowdsourcing curation steps then processed these candidates, in some cases confirming and/or completing a polymer–T_g value and in others establishing that no such value is in fact present. Curating the output of the NLP extraction required only a half-hour of expert time and a combined six hours of untrained crowds. To date, we have extracted 259 T_g values from a subset of our articles and expect this number to increase dramatically as we improve our pipeline and apply it to new data. In comparison, the recent edition of the expert-curated *Physical Properties of Polymer Handbook* [19], last published in 2007, contains only $\approx$600 T_g values. The most recent and machine-accessible output of our IE pipeline is freely available at http://pppdb.uchicago.edu and https://materialsdatafacility.org [20].

The primary contributions of this paper are: (1) the design of a hybrid extraction pipeline that combines computer and human strengths; (2) demonstration that this method can accurately extract properties from publications; and (3) design and evaluation of extraction and curation tools, including a rule-based parser for T_g, a polymer identification module for distinguishing polymers from other chemical compounds, a polymer proximity search module for recovering polymer names from related text, crowdsourcing modules for identifying unrecognized polymers and flagging anomalous polymer names, and a prioritization model to guide curation effort.

The rest of this paper is as follows. Section II reviews related work in the information extraction of scientific facts. Section III motivates the problem by introducing T_g and discusses the challenges associated with automated extraction. Section IV describes the design and implementation of our IE pipeline. Section V evaluates the accuracy of the various stages in our pipeline. We discuss future work in Section VI before concluding in Section VII.

II. Related Work

IE methods have been applied in various scientific domains. The medical community has long been interested in the automated extraction and aggregation of data from medical text. Medical Language Extraction and Encoding System (MedLEE) [21, 22], cTAKES [23], and medKAT [24] are NLP tools specialized for the medical domain. These tools are designed to extract clinical information from text documents and to translate entities and terms to controlled ontologies and vocabularies. Much research in this domain has focused on the complexity of clinical text, for example there are significant challenges identifying negation, family relationships, temporality, and uncertainty. The general purpose nature of these tools also allows more sophisticated and specialized applications to be developed. For example, MedLEE has been adapted to build biomolecular and genotype-phenotype networks (GENIES [25] and BioMedLEE [26], respectively). These tools tend to be specialized and rely heavily on the development of ontologies, a tedious and time consuming process. Similarly, several NLP tools have recently been developed to mine data from patents and scientific literature in chemistry and materials science [11, 12, 27].

With the recent advances in machine learning and statistical inference approaches, scientific applications are turning their attention to deep learning tools such as DeepDive [13]. PaleoDeepDive [28], built upon DeepDive, automatically extracts paleontological data from text, tables, and figures in scientific publications. GeoDeepDive [29] performs similar tasks in the geosciences. For good performance in such applications, IE software often relies on and extends large databases: for example, PaleoDeepDive builds on PaleoDB [30] and GeoDeepDive builds on Macrostrat [31]. However, many fields, including materials science, do not yet have access to large and structured sets of texts that deep learning systems can use to learn scientific facts and relationships. The IE pipeline is an intermediary, but essential, step towards accumulating such structured data.

Because of the challenges in fully automated IE systems (e.g., dependence on ontologies and/or large training datasets) but also for validation purposes, humans are often involved in the extraction of scientific facts as domain experts. There is also recent interest in using crowdsourcing or "human computation" to solve problems that computers cannot handle correctly or cost-efficiently. Previous work has leveraged crowdsourcing to support extraction of data from tables within PDF documents [16] and also to ensure human quality control (i.e., expert curation) [15] while extracting empirical observations from literature. CrowdDB [32] uses human input to answer queries that neither database systems nor search engines can adequately answer due to the nature of the queries (e.g., discovering new data not included in a database). In our work, we aim to identify such cases—where humans are better suited for a task—and use the complementary strengths of humans and computers to populate a database of scientific facts. Wallace et al. [33] also pursue this goal, using a hybrid machine learning and crowdsourcing approach to identify published randomized controlled trials (RCTs) [33]. They use machine learning classifiers to recognize citations that are deemed highly unlikely to describe RCTs, deferring to crowdsourcing otherwise.

III. Motivation

We first provide a brief description of polymers and T_g and review both the state-of-the-art in NLP and the challenges associated with its application to the T_g problem.

A. Glass Transition Temperature

Polymers are molecules formed by covalently bonding small molecules, referred to as monomers, together. As the resulting polymer molecules generally have large molecular masses, potentially exceeding three orders of magnitude greater than water, they are sometimes referred to as macromolecules. Due in part to their large molecular masses, often in the form of long chains, polymers have a variety of useful properties. For example, the long chains of *poly(ethylene terephthalate)* become entangled, making them harder to pull apart; this results in strong but lightweight water bottles. In addition to being strong, many synthetic polymers are also extremely cheap as they can be synthesized from petroleum-based feedstocks. The combination of low cost and useful properties has resulted in polymers becoming a ubiquitous part of life.

In the design of new polymeric materials, the temperature relative to the T_g can have a profound effect on the properties of the polymeric material. T_g is defined as the temperature at which a polymer transitions from a solid, amorphous, glassy state to a rubbery state as the temperature is increased. Physically, when polymers are in the glassy state, the molecules are trapped and cannot move past each other due to a lack of thermal energy, while when they are in the rubbery state, the molecules are mobile. As the properties for the two states are drastically different, the glass transition plays a key role in both choosing a polymer for a given application and in the processing of the polymeric material. For example, plexiglass (*poly(methyl methacrylate)*), used as a lightweight substitute for glass, has a high T_g of roughly 110 °C, while neoprene (*polychloroprene*), used for laptop sleeves, has a low T_g of roughly -50 °C [34]. Exact, as opposed to rough, values of T_g require additional contextual information such as the molecular mass. We plan to capture such information in future work. However, as extracting contextual information is significantly more challenging than the already difficult task of extracting polymer–T_g pairs from literature, we focus on the polymer–T_g pairs first.

B. Natural Language Processing

Rule-based methods are commonly used for simple information extraction tasks. Such methods are straightforward to understand and allow developers to trace and fix errors; they are suitable for simple, well-defined problems (e.g., extracting spouses by identifying the subject and object in sentences containing the word *married*). However, they require tedious effort to construct and modify, as many rules are typically required to extract the same information expressed in various forms. In contrast, statistical and machine learning techniques are trainable, adaptable, and require little manual labor; however, they are opaque and require training data. Researchers often combine the two methods to increase the completeness and accuracy of extracted information [35]. Still, challenges remain, including the lack of the annotated corpora need to train machine-learning models. The lack of corpora is particularly common in fields such as bioinformatics [36] and our own, polymer science. Other challenges, not limited to specific scientific domains, include automatically deciphering subtleties in the English language, in general, and language particular to the domain itself. In polymer synthesis papers, for example, authors sometimes omit the name of the polymer, instead referencing or describing the underlying chemistry. In these cases, the polymer name is not readily apparent, and may require an expert polymer scientist to extract that information.

IV. Design and Implementation

The desired output of our pipeline is a set of polymer–T_g pairs, which can then be used to construct a machine-accessible database of values. Thus, the task can be seen as a two-part process consisting of recognizing polymer names and temperatures and establishing a relationship (t is a T_g of p) between pairs of entities. In order to reduce the burden on curators, we combine complementary human and machine strengths throughout our pipeline. We base our pipeline on a leading materials NLP toolkit, ChemDataExtractor [12], and develop automated and crowdsourcing modules to extract and curate polymer–T_g pairs. We focus here on extracted text excerpts containing a single T_g value. While multiple T_g values may be reported for a single polymer (e.g., prepared with different processing methods), we focus on pairs of polymers mapped to a single T_g for this work. In this section, we first describe the pipeline at a high level and then present the NLP toolkit, our various extraction and curation models, and methods used to prioritize human review.

A. Our Pipeline

Figure 1 illustrates our current pipeline with its six main stages. In stage 1, an extended version of a general-purpose materials NLP toolkit called ChemDataExtractor is used to extract a set of **T_g candidates** from text; in stage 2, compound names identified by the NLP Module are processed to create a polymer dictionary. As we describe below, the candidates identified in stage 1 can be in various forms: compound–T_g pairs; solitary T_gs, with no associated compound; and label–T_g pairs, in which the T_g is associated with a label rather than a compound. Each form requires further processing, which is performed in stage 3 via two automated curation modules and one crowdsourcing module. The results of those three modules are combined as the **proposed polymer–T_g pairs**. Stage 4 engages crowds in flagging erroneous results, stage 5 prioritizes final validation and curation of the proposed pairs, and stage 6 applies final expert review.

Designing a system to make use of crowds requires tailoring tasks to the expertise of the participants. For example, it is significantly easier for a nonexpert to mark a polymer–T_g pair as correct or incorrect than to extract the pair from a paragraph of text. Thus, we focus our crowdsourcing modules on simple micro-curation tasks. We have developed crowdsourcing modules to address two curation tasks: resolving labels that refer to polymer names and flagging anomalous polymer names.

The output of our pipeline is a set of **confirmed polymer–T_g pairs**, each associating a polymer name (with acronyms and/or synonyms) with a single T_g. These pairs are represented in a JSON format that can be easily processed and loaded into a database. Listing 1 shows an example record.

B. Natural Language Processing Module

The first phase of our pipeline requires the identification and extraction of structured representations of information embedded within text. There has been a wealth of research into creating specialized systems for extracting materials [11, 27]

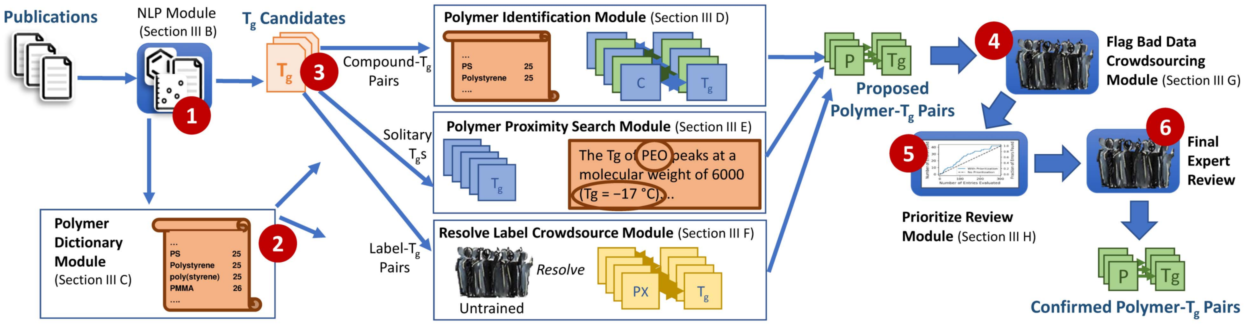

Publications
NLP Module (Section III B)
1
T_g Candidates
3
Compound-T_g Pairs
Solitary T_gs
Label-T_g Pairs
Polymer Dictionary Module (Section III C)
...
PS 25
Polystyrene 25
poly(styrene) 25
PMMA 26
....
2
Polymer Identification Module (Section III D)
...
PS 25
Polystyrene 25
....
C
T_g
Polymer Proximity Search Module (Section III E)
T_g
The Tg of PEO peaks at a molecular weight of 6000 (Tg = −17 °C)...
Resolve Label Crowdsource Module (Section III F)
Resolve
Untrained
PX
T_g
P
Tg
Proposed Polymer-T_g Pairs
4
Flag Bad Data Crowdsourcing Module (Section III G)
5
Prioritize Review Module (Section III H)
6
Final Expert Review
P
Tg
Confirmed Polymer-T_g Pairs

Fig. 1: The six-stage hybrid IE pipeline, showing (1) the NLP Module, which identifies T_g candidates; (2) the Polymer Dictionary Module, which identifies polymer names in NLP output; (3) the three automated extraction and crowdsourcing modules used to process different forms of candidates; (4) the Flag Bad Data Crowdsource Module, in which crowds flag anomalous results, (5) the Prioritize Review Module, which ranks extracted polymer–T_g pairs to prioritize expert validation, and (6) the Final Expert Review.

```
{
      "names": [
            "PBMA",
            "poly(butyl methacrylate)"
      ],
      "glass_transitions": [
            {
              "units": "°C",
              "value": "20"
            }
      ]
}
```

Listing 1: This polymer–T_g record indicates that the polymer *poly(butyl methacrylate)*, also known as *PBMA*, has a T_g of 20 °C.

and other domain-specific [14, 36–38] content from text. Thus, we choose to extend an existing NLP toolkit, ChemDataExtractor [12], to extract T_g values from documents.

1) ChemDataExtractor: ChemDataExtractor is a best-of-breed system for materials extraction, as evidenced by its performance in the relevant chemical compound and drug name recognition (CHEMDNER) community challenge [39]. It implements an extensible end-to-end text-mining pipeline that can process common publication formats including Portable Document Format (PDF), HyperText Markup Language (HTML), and eXtensible Markup Language (XML); it also supports extraction from headings, paragraphs, and captions, and produces machine-readable structured output data that can be used for subsequent processing. ChemDataExtractor automatically extracts chemical named entities and their associated properties, measurements, and relationships from scientific documents. It uses a combination of machine learning (linear-chain conditional random field) models, dictionary-based approaches, and regular expressions for entity recognition. It also detects and associates acronyms and synonyms with polymer names. Entity properties are extracted using a rule-based approach customized for specific properties. Extractors are provided for properties such as melting point and spectrum types, but not T_g.

2) Extending ChemDataExtractor for Glass Transition Temperatures: Our T_g extraction module incorporates specialized knowledge about the forms in which T_g values are expressed in scientific articles. Adapting the format of ChemDataExtractor's melting point extractor, our module contains rules that detect a prefix for a temperature (e.g., "a glass transition temperature of") and then detect and extract the associated temperature (e.g., "20 °C"). ChemDataExtractor then links these values with the associated compound(s). Of course, T_gs are expressed in many formats and therefore our rules must include variations of such statement structures. For instance, we include rules that match various quantifiers, such as "a glass transition temperature **range** of." Similarly, our rules capture approximate values, where temperatures are preceded by terms such as **ca.** or **around**. Further, our rules support variations of glass transition temperature including T_g, glass transition temp. and more. In total, we defined two dozen rules to address different variations and representations of glass transition temperature. Our T_g extractor has since been integrated into ChemDataExtractor.

The output of our extended ChemDataExtractor is a set of JSON records, each containing one or more T_g values and, optionally, an associated chemical compound name plus any automatically-detected acronyms and synonyms.

C. Polymer Dictionary Module

The materials literature includes references to a wide range of compounds beyond just polymers. The original ChemDataExtractor does not distinguish polymers from non-polymers and thus we face the challenge of correctly identifying which chemical name entities in a paper correspond to polymers. Unfortunately, no complete dictionary for polymer names exists and the standardized International Union of Pure and Applied Chemistry (IUPAC) naming conventions [40] often result in lengthy and, hence, rarely used names. Thus polymers are expressed using a combination of common names, IUPAC names, and trade names. The polymer identification problem is further complicated by the fact that values are often reported for *copolymers*, in which two or more monomers are used during synthesis.

The Polymer Dictionary Module implements heuristics for identifying those compound names extracted by stage 1 that likely correspond to polymers, and collects the resulting names in a **polymer dictionary**. These heuristics include rules related to text-based names (e.g., prefixes of "P" and "poly") as well as rules prescribed by the IUPAC guide [41] for forming polymer names. The latter is valuable for identifying copolymers. For example, names containing the substring "*-alt-*" indicate copolymers comprising two species of monomeric units in alternating sequence.

This module also handles synonyms and acronyms, a common occurrence in polymer science. For example, we may find the polymer *Polystyrene* represented in the same or different articles by the synonym *poly(styrene)* or the acronym *PS*. ChemDataExtractor includes mechanisms for identifying and grouping synonyms and acronyms. We record these groups in our polymer dictionary. We also include both singular and plural representations, for example *polystyrene* and *polystyrenes*. To avoid confusion with acronyms, we only consider plurals for names longer than four characters. Thus, for example, *PSS*, the acronym for *poly(styrene sulfonate)*, is not identified as the plural form of *PS*. We exclude copolymers from our dictionary as these are easily recognizable via our implemented IUPAC polymer heuristics.

To bootstrap the polymer dictionary, we ran our polymer identification heuristics over all 6 090 full-text HTML publications from *Macromolecules* and thereby populated the dictionary with 12 814 polymer names and acronyms in 9 178 different detected groups.

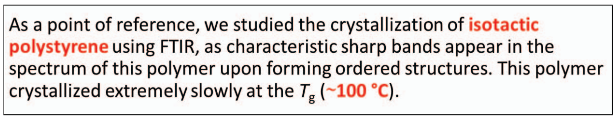
D. Polymer Identification Module

For cases where compound-T_g pairs were idenified using the NLP Module, the Polymer Idenification Module determines which of those compounds are polymers and which are not. To do so, the module simply labels any compound present in the polymer dictionary produced by the Polymer Dictionary Module as a polymer and all other entires as non-polymers.

E. Polymer Proximity Search Module

One significant type of error for text extraction are T_g values that are not associated with a polymer name. To correct these errors, we have developed a proximity-based approach for determining whether the polymer name is mentioned nearby where the temperature was found (for example, in the previous sentence or paragraph). For each sentence in the document, we determine whether it contains a T_g value, and if so, return the closest polymer name (using the polymer dictionary) within the sentence, if any such name is to be found. If no polymer is found, we extend the search to the preceding sentence, as illustrated in Figure 2.

This process increases the number of polymer–T_g pairs discovered; however, it may decrease the accuracy of the extracted pairs. We discuss validation in Section IV-G.



As a point of reference, we studied the crystallization of isotactic polystyrene using FTIR, as characteristic sharp bands appear in the spectrum of this polymer upon forming ordered structures. This polymer crystallized extremely slowly at the T_g (~100 °C).

Fig. 2: The NLP Module yields a solitary T_g record in this example text [42], as the corresponding compound is mentioned in the previous sentence. The Polymer Proximity Search Module disambiguates the reference and proposes *isotactic polystyrene* as a (correct) candidate match for the T_g.

F. Resolve Label Crowdsource Module

This first crowdsource module addresses errors where the text extraction matched T_g values to labels (e.g., *Polymer A*) rather than the actual polymer name. These labels frequently occur in the polymer literature to avoid repetition of complex polymer names, such as the following.

```
poly(1,2:3,4-di-O-isopropylidene-6-O-(2′—formyl-4′-
    vinylphenyl)-d-galactopyranose)
```

We created an interface that presents labels and the paper in which each appears, and asks the crowd to enter the polymer name for each label. As this task requires little knowledge of polymer science, we use an untrained crowd to resolve references. We provide these people with just a simple training guide (less than one page) to describe the task. In an attempt to quantify accuracy, we allow crowd members to specify their confidence (1–5) along with their input. Our goal is to use this confidence score to prioritize results for future review.

G. Flag Bad Data Crowdsource Module

The second crowdsource module presents users with a list of polymer–T_g pairs and asks them to flag whether the polymer names are incomplete or incorrect. The polymer names identified by the text extraction tool are sometimes not specific enough to identify the polymer being studied. As one example, the term "*hydroxyl copolyimides*" describes a family of polymers rather than a specific polymer, and therefore cannot be attributed a single T_g value. Given the complexity we use an expert crowd of polymer scientists to complete this task. Our flagging interface does not delete any data from our set, but rather records user "votes." We then use this information to prioritize further review.

H. Prioritize Review Module

Every stage of the pipeline uses a variety of methods to extract values with varying confidence. Thus, each proposed polymer–T_g pair has an associated probability of accuracy. For example, a pair extracted from a single sentence using our NLP rules and subsequently reviewed by an expert is likely to be accurate. In contrast, a pair in which the polymer name is a synonym, was found in the sentence preceding that containing the T_g value, and was not reviewed by a human, is less likely to be accurate. To formalize this concept, we explore methods for estimating confidence in a particular value and use this metric to prioritize (crowdsourced) curation tasks.

Our initial approach for the prioritization method relies on characteristics of polymer names and their associated T_g values. Hypothesizing that polymer names that appear more frequently in the database have a higher likelihood of being correct than infrequently used names, we assign a confidence to each polymer name based on its frequency of occurence. Further hypothesizing that outlier or extreme temperatures are more likely to indicate errors, we determine the minimum, mean, and maximum of all T_g values in our current database and use those values to identify outliers, to which we assign lower confidence values. These two scoring methods can be combined. For example, if two records appear equally infrequently, we prioritize for review the one with temperature farthest from the mean. Entries with confidence scores under a fixed threshold will then be funnelled to Stage 6 of the pipeline for expert review as shown in Figure 1.

V. Evaluation

We quantitatively evaluated our pipeline by comparing results against a gold standard, human-reviewed dataset. In this section, we describe our input dataset and then present our evaluation of each module in our pipeline.

A. Dataset

Our **input dataset** comprised of 6 090 publications in full-text HTML format. To obtain these publications, we automatically searched the journal *Macromolecules* using the keyword "T_g" over the ten-year period 2006–2016. We downloaded the full-text publications matching this query and sampled additional *Macromolecules* issues from the last decade to increase and diversify our corpus. This is the same dataset that we used to build our polymer dictionary, as described previously.

B. Natural Language Processing Module

Execution of the T_g–extended ChemDataExtractor NLP module described in Section IV-B identified 364 561 records, of which 1 330 were candidate T_g values from 927 distinct publications: 846 compound–T_g pairs, 456 solitary T_gs, and 28 label–T_g pairs. (Another 112 linked more than one compound and/or T_g value, a case that we leave for future work.) We stored these records in a database for convenient access to their features, which include the name of the associated compound, when present, and any synonyms for that compound.

C. Assembling a Gold Standard Dataset

We manually selected a subset of 50 papers for which the NLP module had identified one compound–T_g pair for which the compound contained the string "poly." We then had two polymer scientists each read 25 of these publications to identify all polymer–T_g pairs that they contain. The result is a gold standard dataset containing a total of 62 polymer–T_g pairs. We used this dataset for various evaluation steps.

To gain some initial experience with the use of this dataset, we also asked our experts to evaluate the accuracy of the 50 compound–T_g pairs identified in these papers by the NLP module. In evaluating precision, we assigned points to each extracted entry as follows: 1 point for **fully correct** entries, i.e., entries that were completely unambiguous and correct; 0.5 points to **partially correct** entries, in which information was missing (e.g., the module extracted *polyurethanes.11*, a correct but idiosyncratic name, which an expert clarified by adding *polyurethanes with various side chains*); and 0 points to other **incorrect** cases, such as those with an incomplete polymer name (e.g., the module extracted *hydroxyl copolyimides* instead of *APAF-ODA hydroxyl copolyimides*: the former describes a vast family of polymers and cannot be clarified without additional information).

The NLP module extracted 17 fully correct and 4 partially correct polymer–T_g pairs from the 50 articles, for a precision of 38 %. As our experts identified 62 T_g values in the 50 articles, the recall was 31 %. While the expert reviews, being aimed at assembling a gold standard, were particularly rigorous, these low values emphasize the difficulty of our task and the need for a hybrid solution. In most cases, errors were related to identification of the polymer name rather than the T_g value. In fact, for the subproblem of locating T_g values, our T_g extraction rule achieved 88 % precision (44 out of 50 cases) and 71 % recall (18 T_g values missed out of 62 total).

Precision: We attribute our low precision to three main reasons. A first is that the compound name was incorrectly or partially identified ≈50 % of the time. The low performance in polymer name recognition may be explained by the fact that the entity recognition component of ChemDataExtractor was trained on biomedical newspaper and biomedical training corpora, supplemented with unsupervised word cluster features derived from chemistry articles. The use of biomedical training data is due to the lack of appropriate annotated corpora for training machine learning models for polymer name recognition, a general problem in materials informatics. Moreover, our experts noted that some polymer names were difficult even for humans to extract, as they were not named but rather described in terms of their components: e.g., "*A cross-linked polymer with DABBF linkages was prepared by polyaddition of poly(propylene glycol) (PPG) (Mn = 2700), hexamethylene diisocyanate (HDI), dihydric DABBF, and triethanolamine (TEA) as a cross-linker in the presence of di-n-butyltin dilaurate (DBTDL, catalyst) in N,N-dimethylformamide (DMF) in a manner similar to that previously reported (Figure 1)*" [43].

A second difficulty, which arose in 8 % of the cases, was that one of our T_g extraction rules was loosely defined as simply "transition," to avoid tokenizing issues around the term "glass-transition." We expected that in the context of polymer science the most common transition temperature would be T_g. However, while this rule sometimes functioned as expected, it also matched sentences with "gel transition" and "phase transition" temperatures. We could redefine the loose transition rule, but while this would increase precision, it would also decrease recall. Initially, we view high recall as a preferable to high precision in our "big-data" approach, as we expect later pipeline stages to improve the precision.

A third difficulty, arising in 4 % of the cases, was that complex sentence structure led to incorrect T_g values being extracted. For example, in sentences describing increases or decreases in temperature relative to a previously mentioned value, the software identified the difference as T_g: e.g., "*Comparing DSC results for dried composites (Figure 3b), a drop in Tg of 17 °C was observed for the clay composite, whereas the corresponding drop in Tg of the aerogel composite was only 3 °C*" [44]. One way to improve precision in such cases would be to analyze sentence complexity, as indicated by features such as number of words and the use of comparison terms such as "lower/greater" and "decrease/increase," and then defer to trained crowds for sentences above a certain threshold.

Recall: We view improving recall as an iterative process as we continue to find additional ways that T_g is expressed in the literature. During the evaluation of the NLP module, we inspected the results and added new rules to our T_g extractor to increase recall. For example, sometimes authors referred to the *"T_g value of"*; the extra *"value"* term was not included in the original parser. Another slightly more complex example consists of capturing a temperature expressed in the form *"T_g of* ***<polymer name>*** *is/was ..."*. This rule depends on correctly identifying the polymer name in the sentence, as some polymer names, which sometimes include dashes, spaces, and colons, will not always correspond to the regular expression class of words. We plan to use a larger dataset for evaluation, examine more cases of missed T_gs and identify new general rules that will further improve recall.

D. Polymer Identification Module

To test the polymer name classifier described in Section IV-D, we selected 100 papers: the 50 used in Section V-B plus 50 additional papers with compound–T_g records for which the compound names did not include "poly."

Using our full polymer name dictionary (prefixes and IUPAC guidelines as well as simple "poly" keyword search), we classified the compounds from the 100 papers. We achieved 91.8 % precision and 93.2 % recall. In other words, we correctly classified 91.8 % of the compounds as polymers and misclassified 6.8 % of the extracted compounds. An example of a false positive is identifying a class of polymers (e.g., *polyimides*) rather than an individual polymer. An example of a false negative is the copolymer *UPy-OPG-MAA*: as none of its three components existed in the polymer name dictionary, our heuristics could not identify it as a copolymer. The addition of polymer heuristics improved the performance of our polymer classification by correctly discovering additional polymers (16 % of the compounds initially classified as "non-polymers"), which were not detected by a simple string search of the names, hence potentially increasing the number of polymer–T_g pairs in the final output. They are particularly useful for detecting copolymers using IUPAC conventions (e.g., *PPDL-block-PLLA*) formed of previously seen polymer components (e.g., *PPDL* or *PLLA*). We have since composed a list of common polymer families that will further improve our classification results.

E. Polymer Proximity Search Module

Recall from Section IV-E that this module seeks to address the problem of T_g values that were extracted without a polymer name. To test this module, we first identified 115 records containing solitary T_g values. The module returned a polymer for 74 out of these 115 records (64.3 %). We executed the proximity search heuristic to consider the same and previous sentences and compared the identified polymer names to those identified by an expert. Our proximity search suggested correct polymer matches for 31 of the 63 records (49.2 %) in which the matching polymer was located within the same sentence. Its search of the preceding sentence identified correct polymer matches in 6 of the 11 records (54.5 %) in which the matching polymer was in that sentence. Together, searching both the T_g and preceding sentence led to the recovery of 37 polymers: 50.0 % of the original 74 solitary T_g mentions or 32.1 % of the test dataset, which includes false negatives. See Table I for a summary of the results.

TABLE I: Polymer proximity search module evaluation.

	True Positives	False Positives	Gold
Same sentence	31	32	63
Previous sentence	6	5	11
No candidate returned			41
Total	37	37	115

We note that success here requires correct identification of both the polymer and the temperature to be linked. Some compounds were only partially identified and the complete polymer–T_g pairs were not correctly recovered. Since the proximity search module uses the polymer database, improving polymer name recognition and the T_g parser will in turn increase proximity search performance. In some cases, proximity search introduced false positives for a different reason, as the compound closest to the temperature was used for comparison and was not associated with the extracted T_g for instance. Nevertheless, confirming or rejecting matches from this module is a less difficult task than extracting the polymer–T_g pairs.

F. Crowdsourcing Modules

Recall from Sections IV-F and IV-G that we have deployed two crowdsourcing modules: one to recover polymer names from author-defined labels and one to flag polymer–T_g pairs deemed to require further review.

In the first case, we presented three (non-expert) reviewers with polymer name labels and asked them to extract the polymer name from the full text. We also asked them to state their confidence (1-5). We identified 28 records for review (based on regular expression matching of the form "Polymer *[a-zA-Z0-9]*"). The three reviewers correctly identified 82.1 %, 78.6 %, and 35.7 % of those 28 records and reported an average of two hours of work. A simple consensus method across our three reviewers (selecting the answer from two or more reviewers in agreement) obtained 78.6 % accuracy when resolving these labels. Only in two cases did no reviewer identify the correct label, seemingly indicating that this task was at an appropriate level of difficulty for our crowd. These results show that the use of untrained crowds can reduce the need for expert validation substantially. Table II summarizes reviewer performance and confidence scores. It shows the number of correct answers from reviewers with their reported confidence scores. Reviewer 2 correctly identified 21/21 labels with high confidence, 2/3 with medium confidence and 0/4 with low confidence.

TABLE II: Crowdsourcing for resolving polymer labels.

		Confidence (correct/total)			Time
	Correct	**High (1-2)**	**Med (3)**	**Low (4-5)**	**spent (hours)**
Reviewer 1	23	23/28	0	0	3
Reviewer 2	23	21/21	2/3	0/4	2
Reviewer 3	10	0	0	10/28	1

In the second crowdsourcing task, we presented an expert polymer scientist with 302 compound–T_g pairs extracted by the NLP module for which the compound matched the string "poly." The reviewer took about 30 minutes to identify 43 (14 %) of these values as incomplete or incorrect, leaving the 259 confirmed polymer–T_g pairs noted in the abstract. Erroneous values included names that describe a class of polymers as opposed to a specific polymer (e.g., *polyolefin*) and unrecognized labels (e.g., *copolymer 10*), and additional descriptors (e.g., *macroporous poly(N-isopropylacrylamide) gel*). Overall, these results suggest that our extractor performs as expected in the majority of cases.

We will next apply this same process to the additional proposed polymer–T_g pairs produced by our system. In addition to improving our dictionary, we are currently compiling a list of common polymer family names and working on a list of common descriptors to ignore.

G. Prioritizing Review

We applied our scoring model (using polymer name frequency and T_g value distance from the median) to 302 compound–T_g pairs for which the compound name matched the string "poly." We compared the pairs prioritized by the scoring model against those flagged by experts in the previous crowdsourcing step. After ordering these pairs by confidence, we observed that 10 of the first 50 entries had been flagged as erroneous by our reviewers (see Figure 3), which is 40% more than would be expected if entries were randomly selected (≈7 errors). While not an extraordinary decrease in the number of reviews, it was achieved by a basic ranking scheme; we expect more sophisticated approaches to further reduce the human effort required to improve the quality of our database.

We plan to use a similar scheme to score entries in the polymer dictionary. The scheme will consider frequency, number of synonyms, and number of duplicate entries (same acronym for different polymers) to assign a confidence score to each entry. We anticipate that this approach will be able to detect additional unrecognized polymers: for example, *poly(2,4′-BFa)* where author-defined monomer *2,4′-BF-a* is specified elsewhere in the publication.

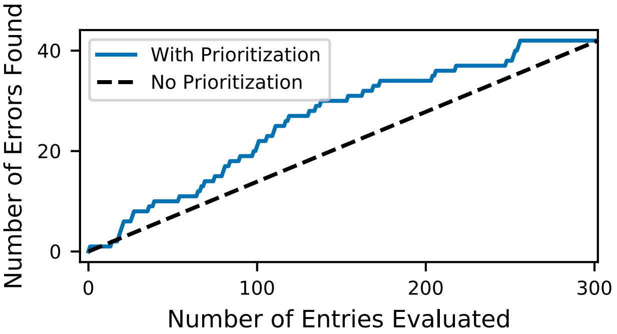


Fig. 3: Results of prioritizing crowdsourcing. The blue, solid line shows the number of errors found as a function of the number of expert reviews if the entries are evaluated following our prioritization scheme. The black, dotted line shows the number of errors found if entries are evaluated in a random order.

H. Summary of Results

Table III aggregates the results of our evaluation across the four types of T_g candidates that we have examined. The **Initial** column gives the number of each type extracted from our 6 090 articles, with poly–T_g here denoting compound–T_g pairs for which the compound name contains the string "poly" and nonpoly–T_g the remaining compound–T_g pairs. The **Yield** column indicates the number of T_g candidates of each type that are estimated to be correct, based on review. (For polymer–T_g and label–T_g, this is a full review; for compound–T_g and solitary T_g, the numbers are estimates based on expert review of a subset.) The **Pairs** column gives the number of polymer–T_g pairs that we expect from each method. Thus, we expect the final number of pairs extracted from our initial set of 6 090 articles to increase significantly—perhaps by 145 % to approximately 500—once we complete expert review.

TABLE III: Summary of module performance and expected number of polymer–T_g output from initial data.

Input Type	Initial	Module	Yield	Pairs
poly–T_g	302	Flag Bad Data	86.0 %	259
nonpoly–T_g	544	Polymer Identification	16.0 %	87
solitary T_g	456	Proximity Search	32.1 %	146
label–T_g	28	Resolve Labels	78.6 %	22
Totals	**1 330**			**514**

VI. Future Work

While our pipeline initially focuses on extracting polymer–T_g pairs, our approaches are equally applicable to other properties and forms of data.

Polymer properties, such as T_g, are often dependent on important contextual information, such as molecular mass and geometry (confined or bulk) as well as the experimental methods used to calculate values. We intend to develop methods to capture such information to provide context to extracted values. As previously mentioned, we also plan to make improvements to the dictionary and evaluate its accuracy using experts.

Given the significant cost of manual curation, we are also investigating more advanced methods to prioritize where human effort should be used. Here we discuss two ideas relevant to this topic: validation of extracted data via machine learning models and experiment design.

A. Machine Learning Validation

The T_g of amorphous polymers is the most important and widely studied polymeric property because many other polymer properties, such as heat capacity and viscosity, are affected by this transition [45]. Many researchers have developed machine learning models for T_g [45–48] that we could retrain, using the entries in our database, to make predictions that would in turn validate extracted values. These T_g models provide rough estimates or reasonable ranges for the T_g values of various polymers, which would serve as physics-based validation of our extracted values and help prioritize curation.

B. Experiment Design

A major challenge with a hybrid pipeline is determining when to employ human expertise and, when human expertise is needed, what form of expertise to apply. While we work towards higher levels of accuracy from our automated modules, we do not expect the need for human input to disappear. We have explored several methods including expert review, untrained confidence scores, and a scoring mechanism for prioritization; however, none is without limitation. As the number of publications processed by our pipeline increases, this careful scrutiny of the data will become costly and eventually unworkable. We want to identify when and how to inject different types of human input into the pipeline efficiently. In other words, we want to increase accuracy while minimizing the quantity and cost of crowd input.

We plan to explore a more rigorous approach to automatic partitioning and assignment of extraction tasks by applying techniques from *optimal experiment design* [49–51] to maximize the accuracy of extracted data while minimizing the time and cost of human involvement. To this end, we expect to:

- Calculate the accuracy of values derived from a variety of automated and crowdsourcing modules.
- Assign values to datasets, for example in terms of their yield in polymer–T_g pairs and/or the rarity of those values, and then measure how dataset value changes with each automated and crowdsourced task.
- Assign levels of difficulty to tasks based on completeness and accuracy of the data to be processed and/or the information needed to complete the task, to help decide where to crowdsource various tasks.
- Assign costs to module usage so that we can compare, for instance, the costs of computational vs. crowdsourced modules; determine the cost of using crowds (e.g., person-hours); and quantify the differences in cost between a trained and untrained crowd.

We plan to investigate these topics as we develop the next generation of our pipeline.

VII. Conclusion

Despite significant progress in natural language processing and machine learning approaches to information extraction, there remains a gap between the current data extraction needs in fields such as materials science and the capabilities of state-of-the-art tools. We have described a hybrid human-machine IE pipeline that we have so far used to extract 259 glass transition temperature (T_g) values for polymers from 6 090 scientific articles, with an expectation of many more as we improve our methods and process more articles.

Our pipeline uses domain-specific automated and crowdsourcing extraction and curation modules to extract high-quality and accurate polymer–T_g pairs. The polymer classifier module achieved 91.8 % precision and 93.2 % recall. The polymer proximity search module correctly identified missing polymers for 50.0 % of those T_g values without polymers. We crowdsourced the recovery of unrecognized polymer names for an additional 22 polymer–T_g pairs and demonstrated that using untrained crowds for simple, well-defined domain-specific tasks can decrease the need for expert validation by about three fourth (78.6 % labels resolved by non-experts using concensus method). We have started the validation of automatically extracted data and presented a simple scoring scheme to prioritize the process. Our initial results show that even a simple method for assessing the quality of extracted data can effectively increase the impact of human curation.

While the size of our T_g database is not yet best-in-class, the hybrid pipeline presented in this work offers a sustainable and accelerated route to producing new materials property datasets. With only a few hours of effort from expert and non-expert curators, we were able to screen over 6 000 articles and produce a refined dataset of 259 polymer–T_g pairs from just 927 articles. Thus, our results demonstrate the considerable potential of combining automated and crowdsourcing modules to extract scientific facts from literature in an efficient and cost-effective manner. We continue to refine our automated extraction tools and develop yet more effective ways of prioritizing human curation for maximum benefit, and to use these tools to populate our open database. Our verified polymer–T_g pairs are available at both http://pppdb.uchicago.edu and https://materialsdatafacility.org.

Acknowledgments

We thank our crowd members for their help. This work was supported in part by NIST contract 60NANB15D077, the Center for Hierarchical Materials Design, and DOE contract DE-AC02-06CH11357. Official contribution of the National Institute of Standards and Technology; not subject to copyright in the United States.

References

[1] N. Nosengo, “Can artificial intelligence create the next wonder material?” *Nature*, vol. 533, no. 7601, pp. 22–25, may 2016. [Online]. Available: http://www.nature.com/doifinder/10.1038/533022a

[2] J. Hill, G. Mulholland *et al.*, "Materials science with large-scale data and informatics: Unlocking new opportunities," *MRS Bulletin*, vol. 41, no. 05, pp. 399–409, 2016. [Online]. Available: http://www.journals.cambridge.org/abstract_S0883769416000932

[3] J. J. de Pablo, B. Jones *et al.*, "The Materials Genome Initiative, the interplay of experiment, theory and computation," *Current Opinion in Solid State and Materials Science*, vol. 18, no. 2, pp. 99–117, 2014.

[4] K. M. Tolle, D. S. W. Tansley *et al.*, "The fourth paradigm: Data-intensive scientific discovery," *Proceedings of the IEEE*, vol. 99, no. 8, pp. 1334–1337, Aug 2011.

[5] A. Agrawal and A. Choudhary, "Perspective: Materials informatics and big data: Realization of the "fourth paradigm" of science in materials science," *APL Materials*, vol. 4, no. 5, p. 053208, 2016.

[6] S. Otsuka, I. Kuwajima *et al.*, "PoLyInfo: Polymer database for polymeric materials design," in *International Conference on Emerging Intelligent Data and Web Technologies*. IEEE, 2011, pp. 22–29.

[7] S. Bird, E. Klein *et al.*, *Natural language processing with Python: Analyzing text with the natural language toolkit*. O'Reilly Media, Inc., 2009.

[8] S. Hellmann, J. Lehmann *et al.*, "Integrating NLP using linked data," in *International Semantic Web Conference*. Springer, 2013, pp. 98–113.

[9] C. D. Manning, M. Surdeanu *et al.*, "The Stanford CoreNLP natural language processing toolkit," in *ACL (System Demonstrations)*, 2014, pp. 55–60.

[10] N. A. Lewinski and B. T. McInnes, "Using natural language processing techniques to inform research on nanotechnology," *Beilstein Journal of Nanotechnology*, vol. 6, no. 1, pp. 1439–1449, 2015.

[11] L. Hawizy, D. M. Jessop *et al.*, "ChemicalTagger: A tool for semantic text-mining in chemistry," *Journal of Cheminformatics*, vol. 3, no. 1, p. 17, 2011.

[12] M. C. Swain and J. M. Cole, "ChemDataExtractor: A toolkit for automated extraction of chemical information from the scientific literature," *Journal of Chemical Information and Modeling*, vol. 56, no. 10, pp. 1894–1904, 2016.

[13] C. De Sa, A. Ratner *et al.*, "DeepDive: Declarative knowledge base construction," *ACM SIGMOD Record*, vol. 45, no. 1, pp. 60–67, 2016.

[14] A. Rzhetsky, I. Iossifov *et al.*, "GeneWays: A system for extracting, analyzing, visualizing, and integrating molecular pathway data," *Journal of Biomedical Informatics*, vol. 37, no. 1, pp. 43–53, 2004.

[15] C. Seifert, M. Granitzer *et al.*, "Crowdsourcing fact extraction from scientific literature," in *Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data*. Springer, 2013, pp. 160–172.

[16] J. Takis, A. Islam *et al.*, "Crowdsourced semantic annotation of scientific publications and tabular data in PDF," in *11th International Conference on Semantic Systems*. ACM, 2015, pp. 1–8.

[17] R. B. Tchoua, J. Qin *et al.*, "Blending education and polymer science: Semiautomated creation of a thermodynamic property database," *Journal of Chemical Education*, vol. 93, no. 9, pp. 1561–1568, 2016.

[18] R. B. Tchoua, K. Chard *et al.*, "A hybrid human-computer approach to the extraction of scientific facts from the literature," *Procedia Computer Science*, vol. 80, pp. 386–397, 2016.

[19] H. B. Eitouni and N. P. Balsara, "Thermodynamics of polymer blends," in *Physical Properties of Polymers Handbook*. Springer, 2007, pp. 339–356.

[20] B. Blaiszik, K. Chard *et al.*, "The Materials Data Facility: Data services to advance materials science research," *JOM*, vol. 68, no. 8, pp. 2045–2052, 2016.

[21] C. Friedman, P. O. Alderson *et al.*, "A general natural-language text processor for clinical radiology," *Journal of the American Medical Informatics Association*, vol. 1, no. 2, pp. 161–174, 1994.

[22] C. Friedman, G. Hripcsak *et al.*, "Representing information in patient reports using natural language processing and the extensible markup language," *Journal of the American Medical Informatics Association*, vol. 6, no. 1, pp. 76–87, 1999.

[23] G. K. Savova, J. J. Masanz *et al.*, "Mayo clinical text analysis and knowledge extraction system (cTAKES): Architecture, component evaluation and applications," *Journal of the American Medical Informatics Association*, vol. 17, no. 5, pp. 507–513, 2010.

[24] "medkat," http://ohnlp.sourceforge.net/MedKATp, accessed Sep, 2017.

[25] C. Friedman, P. Kra *et al.*, "GENIES: A natural-language processing system for the extraction of molecular pathways from journal articles," in *ISMB (supplement of bioinformatics)*, 2001, pp. 74–82.

[26] L. Chen and C. Friedman, "Extracting phenotypic information from the literature via natural language processing," *Studies in Health Technology and Informatics*, vol. 107, no. 2, pp. 758–762, 2004.

[27] D. M. Jessop, S. E. Adams *et al.*, "OSCAR4: A flexible architecture for chemical text-mining," *Journal of Cheminformatics*, vol. 3, no. 1, p. 41, 2011.

[28] S. E. Peters, C. Zhang *et al.*, "A machine reading system for assembling synthetic paleontological databases," *PLoS One*, vol. 9, no. 12, p. e113523, 2014.

[29] C. Zhang, V. Govindaraju *et al.*, "GeoDeepDive: Statistical inference using familiar data-processing languages," in *ACM SIGMOD International Conference on Management of Data*, pp. 993–996.

[30] "Paleodb," http://paleodb.org, accessed Sep, 2017.

[31] "Macrostrat," http://macrostrat.org, accessed Sep, 2017.

[32] M. J. Franklin, D. Kossmann *et al.*, "CrowdDB: Answering queries with crowdsourcing," in *ACM SIGMOD International Conference on Management of Data*, 2011, pp. 61–72.

[33] B. C. Wallace, A. Noel-Storr *et al.*, "Identifying reports of randomized controlled trials (RCTs) via a hybrid machine learning and crowdsourcing approach," *Journal of the American Medical Informatics Association*, 2017.

[34] J. Brandrup, E. H. Immergut *et al.*, Eds., *Polymer Handbook*, 4th ed. Wiley-Interscience, 1999.

[35] L. Chiticariu, Y. Li *et al.*, "Rule-based information extraction is dead! Long live rule-based information extraction systems!" in *Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 827–832.

[36] D. Zhou and Y. He, "Extracting interactions between proteins from the literature," *Journal of Biomedical Informatics*, vol. 41, no. 2, pp. 393–407, 2008.

[37] F. Rinaldi, G. Schneider *et al.*, "Mining of relations between proteins over biomedical scientific literature using a deep-linguistic approach," *Artificial Intelligence in Medicine*, vol. 39, no. 2, pp. 127–136, 2007.

[38] M. Krauthammer and G. Nenadic, "Term identification in the biomedical literature," *Journal of Biomedical Informatics*, vol. 37, no. 6, pp. 512–526, 2004.

[39] M. Krallinger, F. Leitner *et al.*, "CHEMDNER: The drugs and chemical names extraction challenge," *Journal of Cheminformatics*, vol. 7, no. 1, p. S1, 2015.

[40] R. G. Jones, E. S. Wilks *et al.*, Eds., *Compendium of Polymer Terminology and Nomenclature*. The Royal Society of Chemistry, 2009. [Online]. Available: http://dx.doi.org/10.1039/9781847559425

[41] R. C. Hiorns, R. J. Boucher *et al.*, "A brief guide to polymer nomenclature," *Polymer*, vol. 54, no. 1, pp. 3–4, 2013.

[42] J. Mattia and P. Painter, "A comparison of hydrogen bonding and order in a polyurethane and poly (urethane- urea) and their blends with poly (ethylene glycol)," *Macromolecules*, vol. 40, no. 5, pp. 1546–1554, 2007.

[43] K. Imato, A. Takahara *et al.*, "Self-healing of a cross-linked polymer with dynamic covalent linkages at mild temperature and evaluation at macroscopic and molecular levels," *Macromolecules*, vol. 48, no. 16, pp. 5632–5639, 2015.

[44] S. Bandi and D. A. Schiraldi, "Glass transition behavior of clay aerogel/poly (vinyl alcohol) composites," *Macromolecules*, vol. 39, no. 19, pp. 6537–6545, 2006.

[45] B. E. Mattioni and P. C. Jurs, "Prediction of glass transition temperatures from monomer and repeat unit structure using computational neural networks," *Journal of Chemical Information and Computer Sciences*, vol. 42, no. 2, pp. 232–240, 2002.

[46] A. DiBenedetto, "Prediction of the glass transition temperature of polymers: A model based on the principle of corresponding states," *Journal of Polymer Science Part B: Polymer Physics*, vol. 25, no. 9, pp. 1949–1969, 1987.

[47] T. Le, V. C. Epa *et al.*, "Quantitative structure-property relationship modeling of diverse materials properties," *Chemical Reviews*, vol. 112, no. 5, pp. 2889–2919, may 2012. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/cr200066h

[48] X. Yu, "Support vector machine-based QSPR for the prediction of glass transition temperatures of polymers," *Fibers and Polymers*, vol. 11, no. 5, pp. 757–766, 2010.

[49] V. V. Fedorov, *Theory of optimal experiments*. Elsevier, 1972.

[50] A. F. Emery and A. V. Nenarokomov, "Optimal experiment design," *Measurement Science and Technology*, vol. 9, no. 6, p. 864, 1998.

[51] V. Fedorov, "Optimal experimental design," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 5, pp. 581–589, 2010.