THE UNIVERSITY OF CHICAGO

HYBRID HUMAN-MACHINE SCIENTIFIC INFORMATION EXTRACTION

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

BY

ROSELYNE BARRETO TCHOUA

CHICAGO, ILLINOIS
AUGUST 2019

For Indira

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

It has been a beautiful journey, both exciting and challenging, also absolutely worth it. I feel a deep sense of gratitude towards the many people that have made up my indispensable support system and allowed me to grow personally and professionally through the years.

I am grateful for my advisor Ian Foster. I have learned so much from you. You have been an amazing professional mentor and also a strong supporter of me as a graduate student mom. Without your complete support, this work would not have been possible. Kyle Chard! You have accompanied and guided me along the sinuous roads of my research and life experience in the past few years. Words can hardly express how supportive you have been towards my career and well-being. I sincerely thank you both for strengthening my research skills, allowing this thesis to be my work, while steering me in the right direction.

I thank my supportive group members: Tyler Sluzacek, Sam Nickolai, Ricardo Barros Lourenco, Zhi Hong, Yuliana Zamora, Takuya Kurihana, Marcus Schwarting, Anna Woodard, Zhuozhao Li, Ryan Chard, Logan Ward, Ben Blaisik, Yadu Nand and Aswathy Ajith. I especially thank you, Tyler, Hong and Sam for contributing to my human information extraction tasks; Logan for your invaluable contributions to discussions and papers.

Thank you, Rick Stevens and Aaron Elmore, for being part of my committee, providing insightful comments about my work, and suggesting ideas for successful completion of my thesis.

I thank Juan de Pablo for supporting and contributing ideas to my research. I also want to express my gratitude for his collaborators at the Institute of Molecular Engineering and the National Institute of Standards and Technology. More specifically, thank you to Jian Qin, Debra Audus, and Shrayesh Patel. I would also like to acknowledge Pr. de Pablo and Pr. Patel's students, especially those who participated in the Materials Database Creation for Macromolecules (MENG 21500) course over the spring quarter of 2015. The work presented in my thesis is inter-disciplinary and your contributions have been crucial.

I also would like to thank the members of the Knowledge Lab who were always willing

to discuss and provide insight into the processes by which knowledge is embedded in and extracted from literature; specifically, Eamon Duede and Sasha Belikov.

Thinking of those who encouraged me to come back to school, I am grateful for Ricky Kendall, Scott Klasky and Bronson Messer, former colleagues at Oak Ridge National Laboratory for helping me take the first steps. Jay Lofstead, you have not only encouraged me in the initial decision but provided friendship and mentoring any time I reached out. Thank you.

In the Computer Science Department, there are many faces and supporting voices I will miss, including the technical and administrative staff, especially Bob Barlett and Margaret Jaffey; many of the faculty, especially Pr. Anne Rogers.

My friends who started with me six years ago, thank you for sharing this experience with me. My friends in Hyde Park and beyond, in particular Paola Tai, Giampierro Sciutto, Samba Gaye, thank you for enriching my social life. Hai Ah Nam, you and your family have become my family in the process of getting this degree. I cannot summarize in a few lines what your always-on-point advice have meant to me. Thank you.

Sophonie Tchoua, my dear husband. I have told you before, I could not have done this without you and our beautiful Indira. You, more than anyone, have shared this adventure with me. You are my strongest motivation. To my parents, Marie Madeleine Spencer, Philippe Jose Barreto, my siblings, Vanessa Barreto Ndong, Nilton Philippe Barreto and your families, my cousin Jean-Jacques de Oliveira: I have brought you all along this journey and you have supported me at every step. Thank you also to my family in law, especially Regine Tchoua for always checking on me. Mom, in additional to everything else you have given, you have been a true, inspiring role model. Knowing you had been there before, and gone on to an amazing career was vital to my experience. To all of you, your unconditional love is fuel to all my projects, notably this beautiful journey. I am forever grateful.

ABSTRACT

A wealth of valuable research data is locked within the millions of research articles published every year. Reading and extracting pertinent information from those articles has become an unmanageable task for scientists. Moreover, these data are loosely structured, encoded in manuscripts of various formats, embedded in different content types, and are, in general, not machine accessible. Thus, studies that automatically leverage this valuable information are not tractable or even possible. Current approaches employ humans to manually extract data, define extraction rules, or annotate training corpora for machine learning approaches through tedious, time-consuming, error-prone and sometimes expensive processes. In the specific case of scientific information extraction, the need for pointed expertise increases costs and decreases the generalization of extraction methods. This thesis seeks to demonstrate that efficient combination of human-computer extraction techniques can considerably alleviate the burden on human curators, thereby speeding up discovery of new scientific facts and decreasing extraction costs. This thesis is investigated in the context of materials informatics, an emerging field that has the potential to greatly reduce time-to-market and development costs for new materials. Such efforts rely on access to large databases of material properties and therefore represent a suitable but not unique application for this research. This work addresses the challenge of populating a database of scientific facts by presenting three approaches with different levels of automation and human involvement. Specifically, these three approaches involve varying amount of untrained, trained and expert input in order to populate a database of polymer properties. The first effort, χDB, engages a semi-expert crowd to extract an important relation in polymer science. Here automation is limited, being concerned only with identifying appropriate elements of scientific articles to present to crowd members. However, the approach is shown to accelerate data extraction speed considerably. χDB is a crowdsourcing system, which employs and assists a semi-expert crowd to extract an important relation in polymer science. Increasing the automation and targeting a different relation, the T_g approach is a pipeline that uses a variety of computer and human modules

or tasks to supplement the output of a well-performing natural language processing software and prioritize expert curation. Having identified, named scientific named entity recognition as a major challenge and prerequisite for relations extraction, polyNER, the third approach uses minimal, focused expert knowledge to generate annotated entity-rich corpora data and bootstrap scientific named entities classifiers. This work shows that systems combining existing software and minimal human input can achieve performance comparable to that of a state-of-the-art domain-specific Natural Language Processing software and demonstrates the potential of hybrid human-computer partnership alternatives to sometimes impractical state-of-the-art approaches.

CHAPTER 1

INTRODUCTION

The amount of scientific literature published every year is growing at an alarming rate. Some studies place the number of scientific journals at more than 33 100 and the number of articles published each year at 3 million [66]. As a direct result, the amount of information (e.g., results of experiments) embedded within published literature is overwhelming. In addition to the sheer volume of articles, important findings are locked in tables, figures and text of various formats. Reading and extracting pertinent information from full-text articles has become an unmanageable task. This problem hinders the advancement of science, making it hard to build on existing results buried in the literature. It also makes it difficult to translate results into applications and thus to produce valuable products. For example, in materials science and chemistry access to large scientific databases enables high-throughput computational searches and discoveries [100]. Our goal is to transform an avalanche of publications into a machine-accessible and human-consumable source of knowledge. Ideally, since machines are capable of processing volumes of text faster than their human counterparts, a fitting solution would involve computers "reading" thousands of papers and outputting structured content for human consumption.

While computer-based solutions have improved significantly over the past few decades, extraction of structured data from unstructured documents remains a challenging task and still requires human supervision. There are several resources to guide automated text extraction for non-scientific entities (e.g., structured or semi-structured data Wikipedia [134], DBpedia [8] or database equivalent of Wikipedia, the CoNLL [117] dataset for Natural Language Processing (NLP) tasks. Many rule-based, machine learning (ML), and hybrid named entity recognition (NER) approaches have been developed for particular entity types (e.g., people and places) [95, 88]. Such resources are not often available for scientific information extraction, except in a subset of scientific fields notably bioinformatics. For instance, the National Center for Biotechnology Information (NCBI) provides access to dozens of databases

for proteins, genes, medical abstracts, etc. (See `https://www.ncbi.nlm.nih.gov/`.) As a result, there is also considerable prior work in biomedical facts extraction. State-of-the-art methods often use hybrid rule-based, machine learning, and statistical techniques to extract entity names and relations from the literature [78, 140]. These methods generally require large amounts of *quality* training data, which is not readily available in many domains. Instead, attempts to extract a new type of entities and entity relations rely on large, carefully annotated training data tailored for new target scientific entities, often requiring some amount of in-depth domain knowledge. Hence, even state-of-the-art machine learned extraction systems do not typically perform well when applied to different domains [74]. For example, considerable effort is involved in selecting and (often manually) generating quality data for trainable statistical named entity recognition systems [75].

Our work is mainly motivated by the pressing need for extraction of materials and their properties from the scientific literature. Materials informatics [99, 56, 35], often referred to as the fourth paradigm of materials discovery [131, 6], 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 databases is a monumental and costly task requiring years of expert labor, as the data that populate these databases must often be extracted manually from free-text publications. While, machine learning efforts have begun in materials science [54, 113, 79, 124], the lack of annotated text hinders attempts to leverage approaches developed for biomedicine for example. We initially target the field of polymer science, one of several sub-fields of materials science with emerging interest in information extraction and lacking expert-annotated training corpora.

1.1 Thesis statement

This dissertation addresses the need for techniques that effectively combine automated and human methods for extracting scientific facts from the literature. Specifically, this work present hybrid computer-human approaches and argues that we can effectively leverage the complementary strengths of computers and humans for the extraction of previously unexplored scientific named entities and relations. We describe three approaches with varying levels of automation and human involvement. In order to explore the need for human input and occasional requirement for pointed domain knowledge, we involve humans with various levels of expertise in these approaches.

1.2 Challenges

Information extraction (IE), which is the automated retrieval of specific information from unstructured to semi-structured text, is a vast and well-established research field. Oftentimes, IE employs natural language processing (NLP), which is the ability of computers to process human (natural) languages. The history of natural language learning goes back several decades and the Special Interest Group on Natural Language Learning (SIGNNL) has been organizing workshops and conferences for at least three decades. The Conference on Computational Natural Language Learning (CoNNL) provides a manually curated dataset for different natural language processing tasks each year. For the most part, such carefully prepared resources are not available for machine-learned scientific entities and relations extraction. The careful and exhaustive manual curation of the corpus is however more expensive as it occasionally requires specialized expertise and cannot easily be crowdsourced by using platforms like Amazon Mechanical Turk, for example [22]. The lack of annotated training data, the need for expertise along with additional challenges specific to scientific IE presented in this work render generalization of annotations and extraction systems particularly difficult.

1.3 Contributions

The primary contributions of this thesis are the designs and implementations of hybrid human-computer approaches that enable, facilitate and improve the extraction of scientific facts (entities & relations) using crowds of varying levels of expertise. Our goal is to address the gap between state-of-the-art NLP solutions and current extraction needs of scientific domains such as materials science, which have not previously been the subject of intense IE research. This thesis offers key insights into ways to combine human knowledge and automated extraction techniques for the purpose of populating a database of scientific facts. In particular, we describe a crowdsourcing platform for the extraction of complex scientific properties using trained crowds supervised by experts. We provide a model for designing human and computer curation modules to capitalize on existing NLP software. Finally, we describe a method for exploring machine learning solutions in the absence of available training data for scientific named entities extraction. In more details, we implemented the following three distinct, yet complimentary human-machine scientific IE approaches:

1.3.1 χDB

The first approach, χDB, tackles the extraction of a challenging polymer property and relatively relies more heavily on human curators. χDB is a system consisting of an automated Web information extraction phase followed by a crowdsourced curation phase. The output is a high-quality human- and machine-accessible *digital handbook* of polymer properties. We show that we are able, using only a small group of students supervised by two experts, to create a quality database of properties with more polymer-polymer Flory-Huggins (or χ) parameters (263) than in other notable handbooks (traditional textbooks for materials data) [130]. The χ parameter, which characterizes the miscibility of polymer blends, is a particularly challenging property to extract, due to the fact that it is published in heterogeneous data formats (e.g., text, figures, tables) and is represented in several different temperature-

dependent expressions, hence the crowdsourced approach. We describe how our approach is likely also to work well for other complex properties and in other scientific domains.

1.3.2 T_g IE Pipeline

The second system is a hybrid Information Extraction (IE) pipeline that combines automation and crowdsourcing to extract the *glass transition temperature* (T_g) of polymers. This pipeline uses comparable numbers of machine and human tasks to customize available domain-specific NLP software for a new polymer—property extraction. T_g is an important property in the design of new polymeric materials that quantifies the temperature at which polymers transition from a glassy state into a rubbery state. The goal is to maximize throughput and accuracy while minimizing the burden on human curators. We extend our previous work, increasing the automation to develop an integrated IE pipeline that combines a general-purpose NLP toolkit, ChemDataExtractor (CDE) [124] to parse text and perform preliminary recognition with subsequent specialized human and computer curation *modules* such as a ranking system to prioritize crowdsourced tasks. 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 [127]. In comparison, the recent edition of the expert-curated *Physical Properties of Polymer Handbook* [41], last published in 2007, contains only $\sim$600 T_g values.

1.3.3 PolyNER

Having identified scientific named entity recognition as a major challenge and a prerequisite for entity relations extraction, and moving towards solutions with minimal human input, the third approach, polyNER is a hybrid computer-human system for semi-automatically identifying scientific entity referents (terms used to refer to an entity) in text. PolyNER operates in three phases, first applying a fully automated analysis to produce an entity-rich set of candidates for labeling; then engaging experts to approve or reject a modest number of

proposed candidates; and finally using the resulting labeled candidates to train a classifier. In both the first and third phases, it uses word embedding models to capture shared contexts in which referents occur. PolyNER thus seeks to substitute the labor-intensive processes of either assembling a large manually labeled corpus (collection of written texts on a particular subject) or defining complex domain-specific rules with a mix of sophisticated automated analysis and focused expert input. We further use active learning with maximum entropy uncertainty sampling to address the need to generate carefully selected training examples for ML models. Using these labels, we train word vector classifiers and achieve NER performance comparable to that of the IE T_g pipeline which relies on domain-specific NLP software [127]. Our system however, took less than five hours of expert time to achieve this result.

The rest of this thesis is organized as follows. Chapter 3 discusses various topics related to scientific information extraction including crowdsourcing, domain-specific natural language processing and semi-supervised machine learning. Chapter 4 describes an approach for crowdsourcing scientific information extraction. Chapter 5 describes an approach for supplementing NLP software. Chapter 6 describes an approach for generalizable scientific information extraction in the absence of training data. While describing our approaches, which differ in levels of automation and human involvement, we discuss time and cost reductions compared to existing alternative approaches. We discuss the generalizability of our hybrid approaches in Chapter 7 and future work in Chapter 8. In particular, we discuss time, accuracy and cost tradeoffs between these systems. We conclude in Section 9.

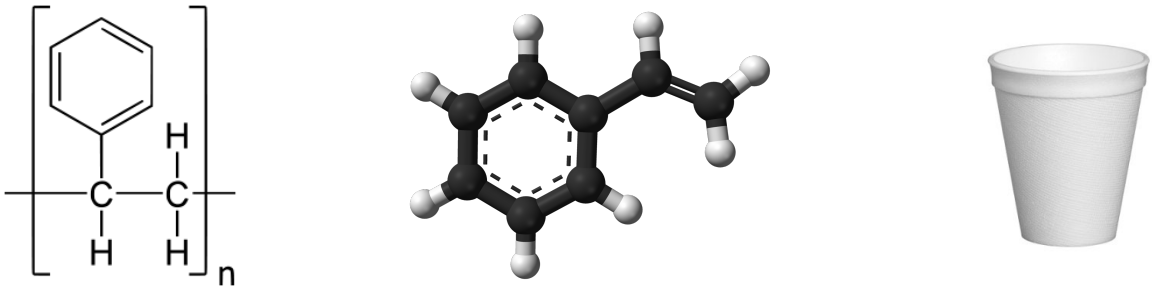
CHAPTER 2

APPLICATION BACKGROUND AND SCIENTIFIC INFORMATION EXTRACTION CHALLENGES

In material science, examples of "scientific facts" can be names and properties of particular materials, such as the melting and glass transition temperatures of polymers. These properties are particularly useful in the design of new products [100]. Polymers are large molecules (macromolecules) composed of many repeating units. 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. Figure 2.1 shows polystyrene (PS: $[C_8H_8]_n$), a common and familiar polymer.

Historically, materials properties have been collected in human-curated review articles and handbooks (e.g., the *Physical Properties of Polymers Handbook* [41], the *Polymer Handbook* [132]). However, this approach is laborious and expensive, and thus such collections are published infrequently. One excellent example of a current digital database is PolyInfo [102], which contains the records for over 200 000 properties of polymers extracted from more than 12 000 articles—another process that required years of manual expert curation effort. We contend that a better approach is to leverage information extraction techniques to process thousands of papers and output structured content for human and machine consumption. As previously mentioned, recent works have studied the automated extraction of chemical compounds [64, 113, 79, 124]. However, these approaches are not often adaptable to new domains and rely on large amounts of annotated training data. Next, we discuss the reasons why scientific entities and relations extraction remain an important research challenge,

especially in cases similar to polymer science, which have not previously been the target of intense IE research. We discuss the difficulties faced when extracting polymers names and properties and relate them to general scientific IE challenges. We also specifically discuss the two properties explored via hybrid human-machine methods in Chapters 4 and 5.



(a): Polystyrene formula. (b): Styrene ball-and-stick model. (c): Polystyrene cup.

Figure 2.1: Example of polymers: polystyrene.

2.1 Polymer Names

The complexity of scientific NER is primarily due to the fact that the same entities, for example biological [72] and chemical [75], can be described in different ways, with vocabularies often specialized to small communities. Such issues are especially evident in polymer science applications. In principle, International Union of Pure and Applied Chemistry (IUPAC) guidelines define polymer naming conventions [58]. However, such guidelines are not always followed in practice [126]. Polymer names may be reported as source-based names (based on the monomer name), structure-based names (based on the repeat unit), common names (requiring domain-specific knowledge), trade names (based on the manufacturer), and names based on chemical groups within the polymer (requiring context to fully specify the chemistry). Often, polymers are encoded using acronyms.

These different naming conventions arise in part because a desire for clarity in communications is at odds with the often complicated monomeric structures found in many polymers [7]. For example, sequence-defined polymers, where multiple monomers are chemi-

cally bound in a well-defined sequence as in proteins, often defy normal naming practices, as it is not possible to list concisely every monomer and their respective positions [85]. Another class of polymers that often suffer from complicated names are conjugated polymers, which exhibit useful optical and electrical properties. Conjugated polymers are complex due to the co-polymerization of multiple monomers (donor/acceptor units), the type and position of side chains along the polymer backbone, and the coupling between monomer units to control regioregularity [57].

Other challenges arise from the use of labels, structure referents (e.g., "micelles," "nanostructures"), and unusual author-coined acronyms. For example, one author defined the acronym DBGA for N,N-dibenzylglycidylamine and then used the string poly(DBGA) to represent poly(N,N-dibenzylglycidylamine). More naming variations result from typographical variants (e.g., alternative uses of hyphens, brackets, spacing) and alternative component orders in copolymers.

The issues just listed make identifying polymeric names a non-trivial exercise not only for computers but also for experts. As previously mentioned, we believe similar issues arise in many fields with specialized vocabularies as evidenced by NLP tools that rely at least partially on domain-specific grammar and ontologies [50, 84]. The challenges of entities being described by multiple referents (synonymy) and conversely, the same word referring to different concepts depending on context (polysemy) also exist in the biomedical field [4]. Scientific NER is also challenging due to the scarcity of entities in scientific articles. For example, it is not uncommon for scientist to write an article about one newly discovered drug or newly synthetized material; in such article there is a large imbalance between the target entity and other words or name entities in the text.

Our long-term goal is to build a hybrid human-computer system in which we leverage both human and machine capabilities for the efficient extraction from text of properties associated with specialized vocabularies.

2.2 Examples of Polymer Properties

Relations between polymers and their properties constitute an important fact that one may want to extract from unstructured text to build an easily searchable knowledge base for example. For example, we may be interested in all the melting point of a particular polymer. Beyond the previously discussed challenge of identifying the polymer name, extracting the relation adds a number of challenges. A naive approach may look for all sentences of the form *Entity X has a melting point of Y* and would yield some results. However, human language is inherently ambiguous and one cannot possibly come up with all phrases that would express this relationship. While this is true for standard relations extraction, it is especially challenging in science, considering for instance that melting point is sometimes abbreviated as *mp*, or T_m. A natural potential alternative to using string matching and/or rules would be to use machine learning models to extract properties as some previous work have done for relations between standard named entities. However, significant challenges remain: How do we obtain training data for these models? How do we deal with uncertain data due to nuances in phrasing, polysemy and synonymy? Moreover, sentences in esoteric scientific articles can be long and contain several property values comparing one to the others as in the following example about glass transition temperature: *The* ***glass transition temperature*** *(T_g) of P3PT was determined from second DSC scan to be* $37°$***C****, which is to be compared to a higher T_g of* $66.9°$***C*** *reported for* ***P3BT(11a)*** *and a lower T_g of* $12.1°$***C*** *reported for* ***P3HT.(20)****.* On the other hand, the link between two entities may sometimes spread across multiple sentences. Consider the following examples: *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 of (*$\sim 100°$***C)****.* Additionally, because these entities can be measured or determined through an important scientific process, researchers and engineers searching for this data are often interested in more metadata about the relation:

method of measuring, quantities and concentrations involved and other experimental details.

We introduce here two polymer properties that we have studied in our work, and opted to extract via hybrid human-machine approaches due to the challenges listed above: χ and T_g. These two properties pose rather different information extraction challenges: the first requires the extraction of a minimum of six metadata fields, including measurement method, while the second requires mainly a number and a unit. Thus, as we will see below, their effective extraction required different levels of human involvement.

2.2.1 Flory-Huggins Interaction Parameter or χ

One of the most important properties in polymer science is the Flory-Huggins (χ) parameter [46], which characterizes the miscibility of polymer blends and polymer solutions. Since polymeric materials are both ubiquitous and typically consist of several components, the χ parameter represents a key property in the design of next-generation materials. For example, directed self-assembly (DSA) is arguably the most promising strategy for high-volume cost-effective manufacturing at the nanoscale. The key concept of DSA is to take advantage of the self-assembling properties of materials such as χ to reach nanoscale dimensions and, at the same time, meet the constraints of manufacturing, without prohibitive high capital equipment costs of making nanoscale semiconductors[1]. The χ parameter, which depends on the temperature and the types of polymer(s) or solvent(s) involved, is universally adopted to characterize the phase diagram[2] of polymer blends. Consequently, many experimental methods have been developed to quantify the temperature dependence of χ, and tabulated values are commonly found in standard textbooks and polymer data handbooks [41]. However, many of these values have not been updated to include recent findings. Moreover, the list of polymer blends found in textbooks is not exhaustive; for example, the previously mentioned handbook contains χ values for only 41 polymer-polymer blends.

1. https://pme.uchicago.edu/features/the_promise_of_dsa_technology_for_nanoscale_manufacturing/

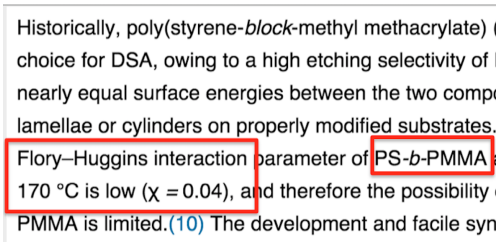
2. A phase diagram is a type of chart used to show conditions (pressure, temperature, volume, etc.) at which thermodynamically distinct phases occur and coexist at equilibrium.

The information extraction problem is complicated by the fact that while thousands of experimentally determined values have been published for hundreds of χ parameters, there is little consensus regarding the "best" measurement method. Different measurement methods yield different values, and different groups have at times reported different values for the same polymers. Part of this variability is due to inherent deficiencies within Flory-Huggins theory, which states that χ is inversely proportional to temperature. However, experimental evidence confirms a more complicated dependence on temperature and blend composition (ϕ) such that published χ values are often labeled as "effective" values in order to acknowledge these deficiencies.

A natural source of information for building such a database is the body of relevant information published in research articles. However, mining the literature for loosely structured scientific entities such as χ values, which are inevitably encoded in different forms (text, tables figures and equations) in manuscripts (see Figure 2.2), is a challenging task [54]. A parameter such as χ is not typically captured as a common metadata element, as are, for example, title, authors, and publication date. Therefore, mining publications for χ requires extracting values from non-standardized text, tables, equations and figures—a challenging task that requires encoding, formatting, and other processing activities [69]. Indeed, identifying and storing the χ parameter only makes sense if the corresponding polymers, solvents, molecular masses, temperatures, methods, and errors are also captured. These reasons motivated the thesis that the population of such a database currently requires hybrid human-computer methods.

2.2.2 Glass Transition Temperature or T_g

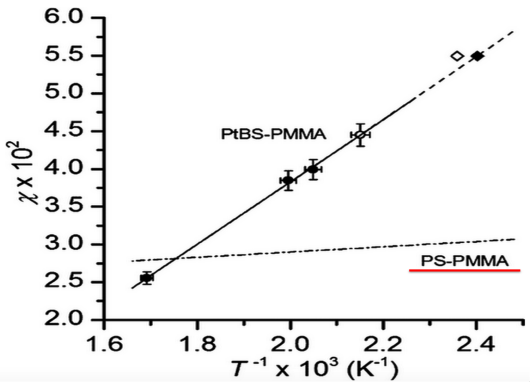
In the design of new polymeric materials, the temperature relative to the T_g can have a profound effect on the properties of the polymeric material. T_g is defined as the temperature at which a polymer transitions from a solid, amorphous, glassy state to a rubbery state as the temperature is increased. Physically, when polymers are in the glassy state, the molecules



(a): χ as found in text from [83].

$$\chi(T) = \frac{3.9 \pm 0.6}{T} + 0.028 \pm 0.002$$

(b): χ as found in an equation from [70].



(c): *Relevant* figure from [70].

material	M_n (kg/mol)	M_w/M_n	f_{PS}[a]	χN_{core}[b]	
PS	61	<1.1			
HDPE (Dow 4452N)	18	5			
6K PS-*b*-PE	3–3	<1.1	0.45	4[c]	
20–20K PS-*b*-PE	20–20	<1.1	0.45	25[c]	
28–10K PS-*b*-PE	28.5–10.5	<1.1	0.69	13[c]	
33–5K PS-*b*-PE	33–5	<1.1	0.84	6[c]	
100K PS-*b*-PE	50–50	<1.1	0.45	65[c]	
200K PS-*b*-PE	100–100	<1.1	0.45	130[c]	
PS (Dow 685D)	150	1.8			
PMMA (Arkema V825N)	52	1.9			
FLPS	72	1.7			
SAN	71	1.6			
42K PS-*b*-PMMA	21–21	1.1	0.54	7[d]	0.9[e]
74K PS-*b*-PMMA	37–37	1.1	0.54		1.6[e]
100K PS-*b*-PMMA	50.6–47.6	1.1	0.55	17[d]	2.1[e]
160K PS-*b*-PMMA	80–80	1.1	0.54	27[d]	
260K PS-*b*-PMMA	130–133	1.1	0.53	44[d]	5.4[e]
900K PS-*b*-PMMA	450–450	1.1	0.54		18.8[e]

(d): χ as found in table from [12].

f

Figure 2.2: Four examples of how the Flory-Huggins (χ) parameter for the same pair of polystyrene (PS) and poly(methyl methacrylate) PMMA may be found in the literature in various forms.

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 [19]. 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.

From this perspective, T_g is a better-suited candidate than χ for automated extraction,

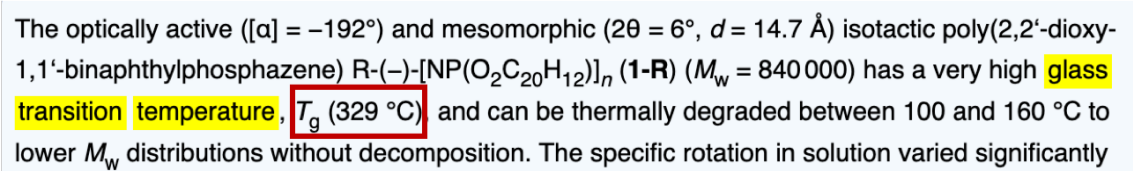


Figure 2.3: Example of T_g in the literature from [25].

as we seek to extract a single polymer (vs. two for χ) and a single matching number (vs. a minimum of 6 metadata fields). 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 well-defined problems (e.g., extracting spouses by identifying the subject and object in sentences containing the word *married*) in standard NER tasks. In our case, identifying the polymer name preceding and the number following such word combinations as "*glass transition temperature*" or "textitglass transition temp." or "T_g". However, designing and maintaining rules 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. Researchers often combine the two methods to increase the completeness and accuracy of extracted information [28]. 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 [142] 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. For these reasons, we contend that a hybrid human-computer approach is still desirable in order to leverage and adapt existing state-of-the-art, off-the-shelf NLP software. While the extraction of T_g described in this work relies on rules

and subsequent curation human and computer modules or tasks, we discuss a generalizable, machine learning approach for polymer name recognition in Chapter 6, which can then be used for general machine learned extraction of polymer properties.

CHAPTER 3

RELATED WORK

We review here current practice for building collections of scientific facts and populating scientific databases including automated information extraction methods, crowdsourcing, domain-specific toolkits and weakly supervised methods.

3.1 Scientific Databases

Major scientific databases have emerged in various fields where data is growing at exponential rates and the need for data sharing is recognized by the community, notably in biotechnology [13, 87]. In materials science, the Materials Project [63] provides access to large numbers of computed values. For polymers, the expert-curated *Physical Properties of Polymers Handbook* [41], last published in 2007, is a valuable source of data. However, while a valuable resource, it lacks recent results from the literature and does not contain an exhaustive list of polymers. Moreover, generating such resources often requires years of human effort in extracting, curating, annotating and validating data. As previously mentioned in Section 2, PolyInfo [102] is an example of large digital database of polymers. However, it is manually curated and also requires years of tedious and costly expert curation. Our three IE approaches attempt to quickly populate a database of scientific facts by combining manual and automated techniques, thereby accelerating scientific discovery, lowering the burden on curator and saving annotation costs.

3.2 Information extraction (IE)

Information extraction (IE) from text has been extensively studied [33]. IE aims to extract structured information from unstructured and semi-structured documents. It often focuses primarily on extracting information from written language via natural language processing [23]. Sub-disciplines include Web IE [10] and IE from PDF documents and images.

Web IE leverages the inherent structure in HTML rather than grammatical rules to extract semantically meaningful information. Web IE approaches work well when extracting information from many pages with the same structure (e.g., real estate listings), but do not work well for heterogeneous web pages or when page structure changes [94]. Extracting information from other data types, such as images and PDFs, is particularly difficult. In the case of images, variations in texture, contrast, font size, style and color, orientation, alignment, etc., all impact the extraction process. Similarly, PDF files, while easy to understand for humans, are not designed for machine accessibility. Thus, it is challenging to extract information from embedded items—such as tables and equations—due to the lack of structure in the document. For example, extraction of tables from PDF documents typically relies on identifying cell borders and attempting to map text locations relative to these borders. As tables differ significantly between documents, a considerable amount of human assistance is needed to achieve good results.

3.3 Crowdsourcing

Crowdsourcing, generally speaking, involves humans working together to solve a problem. Its basic motivation stems from the fact that humans perform certain tasks better than computers. Crowdsourcing is also useful when a task requires multiple opinions or perspectives to reach a consensus. One early application for crowdsourcing was image labeling and recognition. Nowadays many non-trivial problems can potentially benefit from crowdsourcing using platforms like Amazon Mechanical Turk [22]. While the exact definition of crowdsourcing and nuances in its practices are still being studied [43, 39], over the past decades numerous crowdsourcing systems have emerged. Well-known examples include Wikipedia, where many users contribute to create and curate the online encyclopedia, and citizen science projects, where members of the public assist scientists in conducting research.

An instance of the latter class is Galaxy Zoo, a crowdsourced astronomy project that asks participants to classify galaxies appearing in images taken by professional astronomical

facilities via a web portal [82]. To date, more than 175,000 people have provided shape analyses of more than 1 million galaxy images sourced from the Sloan Digital Sky Survey [138], generating around 60 million classifications that lead to more than 50 publications [123]. This success spurred the development of Zooniverse [123], a citizen science web portal that now hosts more than 25 similar projects. The project has also led to studies of crowd motivation, "*gamised*" behavior—where users generate play in a platform—and the "*gamification*" of science projects—where designers embed games into the platform [107, 53, 135].

Our first IE approach is inspired by previous work that suggest that interactive, intelligent approaches that combine the strengths of human interaction with the algorithmic power of AI can help solve problems that could not be solved by either computers or humans alone [59]. In this kind of system, focused on improving graphical user interfaces, human work is both used directly to complete automated services and as feedback for machine learning classifiers training to further facilitate that work in the future. We tailor this concept to the problem of scientific relations extraction, which requires higher levels of expertise and human involvement than in citizen science projects. Indeed, 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 [125] and also to ensure quality control (i.e., expert curation) [120] while extracting empirical observations from literature. Similarly, in the GeneWays system, experts remove controversial collected data during the automatic literature extraction [115]. CrowdDB [47] 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. [136] also pursue this goal, using a hybrid machine learning and crowdsourcing approach to identify published randomized controlled trials (RCTs) [136]. They use machine learning classifiers to recognize citations that are deemed highly unlikely to describe RCTs, deferring to crowdsourcing otherwise.

3.4 Domain-Specific IE

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) [48, 49], cTAKES [118], and medKAT [2] 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 [50] 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 [64, 54, 124].

3.4.1 CDE and CDE+

We specially introduce the ChemDataExtractor (CDE) [124], a state-of-the-art chemical NLP tool that combines a dictionary, expert-created rules, and machine learning algorithms to recognize chemical compounds and their properties. CDE was trained on the CHEMD-NER corpus: a collection of 10 000 PubMed abstracts with 84 355 chemical entity mentions

labeled manually by expert chemistry literature curators, following annotation guidelines specifically defined for this task [75]. CDE primarily uses conditional random field (CRF) based recognizer for chemical names, in combination with a dictionary-based recognizer that provides improved performance for trivial and trade names, and a regular expression-based recognizer that excels for database identifiers and chemical formulas [124]. Whereas a discrete classifier predicts a label for a single sample without considering "neighboring" samples, a CRF is a sequence-level classifier which is better at capturing strong inter-dependencies of the output labels [61]. A CRF layer is often used as the output later of long short-term memory (LSTM) neural networks—another standard named entity recognition approach—to prevent "illegal" label combinations. For example, using the IOB (short for inside, outside, beginning of a named entity) tagging scheme, [beginning,outside] constitutes an "illegal" label combinations. CDE's dictionary-based recognizer uses a word list compiled from the Jochem chemical dictionary [55], with an automatic domain-specific filtering process that excludes entries that lead to false positives. They eliminate redundancy between similar names CDE stores the dictionary as a directed acyclic word graph (DAWG). CDE defines grammar rules which carefully combine part-of-speech tagging, dictionary entries and regular expressions for named entity recognition, and uses a more fine-grained tokenizer and a series of rule-based parsing grammars, each tailored specifically for extracting a certain property type.

We modified CDE with manually defined polymer identification rules [127], creating what we term here CDE+. We compare our methods against both CDE and CDE+ in Section 6.3.3.

3.5 Statistical and Deep Learning Approaches

With the recent advances in machine learning and statistical inference approaches, scientific applications are turning their attention to deep learning tools such as DeepDive [36]. PaleoDeepDive [106], built upon DeepDive, automatically extracts paleontological data from

text, tables, and figures in scientific publications. GeoDeepDive [141] performs similar tasks in the geosciences. For good performance in such applications, IE software often relies on and extends large amounts of training data, which is often expensive to obtain. To circumvent the need for training data, DeepDive uses automatically labels data based on entities stored in large databases. PaleoDeepDive builds on PaleoDB [3] and GeoDeepDive builds on Macrostrat [1]. DeepDive labels any entity pair that appears in the database as *True*. The user defines features (e.g., if a specific keyword appears between two entities, that pair a certain attribute is labeled *True*, but if the entity pairs are too far apart, another attribute is marked *False*), the system then uses statistical inference to determine the probability that each newly discovered pair of interest is *True*. However, many fields, including materials science, do not yet have access to large and structured sets of annotated texts that deep learning systems can use to learn scientific facts and relationships. Our second IE approach is an intermediary, but essential, step towards accumulating such structured data.

3.6 Weakly Supervised Learning

Weakly supervised learning methods work with much less training data. They generally fall under three categories: bootstrapping, semi-supervised learning and active learning. In entity relations extraction, bootstrapping starts from a small set of seed relation instances and iteratively learns more relation instances and extraction patterns. The key difference between the semi-supervised learning and active learning is that the former relies on approximately labeled data (as opposed to correctly labeled data for supervised learning) and the latter starts off with unlabeled data. Semi-supervised learning attempts to label data automatically by using prior knowledge and a set of labeled data. For example, it assumes that if x and y are similar, they probably have the same label (first cluster the whole dataset, then label each cluster with labeled data [145]). Active learning assumes there is a source of knowledge, such as a human expert, that can be queried to label a selected batch of unlabeled data.

3.6.1 Bootstrapping

A representative work on bootstrapping for relations extractions is Snowball. Snowball [5] which improved the DIPRE system [20], used an intuitive idea to collect new entity relations using a set of seed entity pairs. In the DIPRE system, the intuitive assumption is that, given a few seed entity relations, the text between two known target entities in close proximity of each other describes and constitutes a *pattern* of the relation between the two. Since that is not the case in practice, the system uses a limited set of regular expressions to limit useful patterns, hence decreasing the number of false positives. A key improvement of Snowball is that its patterns include named-entity tags (PERSON, LOCATION, ORGANIZATION, etc.). Given a handful of seed tuples of ORGANIZATION and LOCATION, Snowball attempts to learn the relation *HeadquarteredIn* by assuming that each time the tuples appear in close proximity to each other, the text in between illustrates the desired relation. This text can then be used to discover new tuples, which can in turn be used as seeds for the next discovery round. Of course, organizations may be located but not headquartered in multiple cities; hence it is important to inspect the quality of extraction patterns to reduce noise in the generated output.

3.6.2 Semi-Supervised Learning

Distant supervision illustrates the concept of semi-supervised learning by mapping known entities and relations from a structured knowledge base onto unstructured text [106, 36]. With freely available structured knowledge base such as DBPedia [8] and Freebase [18], it is possible to leverage a large set of known entity pairs to generate training data. *Data programming*, as used in the Snorkel system, has users define *labeling functions* to provide labels for data subsets [110]. Errors due to differences in accuracy and conflicts between labeling functions are addressed by learning and modeling the accuracies of the labeling functions. Under certain conditions, data programming achieves results on par with those

of supervised learning methods. While writing concise scripts to define rules may seem to be a more reasonable task for annotators than exhaustively annotating text, it still requires expert guidance. *Snuba* [133] addresses this constraint of data programming by automatically generating heuristics using the labeled and unlabeled data it has access to. Both Snorkel and Snuba use the generative model to aggregate heuristic labels, and learn their accuracies in order to make final predictions. While Snorkel's generative models are designed to model the noise in user-defined heuristics—which are much more accurate than automatically generated heuristics—Snuba introduces a statistical measure to automatically recognize when such generative model is not learning heuristic accuracies successfully and therefore abstain from labeling when if the heuristic has low confidence.

In data programming as in boostrapping and distant supervision it is important to evaluate the quality of functions and extraction patterns to decrease noisy patterns. While Snorkel and Snuba demonstrate good results with entity relations and text classification, the systems assumes a set of heuristics (user programmed, or greedily mimicking the user manual process) to learn from. Such assumption is quite sensible for examples such as topic modeling (operations over bag of words) or entity relations (regular expression heuristics); however, in the case of scientific NER it remains more challenging. For example, in the case of polymer name recognition, besides pattern matching the string "poly" (which will not work for acronyms and non-standard names), there are no other intuitive user-based rules description of a polymer to leverage. In other words, we cannot easily replace human and expert labeling of our corpus.

3.6.3 Active Learning

Active learning [144] assumes that gold standard labels for unlabeled instances can be obtained by querying an oracle (domain expert or source of knowledge). The goal of active learning is to decrease labeling costs by requesting a limited number of labels from the oracle, that have been deemed most valuable by the learner. Uncertainty sampling approaches

define "valuable" data by measuring uncertainty in the predictions. For example, in the case of a single learner, querying predictions with maximum entropy in which the learner assigns all classes with equal probability [80] or predictions closest to the decision boundary in the case of support vector machine classifiers [24]. In the case of multiple learners, query-by-committee requests labels for unlabeled instances on which the learners disagree the most [122]. Uncertainty sampling and query-by-committee are representative approaches based on informativeness, where informativeness measure show well an unlabeled instance helps reduce the uncertainty. Another selection criterion addresses representativeness, which measures how well an instance helps represent the structure of input patterns; in this case selection is made by querying data from unlabeled clusters of data [98, 34].

Our third IE approach, focused on NER, combines semi-supervised and active learning [98, 11]. We use active learning to efficiently use expert time and obtain quality training data. We build a scientific entity classifier that can complements other approaches. For example, it could be used as a scientific entity tagger (i.e., recognizer) to be used with data programming to extract relations.

CHAPTER 4

CROWDSOURCING SCIENTIFIC INFORMATION EXTRACTION

4.1 Introduction

Historically, for the reasons mentioned in Chapter 2, materials properties have been collected in human-curated review articles and handbooks. Therefore, we first explore crowdsourcing-based methods to extract a particularly challenging property—the Flory-Huggins or χ parameter. This parameter is represented in several different temperature-dependent forms and encoded in text, figures, equations, and tables. Further, the χ value alone provides little value without at least six metadata entries: a pair of two polymers or a polymer/solvent pair, the χ value, which can be 1 term or 2-3 equation terms along with a unit, and the measurement method used. The measurement method is particularly important as there is little consensus on the best method for measuring the χ value. Ideally, other information is also needed such as concentrations of compounds and error terms, increasing the number of input fields to be extracted to up to 30. Due to such complexity, our first approach—the semi-automated χDB system—relies on a relatively high level of human input from trained crowds to extract polymers and their properties. The χDB system consists of an automated Web information extraction phase followed by a crowdsourced curation phase. The output of this workflow is a high-quality human- and machine-accessible *digital handbook* of polymer properties.

The rest of this chapter is organized as follows. Section 4.2 describes the χDB architecture. Section 4.3.1 presents the data collected via crowdsourcing. Section 4.3.2 explores the application of machine learning algorithms to improve the automatic selection of χ-relevant publications. Finally, we conclude and discuss future work in Section 4.4.

4.2 Design and Implementation

Mining the literature for a loosely structured property such as the χ parameter requires extracting values from a variety of objects, including text, figures, tables, and equations; processing the many different forms in which the property occurs, e.g., a single number at a given temperature or a linear equation as a function of temperature; and identifying associated information such as the polymers and solvents involved, their molecular masses, the temperature(s) at which experiments were performed, the methods used, and any error estimates. Thus, the techniques used to find, extract and store χ must be flexible.

Given these multiple levels of complexity, we have developed χDB—a hybrid machine-human system that leverages both automatic extraction and expert human review via crowdsourcing. The χDB workflow shown in Figure 4.1 comprises three main phases: automatic download and first-level extraction of publications; crowdsourced extraction and review (the "review process") of χ values, and finally the exposure of a curated database of χ values (the "Digital Handbook of Properties"). In the rest of this section, we define the χDB data model and then describe the system architecture used to realize each of these workflow phases.

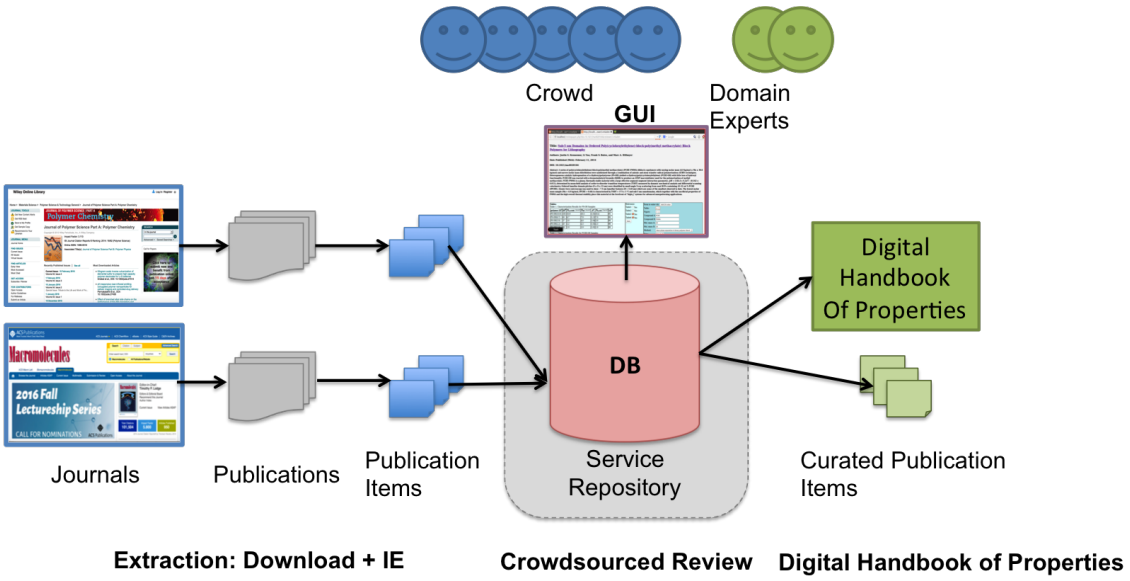


Figure 4.1: χDB architecture

4.2.1 Data Model

The χDB data model is designed to represent (1) the complex extraction and review workflow, (2) the various temperature-dependent formats in which χ occurs, and (3) the complete provenance of each extracted value. The χDB data model includes seven core tables: `papers` (extracted publications), `items` (extracted publication items), `sources` and `reviewed_sources` (reviewed information before and after consensus), `chis` and `reviewed_chis` (χ values before and after consensus), and `reviewed_papers` (classified papers). One challenge when defining the data model is the need to support different representations in which χ is specified. After reviewing the literature, we developed a data model that could include four main representations of χ:

1. a number at a specific temperature;
2. a linear equation in terms of temperature: $\chi = A + \frac{B}{T}$;
3. a quadratic equation in terms of temperature: $\chi = A + \frac{B}{T} + \frac{C}{T^2}$;
4. a number that combines χ and N, where N is proportional to the degree of polymerization or molecular weight: χN; and a final catch-all class,
5. other representations.

4.2.2 Extraction

χDB first discovers and downloads relevant publications—in this case publications that contain the keyword *Flory-Huggins*—from suitable journals. It then uses an HTML tag parser to extract structured publication metadata, including Digital Object Identifier (DOI), title, authors, and date of publication. This information is used to index the publication such that it can be linked to other stored information (e.g., referenced values in other papers). Finally, the publication is parsed into *items* (e.g., abstract, figures, tables, equations, text) that are

separately downloaded and can be reviewed individually. Links between publication items and their originating publication are maintained so that they can be displayed to reviewers in a coherent manner. The full text and the original URL are also stored such that reviewers and users can retrieve the original publication.

We implemented this phase in three components: a Python web crawler (to discover relevant publications), a downloader (to download a copy of the publication), and a WebIE extractor (to extract metadata and items from the publication). We initially focused on *Macromolecules*, a leading scientific journal on polymers. The crawler is configured to use the *Macromolecules* search capabilities to prioritize downloads. After discussion with experts, we chose the search term *Flory-Huggins* and specified a date range from January 2010. The crawler returns a ranked list of publications. The downloader uses these results to download each publication (as an HTML file) using the URL returned by the crawler. The downloader extracts relevant metadata from the structured web page (DOI, title, authors, etc.) Finally, a Python WebIE script parses the HTML to detect and extract items from the publication (e.g., abstract, images, equations, and tables). The abstract and the HTML tables are stored directly in the χDB database. Figures and equations are downloaded and referenced in the database.

4.2.3 Crowdsourced Review

To assemble a crowd for reviewing extractions we developed a materials science course that combined teaching the fundamentals of polymer chemistry and physics and reviewing the literature containing χ parameters. The reviewing component of the course tasked the students with extracting χ parameters using the χDB system. This involved reviewing the free-text publication, and entering any χ values that they identified.

We implemented this phase as a PHP-based web service and PHP/HTML website. Due to copyright restrictions, the reviewing components of χDB are accessible only within the University of Chicago network. The review interface includes two main pages: a list of all

publications with assigned reviewers and a review page for reviewing publications and items. We implemented a consensus-based review process using two reviewers per paper to reduce error. We rely on a second class of reviewers (experts) to resolve conflicting reviews.

An individual review consists of scanning extracted items for χ values. Once identified, reviewers are asked to extract χ values from all of these items, with the exception of figures as extractions from figures are likely to be inaccurate. The reviewer enters each extracted χ value in an online form. The item from which a value is extracted is marked as *relevant*. Note: items may be marked as *relevant* even if they do not contain any χ values. For example, a *relevant* figure may be a phase diagram or a micrograph of the material; a *relevant* table may contain supporting information. If a paper contains a single χ value or a single *relevant* item, it is also marked as *relevant*. Consequently, a paper that contains neither is classified as *irrelevant*. Figure 4.2 shows an example of the review form. To ensure that the resulting database is unambiguous, we define a set of minimum required information for submission of a χ value. Some χ values are embedded directly in the text (rather than in an extracted item); therefore, reviewers are able to retrieve the full text article via the link on the review page. If additional χ values are found in the full text, reviewers click the "Add Chi" button next to the abstract with the possibility to indicate in the form that the value was actually extracted from the main text. Second reviews of the same publications consist of a similar process; however second reviewers are able to view the previous reviewers' input before submitting their own, giving them the opportunity to identify errors or conflicts between reviews. In the case of errors, the interface allows submission of either review; in the case of conflicts it allows the publication to be flagged for expert review.

Students reported an average of 15 minutes to review *relevant* publications and five minutes to review *irrelevant* publications. Submissions from second reviewers are automatically stored in our Digital Handbook of χ values.

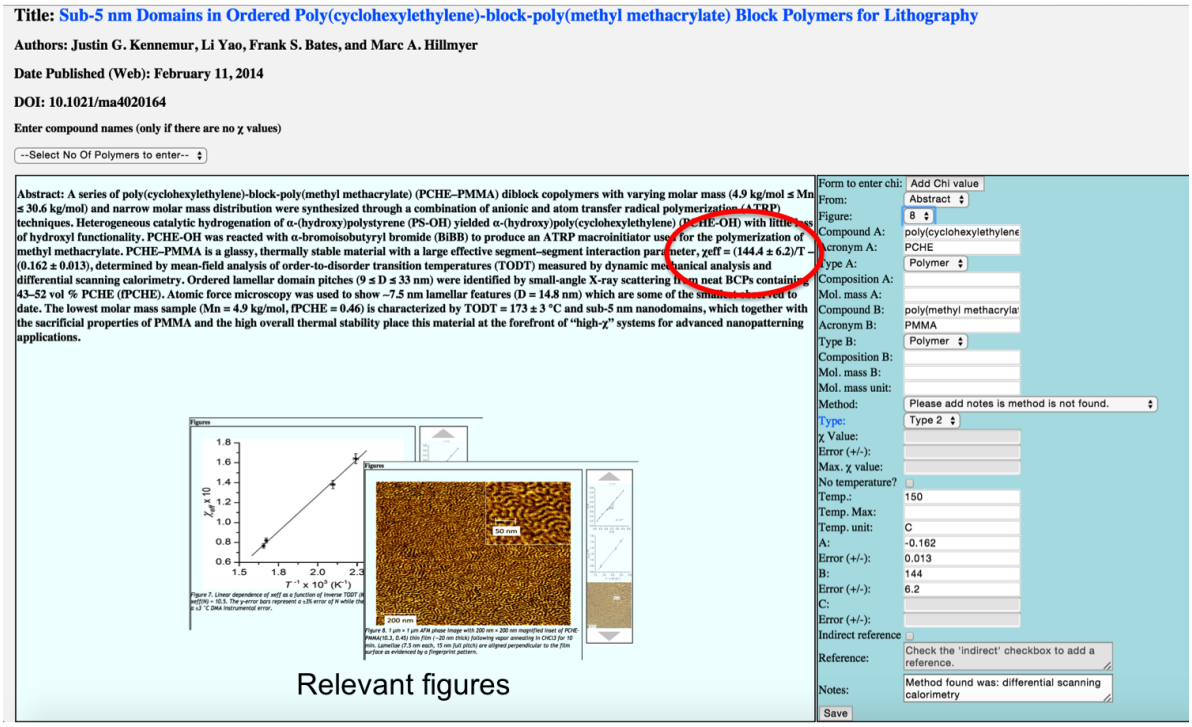


Figure 4.2: Screenshot of the χDB Graphical User Interface with the χ entry form enabled

4.2.4 Digital Handbook of χ Values

Once a χ value has passed through the review cycle, it is stored in the curated section of the database with associated provenance information that links the value back to the original publication, the item in which it was found, and the reviewers that extracted the value. To facilitate broad access to the database, χDB offers a web service API and HTML website. The website allows users to browse and search the database for specific χ values. The web service API provides programmatic access to χ values for use by custom applications, for example to retrieve χ values for a set of specific polymers that may then be used for calculations or visualizations. Both the website and web service are available at `http://pppdb.uchicago.edu`.

The website allows users to query for information related to a particular polymer. Once the user selects a particular polymer from the search interface, he or she is presented with a table of searchable χ values that relate to that polymer. Each row in the table includes the

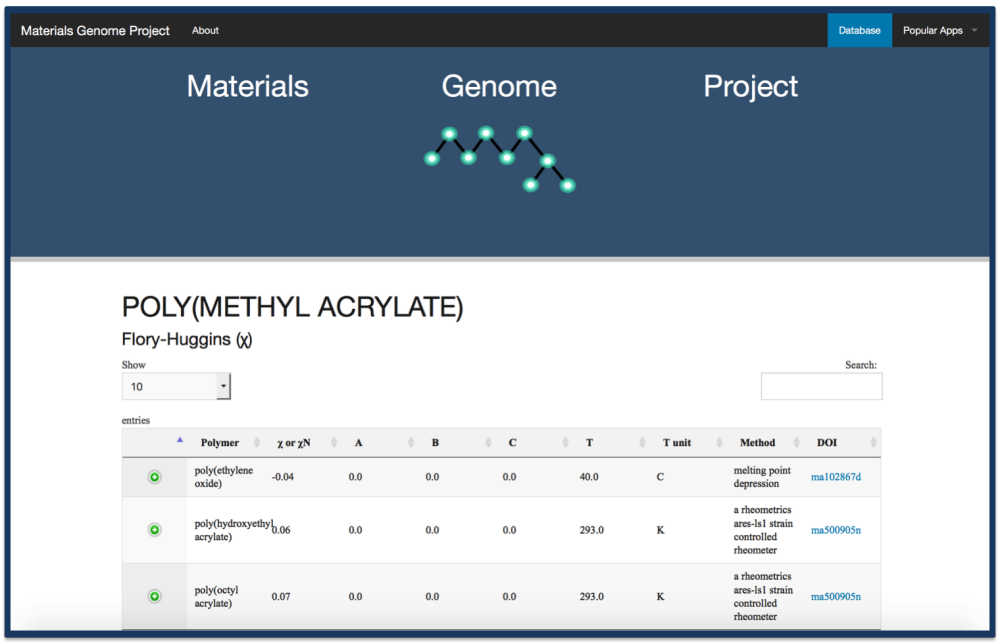


Figure 4.3: Screenshot of the χDB Digital Handbook

second compound (polymer or solvent) involved in the interaction, the measurement method used (where available), the temperature at which the parameter was measured (in various forms), and a link to the original publication. Rows can also be expanded to show additional metadata such as molecular masses and concentration. Figure 4.3 shows an example of χ values for poly(methyl acrylate) in the Digital Handbook.

The χDB REST API supports querying the Digital Handbook for χ values that relate to a specific polymer-polymer or polymer-solvent pair. The REST API has been used to create a Flory-Huggins phase diagram generator for specific polymer blends. This application determines the liquid-liquid curves for a binary blend of polymers, as well as a polymer solution.

4.3 Evaluation

Here, we first give a brief qualitative review of the data collected by our crowds before evaluating the prioritization of human paper reviews. By having an informed expectation of how the χ parameter is published in the literature and prioritizing reviews, we can then

improve future hybrid crowdsourcing and automated extraction efforts.

4.3.1 Scientific Insight from Data Collection

During the class and over a two-month period immediately thereafter, students reviewed 376 publications from the period 2010–2015 in *Macromolecules*. We briefly explore here the results of extractions, looking specifically at the characteristics of the χ values, the range of compounds for which χ values were collected, and the methods used to derive χ values.

χ Values: Of the 376 publications reviewed, students deemed 259 (69 %) of the papers *relevant*, of which 145 (38.5 %) of the papers contained one or more χ values. Our dataset includes 388 χ values, including 237 (61 %) polymer-polymer χ values. Measured χ values account for approximately half (48.5 %) of all χ values extracted, the other half (51.6 %) are cited from other publications. Of these measured values, the dataset includes 84 (21.7 %) measured polymer-polymer χ values. In the most focused case of measured polymer-polymer pairs, we found that 70.9 % of χ values were embedded directly in publication text, and 9.7 % in the abstract. Combined, these values indicate that mining text for χ values would potentially capture about 80 % of χ values. The vast majority (89.0 %) of χ values that we identified were published as type 1 or 2 i.e., a number or a linear function of temperature.

Compounds: Polystyrene (PS) is the most studied polymer by a large margin, with 140 χ values collected. The second and third most frequent, Poly(methyl methacrylate) (PMMA) and Polyisoprene (PI), have 59 and 22 χ values, respectively. The average number of χ values per polymer is 4.74. Not surprisingly, the most frequent polymer pair is PS–PMMA, with 36 χ values.

Methods: One final area of great interest to our experts was evaluating the method used to measure the χ values. Unfortunately, the method was not always present (or clear) in publications. Students were unable to identify the method for 62 (16.0 %) of the 388 χ values found and were unsure about 12 others (3.1 %), resulting in a total of 19.1 % χ values with no identified method. Originally, experts provided a list of seven methods that they

expected would be commonly used. Analysis of our dataset reveals that, for the target case of measured polymer-polymer values these methods are indeed the most commonly used, with only four of the 84 measured polymer-polymer values not using one of these seven methods.

4.3.2 Prioritizing Reviews

While our approach has established a rich database of χ values, there is potential for further improvements. For example, only 38.5 % of our selected publications contained χ values; thus, about 62 % of the papers curated by reviewers did not in fact contribute to the digital handbook. As a first step towards improving this ratio we have investigated the application of machine learning techniques to optimize the prioritization and classification of *relevant* publications.

To undertake this task, we used the Support Vector Classifier (SVC) from Scikit Learn [103], an open source machine learning Python library. SVC is an implementation of Support Vector Machines (SVMs), supervised learning models with associated learning algorithms that analyze data and recognize patterns. The models map data into a feature space to make predictions.

Three performance metrics are commonly used to evaluate the accuracy of classifiers: precision, recall, and F-measure. Precision and recall are expressed in terms of *Positive* and *Negative* predictions, i.e., in our case *Contains* χ and *Does not contain* χ; *True* and *False* predictions correspond to correct and incorrect predictions. Precision measures the percentage of predictions that were correct while recall measures the percentage of items in the test dataset that were correctly predicted. Precision and recall are defined in Equations 4.1 and 4.2, respectively.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \tag{4.1}$$

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{4.2}$$

The F_X-score is a measure of a test's accuracy. The traditional F-measure or balanced F-score (F_1 score) is the harmonic mean of precision and recall; it can be interpreted as a weighted average of the precision and recall, with a best value of 1 and worst of 0. The general formula for positive real β is defined in Equation 4.3.

$$F_\beta = (1 + \beta^2) \times \frac{precision \times recall}{\beta^2.precision + recall} \tag{4.3}$$

4.3.2.1 Test dataset

Our datasets include two sets of abstracts. The first set is composed of all abstracts of publications reviewed by the students, each of which has been classified by them as either *relevant* or *irrelevant*. These 376 publications were selected by the χDB crawler and are therefore biased by the *Flory-Huggins* keyword search. (However, as previously discussed, only 145 of these publications contained χ values.) To address this bias we downloaded an additional 135 publications from two arbitrarily chosen issues of *Macromolecules* (January 12, 2010 and January 26, 2010). Table 4.1 shows the sets of abstracts used in the classification of abstracts; we call the initial and biased set of abstracts "biased abstracts" and the larger set, which contains both the original 376 biased abstracts and the additional 135 unbiased abstracts, "All abstracts." To classify the additional set of papers we visually inspected the abstracts and full text of each publication and reviewed them for χ values.

Table 4.1: Description of abstracts used for classification of χ-relevance.

Category	**Biased abstracts**	**Unbiased abstracts**	**All abstracts**
Relevant	145	2	147
Irrelevant	231	133	364
Total	376	135	511

4.3.2.2 Results

We applied Scikit Learn's Support Vector Classifier to the set of abstracts, varying just the criteria used to identify abstracts as *relevant* or *irrelevant*. The features used by the classifier are generated using a word-weighting scheme commonly used in information retrieval [109]. The abstracts are first converted to a matrix of token counts and subsequently transformed into a normalized tf-idf (term frequency-inverse document frequency) representation. The two terms are multiplied in order to reduce the impact of terms that occur frequently in a given corpus and thus are less informative. We used three different definitions of *relevancy*: includes χ value; includes measured χ value; and includes measured polymer-polymer χ value.

Table 4.2 shows that the performance of the classifier for both sets of abstracts. Accuracy improves as *relevancy* becomes more specific. We also see a small ($\approx$3–7 %) improvement in accuracy when using all abstracts. When using all abstracts, the accuracy of classifying measured between two polymers (as opposed to between a polymer and a solvent) relevant papers is 86.9 % precision and 90.9 % recall.

There is a tradeoff between maximizing the number of *relevant* publications (and minimizing the number of *irrelevant* publications) retrieved. Deciding whether these scores are acceptable depends on the cost of errors (false negatives and false positives). Our observed precision score (of 86.9 %) means that 13.1 % *irrelevant* papers remain; a considerable improvement over the initial 61.5 % of publications that did not contain χ values. The recall score of 90.9 % means that we misclassify $\approx$9 % of *relevant* papers. As ideally we would like to capture all such publications, further work should aim at improving this score. Nevertheless, our results demonstrate the potential of capturing a significant portion of targeted publications in the literature.

We observe that the top 25 features (words) used by our classifier in the most focused case of polymer-polymer pairs include a mixture of more or less χ-related terms. For example,

Table 4.2: Classification of abstracts in χDB

Relevancy (contains)	Metric	Biased abstracts	All abstracts
χ Values	Mean F1 score	0.624	0.679
	Mean precision score	60.5 %	65.1 %
	Mean recall score	64.5 %	71.2 %
Measured χ values	Mean F1 score	0.790	0.835
	Mean precision score	75.9 %	80.9 %
	Mean recall score	82.2 %	86.4 %
Measured polymer-polymer χ values	Mean F1 score	0.852	0.890
	Mean precision score	82.7 %	86.9 %
	Mean recall score	87.8 %	90.9 %

terms like "process," "parameter," and "form" could refer to various experimental settings. On the other hand, the word "domains" (as in microphase domains) is relevant to measuring χ and is also used for a wide variety of applications in which χ is important. χ is a measure of polymer-polymer "interaction" that is present in the list of features. Microphase "morphologies" are relevant to measuring χ via phase diagrams. This combination represents a challenge in further isolating publications that are specifically related to χ and may require incorporating some domain knowledge into the χDB workflow.

4.4 Conclusion

We have developed χDB, a hybrid human computer-system that extracts the Flory-Huggins (or χ) parameter from scientific literature in order to contribute to a digital handbook of polymer properties. Our work to date has extracted 388 χ values for 120 polymers and 30 solvents. Our 237 measured χ values for blends of 63 unique polymers exceed the 134 χ values for blends of 41 unique polymers found in the *Physical Properties of Polymers Handbook* [41]. One reason for our superior performance is that we were able to collect values reported after the 2007 publication of the *Handbook* (84 of our χ values are from 2010 to 2015); another is that our more exhaustive search leads us to find earlier values not reported in the *Handbook*. Our results emphasize the potential for using our approach to create and

maintain a digital database of χ parameters that is more comprehensive and up to date than any survey publication. The database is currently available at `http://pppdb.uchicago.edu`.

Using publications marked *relevant* and machine learning software, we were able to improve the publication selection process considerably, decreasing the number of reviewed publications that do not contribute to the χ database from 61.5 % to 13.1 %. These results may be improved by using alternative methods and by integrating polymer science insight gained through exploration of our data collection. For example, one could explore the utility of focusing on more frequently occurring methods as a publication filter prior to running the classifier. While this work is focused on χ, the steps required to collect a new similar and previously unmined property are straightforward via crowdsourcing; first the crawler must be configured to use a different keyword; the schema for the target property will guide the design of a new input form and the corresponding database table.

CHAPTER 5

SUPPLEMENTING AUTOMATED SCIENTIFIC INFORMATION EXTRACTION WITH HUMAN & MACHINE TASKS

5.1 Introduction

We now tackle the extraction of the glass transition temperature. As mentioned in Section 2.2.2, this is a slightly simpler property to extract as it consists mainly of a single temperature value for a polymer name. For this relation extraction task, we then explore increasing the level of automation and decreasing the level of trained and expert input. We propose a hybrid Information Extraction (IE) pipeline that first extracts candidate properties automatically and subsequently assigns various curation tasks to humans. The goal here is to maximize throughput and minimize the burden on human curators.

Our IE pipeline 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 pipeline to review candidate relationships. We apply this system to extract the *glass transition temperature* (T_g) of polymers, previously described in Section 2.2.2.

We used this IE pipeline to process 6 090 articles published over the last decade in *Macromolecules*. 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.

The rest of this chapter is organized as follows. Section 5.2 describes the design and implementation of our IE pipeline. Section 5.3 evaluates the accuracy of the various stages

in our pipeline. We conclude in Section 5.4.

5.2 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 [124], 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.

5.2.1 Our Pipeline

Figure 5.1 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

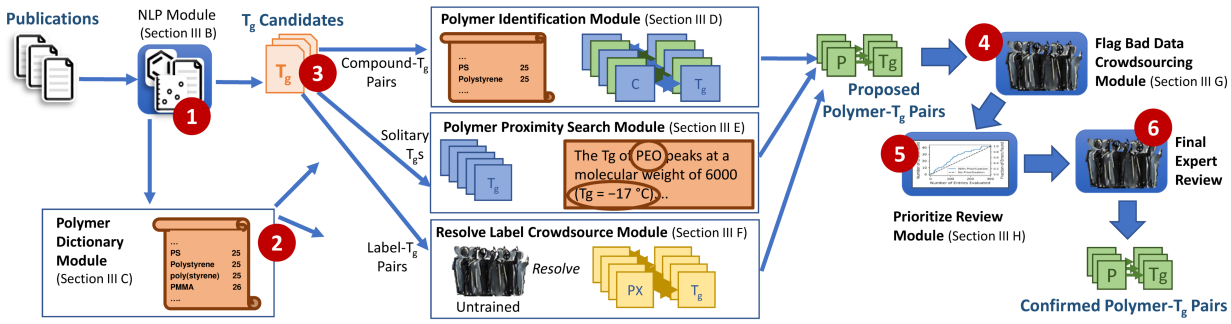


Figure 5.1: The six-stage hybrid IE pipeline, showing (1) the NLP Module, which identifies T_g candidates; (2) the Polymer Dictionary Module, which identifies polymer names in NLP output; (3) the three automated extraction and crowdsourcing modules used to process different forms of candidates; (4) the 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.

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. While in χDB, the crowd need to read the paper, find the required value and fill in an HTML form, here the system presents the user with some extracted data and a specific task to perform on this data. For example, it is significantly easier for a nonexpert to mark an automatically extracted 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 5.1 shows an example record.

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

Listing 5.1: This polymer–T_g record indicates that the polymer *poly(butyl methacrylate)*, also known as *PBMA*, has a T_g of 20 °C.

5.2.2 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 [54, 64] and other domain-specific [115, 112, 142, 76] content from text. Thus, we choose to extend an existing NLP toolkit, ChemDataExtractor [124], to extract T_g values from documents.

5.2.2.1 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 [75]. 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.

5.2.2.2 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.

5.2.3 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 [67] 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 [58] 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.

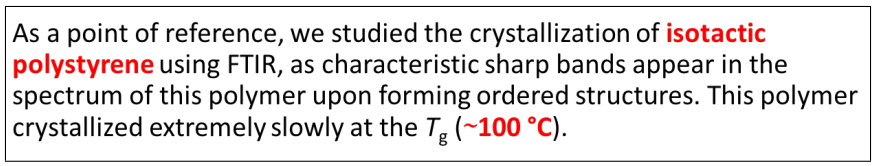


Figure 5.2: The NLP Module yields a solitary T_g record in this example text [90], as the corresponding compound is mentioned in the previous sentence. The Polymer Proximity Search Module disambiguates the reference and proposes *isotactic polystyrene* as a (correct) candidate match for the T_g.

5.2.4 Polymer Identification Module

For cases where compound-T_g pairs were identified using the NLP Module, the Polymer Identification 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 entries as non-polymers.

5.2.5 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 5.2.

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 5.2.7.

5.2.6 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 humans, 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.

5.2.7 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 of the task we use an expert crowd of polymer scientists.

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.

5.2.8 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 occurrence. 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 funneled to Stage 6 of the pipeline for expert review as shown in Figure 5.1.

5.3 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 evaluate each module in our pipeline.

5.3.1 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.

5.3.2 Natural Language Processing Module

Execution of the T_g–extended ChemDataExtractor NLP module described in Section 5.2.2 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.

5.3.3 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). These results motivate our subsequent focus on correctly extracting the polymer names and more broadly on scientific named entity recognition (see Chapter 6).

<u>*Precision*</u>: 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)" [62].

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*" [9]. 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.

5.3.4 Polymer Identification Module

To test the polymer name classifier described in Section 5.2.4, we selected 100 papers: the 50 used in Section 5.3.2 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. Here, we are not recognizing polymer names in entire documents, but simply searching for polymer acronyms stored in our dictionary of polymers, amongst the compound names from extracted compound—T_g pairs. These results confirms the value of linking and aggregating polymer names, synonyms and acronyms across papers into a dictionary. 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. This evaluation does not reflect expert review, but rather indicates that the modules extract names that have previously been linked to a rule-based recognition of polymer names and stored in our dictionary.

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*).

5.3.5 Polymer Proximity Search Module

Recall from Section 5.2.5 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 5.1 for a summary of the results.

Table 5.1: 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.

5.3.6 Crowdsourcing Modules

Recall from Sections 5.2.6 and 5.2.7 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 5.2 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 5.2: 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. 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.

5.3.7 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 5.3), 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.

5.3.8 Summary of Results

Table 5.3 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 non poly–T_g or 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.)

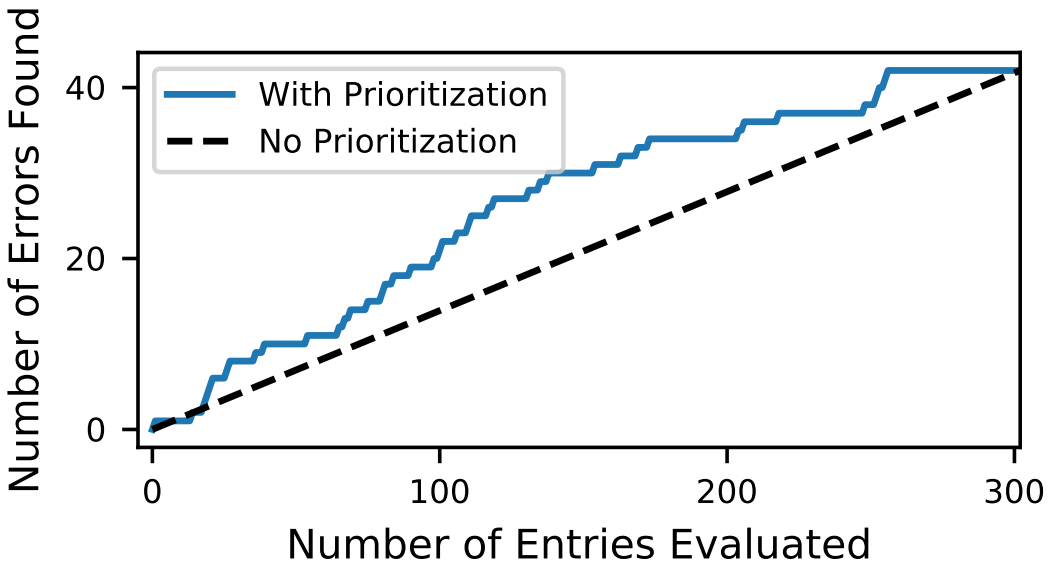


Figure 5.3: Results of prioritizing crowdsourcing. The blue, solid line shows the number of errors found as a function of the number of expert reviews if the entries are evaluated following our prioritization scheme. The black, dotted line shows the number of errors found if entries are evaluated in a random order.

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 5.3: 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**

5.4 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 built on top of an available domain-specific NLP software and subsequently tailored for a new property using a set of automated and human modules or tasks. Therefore, the IE T_g pipeline relies less on human and expert input than the χDB system, in that crowds of untrained users and experts are presented with some data to modify through a specific subtask of the extraction pipeline. We have used it 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 consensus 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. Our verified polymer–T_g pairs are available at both `http://pppdb.uchicago.edu` and `https://materialsdatafacility.org`. Expert scrutiny of these extracted data showed that accurate automated recognition of the polymer name remains a significant challenge in the extraction of the polymer properties, leading us to focus on expert-assisted, machine-learned scientific NER in the next chapter.

CHAPTER 6

GENERALIZABLE HUMAN-IN-THE-LOOP MACHINE LEARNED SCIENTIFIC INFORMATION EXTRACTION

6.1 Introduction

In χDB, we crowdsourced the extraction of the Flory-Huggins parameter; in the T_g pipeline we augmented automated NER with human expertise to extract the glass transition temperature. After expert review of χ values extracted via the χDB approach, the number of correct values extracted decreased from 388 to 263 experiment. While this number is still greater than that found in the *Physical Properties of Polymer Handbook* [41], it represents a decrease of $\sim 32\%$ mainly due to duplicates attributed to incorrect typographical errors, linking of names, synonyms and acronyms. In the T_g pipeline, 50% of the errors in the NLP toolkit output of polymer—T_g pairs was attributed to incorrectly identifying the polymer name. Having identified, polymer name recognition as a significant challenge (and prerequisite) in the extraction of polymer properties and as we aim to reduce human involvement, we now explore automated scientific entity recognition. Scientific NER remains challenging due to non-standard encoding, the use of multiple entity referents and other reasons discussed in Section 2.1. As previously mentioned, such challenges are not unique to polymer science and machine learning approaches for NER have been developed in medicine and biology [30, 78] and some in chemistry [64, 113, 79, 124]. These approaches rely on large, carefully annotated corpora of training data, a luxury not yet available in domains like polymer science. Hence, we present polyNER, a hybrid computer-human system for semi-automatically identifying scientific entity referents in text. PolyNER operates in three phases, first applying a fully automated analysis to produce an entity-rich set of candidates for labeling; then engaging experts to approve or reject a modest number of proposed candidates; and finally using the resulting labeled candidates to train a classifier. In both the first and third phases, it uses word embedding models to capture shared contexts in which referents occur. We experiment

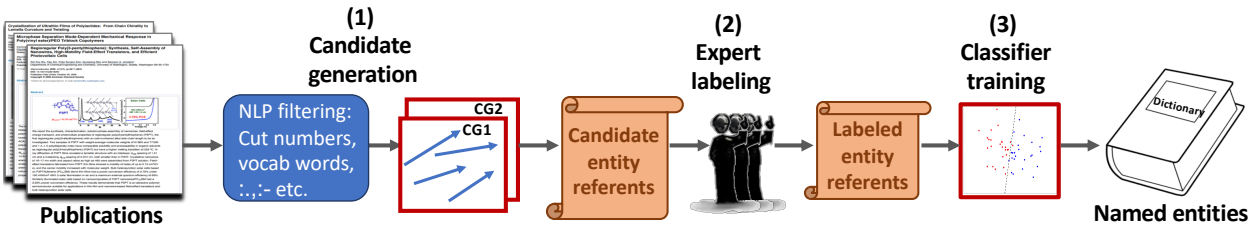


Figure 6.1: PolyNER architecture: showing (1) Candidate Generation, which produces candidate named entities from word vectors, (2) Expert Labeling, and (3) Classifier Training, which uses labeled candidates to train supervised ML models for identifying referents.

with active learning to reach NER performance on par with state-of-the-art tools with minimal input from experts [128]. PolyNER thus seeks to substitute the labor-intensive processes of either assembling a large manually labeled corpus or defining complex domain-specific rules with a mix of sophisticated automated analysis and focused expert input.

The rest of this paper is as follows. We describe design and implementation in Section 6.2 and discuss first evaluation of polyNER in Section 6.3. Sections 6.2.3 and 6.3.3 describe implementation and results for acquiring more labels via active learning to improve performance. We summarize polyNER in Section 6.4.

6.2 Design and Implementation

As noted in the introduction, previous approaches to scientific NER have relied on large expert-labeled corpora to train NER tools. Our goal in polyNER is to slash the cost of NER training for new domains by using bootstrap methods to optimize the effectiveness and impact of minimal expert labeling. As shown in Figure 6.1, rather than having experts review entire papers to identify entity referents, we use NLP tools to identify a set of promising candidate entity referents (Candidate Generation), then in an Expert Labeling step employ experts to accept or reject those candidates, and finally in a Classifier Training step use the accepted candidates to train an entity classifier

Before turning to the details of the polyNER implementation, we define an NLP filtering process that is used in various places in polyNER to filter out words that are unlikely to be

polymer referents. 1) We remove numbers. 2) Hypothesizing that names of scientific entities will not, in general, be English vocabulary words, we remove words found in the SpaCy and NLTK dictionaries of commonly used English words [29, 14]. (We manually remove common polymer names, such as polystyrene and polyethylene, from the dictionaries.) 3) We use SpaCy's part-of-speech tagging functionality to remove non-nouns. 4) We remove unwanted characters (e.g. ':', '.', ',', ':', '-') from the beginning and the end of each candidate, allowing us to recognize, for example, *polyethylene;* (which fails the exact string comparison test against "polyethylene"). 5) We remove plurals (e.g., polyamides, polynorbornenes), as they can represent polymer family names.

6.2.1 Candidate Generation

This first phase uses word vector representations, vector similarity measures, and minimal domain knowledge to identify a set of high-likelihood ("candidate") entity referents (names, acronyms, synonyms, etc.) in a supplied corpus of full-text documents (in the work presented here, scientific publications).

We first apply the NLP filtering process to reduce false positives. We next face the problem of determining whether a particular string is a polymer referent. String matching only gets us so far: for example, "polyethylene" names a polymer, but "polydispersity" does not. We need also to consider the context in which the string occurs. For example, the polymer name "polystyrene" in a sentence "The melting point of polystyrene is ..." suggests that X may also be a polymer in the sentence "The melting point of X is ...".

NLP researchers have developed a variety of *word embedding* methods for capturing this notion of context. A word embedding method maps each word in a sentence or document to a vector in an n-dimensional real vector space based on the linguistic context in which the word appears. (This mapping may be based, for example, on co-occurrence frequencies of words.) We can then determine the similarity between two words by computing the distance between their corresponding vectors in the feature space. Such vector representations can

be created in many different ways [114, 116]. Recently, the efficient neural network-based Word2Vec has become popular [92, 93].

We consider two measures of context-similarity between word vector representations in this step. CG1 uses the Gensim implementation of the Word2Vec algorithm [111] to generate 100-dimension vectors. CG2 employs an alternative FastText word embedding method that considers sub-word information as well as context [16, 68], allowing it to consider word morphology differences, such as prefixes and suffixes. Sub-word information is especially useful for words for which context information is lacking, as words can still be compared to morphologically-similar existing words. We set the length of the sub-word used for comparison—FastText's *n_gram* parameter—to five characters, based on our intuition that many polymers begin with the prefixes "*poly*" or "*poly(*." FastText produces 120-dimension vectors. Both CG1 and CG2 employ the continuous bag-of-words (CBOW) word embedding, in which a vector representation is generated for each word from an adjustable window of surrounding context words, in any order.

We compute a CG1 (Gensim) vector and a CG2 (FastText) vector for each NLP-filtered word in the input corpus, and also for a small set of representative polymer referents. Here we use polystyrene and its common acronym, PS, based on the assumption that polystyrene, as the most commonly mentioned polymer, provides a large number of example sentences in which polymers are mentioned. We can then determine, for each NLP-filtered word, the extent to which it occurs in a similar context to the representative polymers, by computing the similarities between the word's CG1 and CG2 vectors and those for polystyrene and PS. We discard the lower score for each of CG1 and CG2 to obtain two scores per word.

Having thus obtained scores, we then select as candidates, for each of CG1 and CG2, the N highest-scored words, with N selected based on the time available for experts.

We also use a rule-based synonym finder to identify synonyms of generated polymer candidates [119]. For example, if polypropylene has been identified as a candidate, then the expression "*polypropylene (PP)*" leads to PP being added to the candidate list.

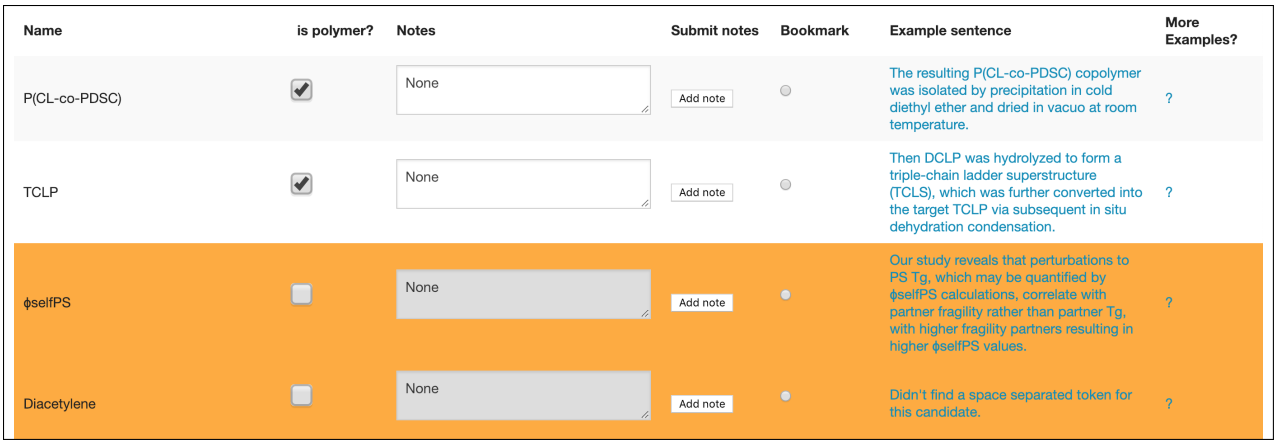


Figure 6.2: Web interface for expert review of candidates. The expert indicates whether the name (column 1) is a polymer (tickbox in column 2), providing notes if desired (column 3). Clicking on "?" delivers up to 25 more example sentences.

6.2.2 Expert Labeling

The previous step produces a set of candidate polymer referents: NLP-filtered strings that have been determined to occur in similar contexts to our representatives. We next employ an expert polymer scientist to indicate, for each such candidate, whether or not it is in fact a polymer referent. The expert simply approves or rejects each candidate via a simple web interface: a task that is more efficient than reading and annotating words in text. The interface (see Figure 6.2) provides the expert with example sentences as context for ambiguous candidates, and allows the expert to access the publication(s) in which a particular candidate appears when desired.

6.2.3 Candidate Discrimination

We next use the expert-labeled data to create an entity classifier. Many classification methods could be applied; we consider three in the work reported here: K Nearest Neighbor (KNN), Support Vector Classifier (SVC), and Random Forest (RF). Previous work has shown that KNNs perform reasonably well in text classification tasks [137]. SVC, an implementation of Support Vector Machines (SVMs), maps data into a feature space in which it

can separate the data into two or more sets. RF groups decision trees ("weak learners") to form a strong learner; it produces models that are inspectable, and includes a picture of the most important features. In each case, we use the 100 (Gensim) + 120 (FastText) = 220 dimensions of the two-word vectors as input features.

Given limited training and testing data, we evaluate all three classifiers, as implemented within scikit-learn [103], in Section 6.3.2. We envision that with more annotated data, we will be able to use neural-network-based classifiers.

6.2.4 Acquiring Additional Labels using Active Learning

As previously mentioned, our main goal in designing polyNER is to slash labeling costs by reducing the time and effort spent by experts to generate training data. Rather than labeling entire documents and phrases, annotators label proposed candidate entities to be classified. Earlier results show that with two hours of labeling we can achieve precision or recall (but not both) on par with state-of-the-art domain specific software, by selecting an ensemble of classifiers for discrimination [129]. The challenge then becomes efficiently engaging experts in order to obtain labels to improve the performance of our scientific entity classification. We do so by incorporating active learning with different sampling strategies into polyNER. PolyNER uses word representations and minimal domain knowledge (a few seed entities) to produce a small set of candidates for expert labeling; labeled candidates are then used to train named entity word vector classifiers. We integrate an active learning loop into polyNER's architecture to incrementally improve classifier performance.

In order to explore whether the use of word vector coordinates as features can accelerate the learning process, we define and compare three alternative sampling strategies: a random strategy that we use as a baseline, and two NLP-based filtering methods. We also apply these methods against two different candidate pools, one set of unlabeled nouns and another set of approximately labeled nouns deemed *similar* to commonly used known entities from our corpus. We describe our sampling strategies and approximate labeling in more details

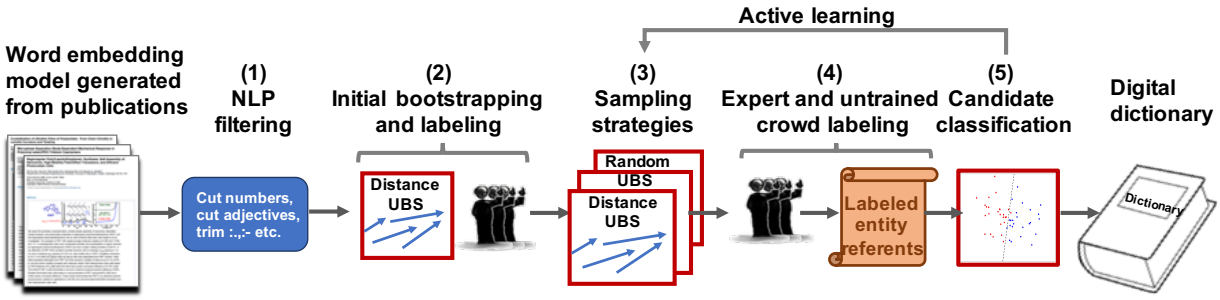


Figure 6.3: PolyNER system showing the different phases of polyNER including the NLP-filtering step, the initial bootstrapping and labeling phase as well as the newly integrated active learning loop to classify scientific named entities. The active learning loop is described in more detail later in Figure 6.4.

in this section. The general architecture of polyNER is illustrated in Figure 6.3. We also describe the labeling process, and the training and testing configuration for our word vector classifiers in Section 6.2.4.6.

6.2.4.1 Computing Word Embeddings

We start use the Gensim continuous bag-of-words (CBOW) implementation of the Word2Vec algorithm to generate vectors as it is light-weight and easy-to-use [111]. Prior parameter tuning indicated that window size and vector size did not have a significant impact on the yield of polymers (less than 5%) on initial bootstrapping (see Section 6.2.4.3 below). Nevertheless, for slightly higher yields, we set the Word2Vec `size` parameter to 100 and the `window_size` to 2; where `size` is the size of the vector, and `window_size` is an adjustable window of surrounding context word used to compute each embedding.

6.2.4.2 NLP-Filtering

PolyNER's NLP filtering preprocessing step removes strings that are unlikely to be polymer referents by performing simple operations such as masking numbers and removing unwanted characters (e.g. ':', '.', ',', ':', '-') from the beginning and the end of each candidate.Here, we also use SpaCy's part-of-speech tagging functionality to remove non-nouns. Hence, we are

left with pre-processed nouns that do not appear in (SpaCy and NLTK) English dictionaries. Note that these steps are generalizable and applicable to multiple science fields. We refer to the set of words that results when these filtering operations are applied to our corpus as the *NLP-filtered candidates*. This set is the output of step 1 in Figure 6.3.

6.2.4.3 Initial Bootstrapping and Labeling

NLP filtering reduces the number of entities to be considered, and increases the target vs. non-target entity ratio. However, there still remain a large pool of potential candidates from which entities are to be selected, of which, in our experience, roughly 5% are polymer names. In order to avoid presenting experts with mostly negative examples, hindering meaningful classification, we boost the number of polymer entities in the first batch of candidates to be annotated by selecting strings with low word vector distance (see Section 6.2.4.1) from a set of *seed entities*: words that are observed to occur frequently in a subset of publications, or that are suggested by experts. We discuss this distance metric in more detail below. Based on preliminary experiments, we set the size of each batch of strings to be labeled to 200, or about an hour of expert time. We then train the initial classifier on this bootstrap set, using 80% of the data for training and 20% for testing. We subsequently used three different sampling strategies for following classifications.

6.2.4.4 Sampling Strategies

We implement three sampling strategies, which we refer to as *Random*, *Uncertainty-Based Sampling (UBS)* and *Distance Uncertainty-Based Sampling (Distance UBS)*. We apply each of these strategies to our NLP-filtered candidates to determine which candidates to present to experts for labeling.

Random Strategy Here, we randomly select 200 of the NLP-filtered candidates.

Uncertainty-Based Sampling using NLP-Filtered Candidates (UBS) Our second

strategy applies maximum entropy sampling to the NLP-filtered candidates. As previously mentioned, maximum entropy selection is an uncertainty sampling method that identifies data points for which a classifier predicts outcomes that lie near the decision boundary between classes. Thus, when predicting whether or not a word vector represents a polymer, maximum entropy arises when the classifier assigns equal probability to the polymer and not-polymer cases. As we have two classes, this equal probability is 0.5. We use the classifier to obtain a probability p for each NLP-filtered candidate. We select the 200 entries for which p is closest to 0.5 as our sample.

Uncertainty-Based Sampling using NLP-Filtered Distance Candidates (Distance UBS) Our third strategy is identical to UBS, except that it works with just a subset of the NLP-filtered candidates, namely the 10 000 that are closest to a set of seed entities.

We use the word embeddings introduced in Section 6.2.4.1 to capture this notion of context, and vector distance between word vectors as a measure of similarity. Whether or not this approach works in practice will depend on whether polymer names are in fact used in consistent contexts that is captured by our word embedding vectors.

We can then determine, for each NLP-filtered word, the extent to which it occurs in a similar context to the seed entities, by computing the similarities between the word's vector and those for our seed entities. As we explain in Section 6.3.3.2, we experiment with one and more seed entities; when dealing with multiple seed entities, we use the lowest distance score for ranking candidates.

6.2.4.5 Bootstrapping

The UBS and Distance UBS sampling methods use a classifier to determine which NLP-filtered entities should be chosen next for expert labeling. This classifier must be trained, and thus we need an initial set of entities to bootstrap this process. We could choose NLP-filtered entities at random for initial labeling, but that choice is unlikely to perform well due

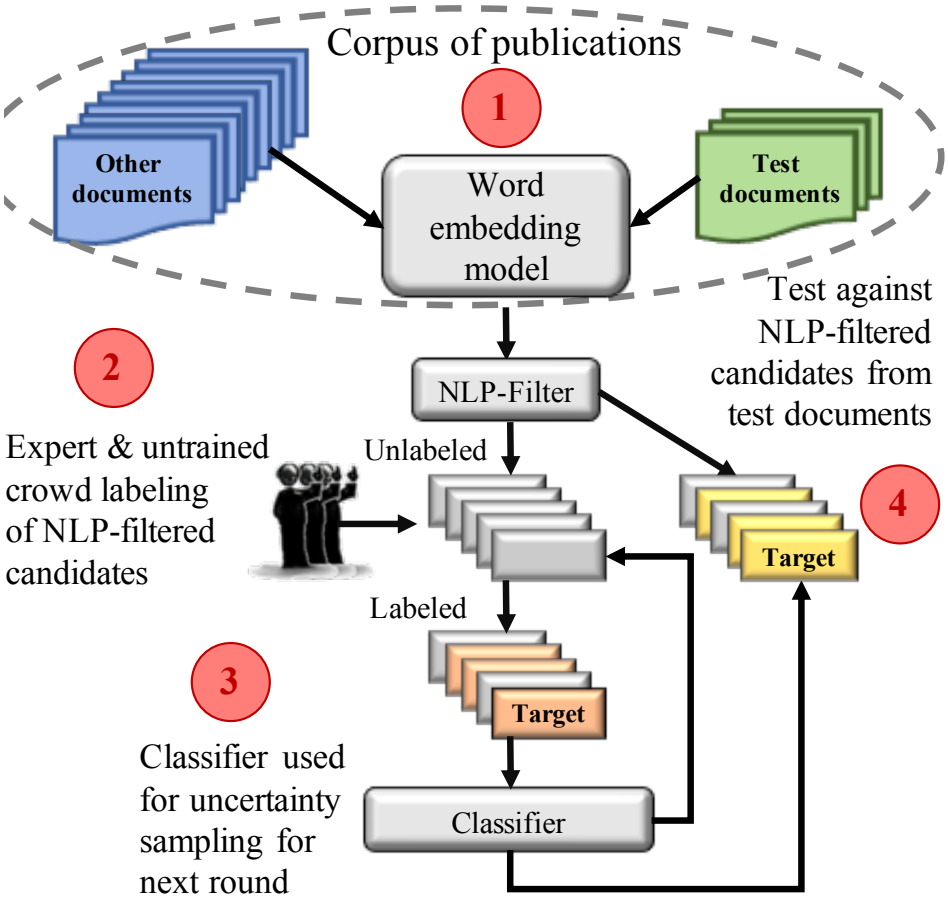


Figure 6.4: Active learning experiment set up. 1) We generate an unsupervised word embedding model using our entire corpus. 2) We propose NLP-filtered candidate entities to untrained and expert annotators before classifying their word vectors. 3) We select the best-performing classifier for uncertainty-based sampling of labels for the next round of active learning. 4) We evaluate this word vector classifier on all NLP-filtered words from a set of test documents.

to the low proportion of polymers (just 5%) in the NLP-filtered corpus. Instead, therefore, we create an initial bootstrap set comprising the 200 NLP-filtered entities that are closest, in word distance vectors, to a set of seed entities. (Recall that we use a batch size of 200 based on an estimate of 60 minutes of expert time required for labeling.) We then train the initial classifier on this bootstrap set, using 80% of the data for training and 20% for testing.

6.2.4.6 Active Learning Loop

We now discuss our active learning process. As discussed in Section 3.6.3, the basic idea here is that we repeatedly select a set of 200 candidate entities (a "sample") for expert labeling, based on what we have learned from previously labeled entities. We run this process independently with the Random, UBS, and Distance UBS sampling strategies, in order to compare their performance.

Use and Evaluation of Classifiers Not specified in Section 6.2.4.4 is the nature of the classifier that the UBS and Distance UBS strategies use to estimate the probability of each entity in the NLP-filtered corpus (or, for Distance UBS, the 10 000-entity subset of the NLP-filtered corpus) being a polymer. (The Random strategy does not use a classifier for sampling, as it selects candidates at random.) As we have no prior knowledge of the distribution of target entities in the vector space, we consider seven distinct classifiers in each round of the active learning process: the scikit-Learn [103] implementations of Decision Tree, Gradient Boosting, K-Nearest Neighbor (KNN), Logistic Regression, Linear Support Vector Machine, Naive Bayes, and Random Forest. In each case, we use the word embedding for each string as input features. In each round i, we train these seven classifiers on the sample data gathered in rounds j, $j<i$, and then use the classifier with the highest recall to determine the p scores that UBS and Distance UBS use when assembling their 200-entity sample in that step. (We use recall, or retrieving a maximum of targets, as a measure of performance, because we want to favor extracting a higher number of targets, potentially

requiring additional curation, over obtaining fewer correct targets.) The 200 entities in the new sample are passed to experts for annotation, and the annotated data are added to the set of training data used in the next learning round.

Untrained and Expert Labeling We engage two domain experts to annotate the candidates generated by the *UBS* and *Distance UBS* sampling strategies. Each expert annotates one strategy; we also perform crosschecking for 10% of the first batch of labels, to get a measure of agreement between experts, with results reported in Section 6.3.3.3. Experts use a web interface (see Figure 6.2) to approve or reject candidates, a task that is far more efficient than reading and annotating words in text. The interface provides example sentences as context for ambiguous candidates and allows the expert to access the publication(s) in which a particular candidate appears.

We aim to reduce the amount of costly expert time used to obtain labels. Therefore, for our baseline of randomly sampled NLP-filtered nouns, we experiment with a two-phase review process. Tokenization is one of the largest sources of error for scientific entities such as polymers, which contain characters such as ':', '(', '–', ',' etc. Tokenization can also generate incoherent tokens from text, equations, captions, etc. Such obvious non-candidates can be fairly easily detected by non-experts. For example, an untrained human annotator may be able to recognize that '$d\Sigma/d\Omega)(Q$' is not a polymer name, and thus save time for the experts. Hence, we assigned two graduate student labelers to curate the candidates generated by the random sampling strategy, which are less likely to contain target entities. We asked these untrained labelers to reject obvious non-candidates via the previously mentioned web interface. Our experts then reviewed the remaining candidates, indicating for each whether it is in fact a polymer referent.

6.3 Evaluation

We report on studies in which we evaluate the performance of both the unsupervised Candidate Generation step and various classifiers trained on the labeled data that results from

the Candidate Generation and Expert Labeling steps. We then report on the active learning results.

6.3.1 Evaluation of Candidate Generation Methods

6.3.1.1 Dataset

We work with two disjoint sets of full-text publications in HTML format from the journal *Macromolecules*: P100 comprising 100 documents with 22,664 sentences and 508,391 (36,293 unique) words or "tokens," and P50 comprising 50 documents with 12,148 sentences and 270,514 (22,571 unique) tokens. For later use in evaluation, we engaged six experts to identify one-word polymer names in P100. They find 467 unique one-word polymer names.

Recall that the polyNER candidate generation module employs two candidate generation methods, CG1 and CG2, plus a rule-based synonym finder. We evaluate the performance of both the complete polyNER candidate generation module and its CG1 and CG2 submodules by comparing the sets of candidates that they each generate from P100, both against the 467 one-word polymer names identified by experts in P100, and against two other polymer name extraction methods, CDE and CDE+, plus a sixth compound method formed by combining polyNER with CDE+. We use exact string-matching between candidates and expert-identified names (all lower cased). Results are in Table 6.1. For each, we evaluate extraction accuracy in terms of precision, recall, and F_1 score. Recall is the fraction of actual positives that are labeled correctly and precision the fraction of predicted positives that are labeled correctly; F_1, the harmonic average of precision and recall, reaches its best value at 1 and worst at 0.

The first two methods considered, CDE and CDE+, serve as baselines. CDE is the previously mentioned state-of-the-art Python package that extracts chemical named entities and associated properties and relationships from text [124] (recall from Section 3.4.1). As CDE aims to extract all chemical compounds, not just polymers, it serves only as a demonstra-

tion of an alternative approach in the absence of a polymer NER system. Its recall is high at 74.5% but its precision is, as expected, low at 8.7%. We extended CDE into CDE+ to extract T_g in Chapter 5; the modifications include the addition of manually defined polymer identification rules. CDE+ achieves a higher precision of 42.2% but a slightly decreased recall of 68.3%. These results emphasize the difficulty of automatically recognizing complex entities such as polymers.

Rows 3 and 4 show performance for CG1 and CG2 when employed independently. Recall that polyNER performs NLP filtering before applying CG1 and CG2. The filtering step eliminate all but 6,878 of the 36,293 unique tokens in P100. Recall also that CG1 and CG2 each assign a score to each of the 6,878 remaining words based on their context-based vector similarities to polystyrene and PS, and select the N highest scoring. In this evaluation, we set N=500.

CG2, which takes word morphology into account, achieves higher precision and recall than does CG1 (41.8% vs. 15.6% precision and 44.8% recall vs. 16.7% recall for CG2 and CG1, respectively). CG2 retrieves more words starting with "poly" (67% of the 500 candidates vs. only 4% for CG1) while CG1 retrieves more acronyms (38% of the 500 candidates contained more upper than lower case letters, vs. 23% for CG2). CG1 returns more false positives. While character level information is useful for unseen words, or in this case for words lacking context information, we cannot dismiss the use of CG1. Authors often introduce polymer names and subsequently use acronyms more heavily, especially for long names. The facts that CG1 returns more acronyms and that there is likely more context information about acronyms, suggests that the performance of CG1, albeit lower, is solely based on context information.

Row 5 shows results for the complete polyNER candidate generator: that is, the combined CG1 and CG2 candidates plus their rule-based extracted synonyms. This method achieves 61.2% recall and 26.0% precision, producing an entity-rich set of candidates without any domain-specific rules and without any (tedious, time-consuming, and costly) expert-

annotated corpus of polymer names. We are encouraged to observe that polyNER retrieves polymers not extracted by CDE: the combined recall for PolyNER ∪ CDE+ (row 6 in the table) is 81.6%—higher than CDE itself. This result suggests that polyNER's candidate generation module can be used not only to annotate automatically a diverse set of polymers based on context, but also to improve on the results of more sophisticated hybrid rule- and ML-based NER tools.

Figure 6.5, which shows every word in P100 in FastText vector space, illustrates the challenges and opportunities inherent in differentiating between polymer and non-polymer word vectors. The polymer names (in red and green) form two rather diffuse clusters that overlap considerably with non-polymers (in blue). Interestingly, the subset of polymer names that are acronyms (the red points) are clearly clustered.

Table 6.1: Results when polymer candidates are extracted from our test corpus, P100, via different methods. For each, we show true positives, false positives, false negatives, precision, recall, and F-score.

#	**Method**	**Total**	**TP**	**FP**	**FN**	**Precision**	**Recall**	**F_1**
1	CDE	3,994	348	3,646	119	8.7%	74.5%	15.6%
2	CDE+	755	319	436	148	42.2%	68.3%	52.2%
3	CG1	500	78	422	389	15.6%	16.7%	16.1%
4	CG2	500	209	291	258	41.8%	44.7%	43.2%
5	PolyNER	1,099	286	813	181	26.0%	61.2%	36.5%
6	PolyNER ∪ CDE+	1,495	381	1,114	86	25.4%	81.6%	38.8%

6.3.2 Evaluation of Candidate Discrimination

We next evaluate how well classifiers trained on expert-labeled output from the Candidate Generation phase perform when applied to full-text documents. Here, we make use of our second dataset, P50, to train and test our classifiers, and P100 to validate the trained classifiers.

Before training our classifiers, we need a set of expert-labeled candidates. Thus, we first apply the CG1 and CG2 methods of Section 6.2.1 to P50, generating a total of 897

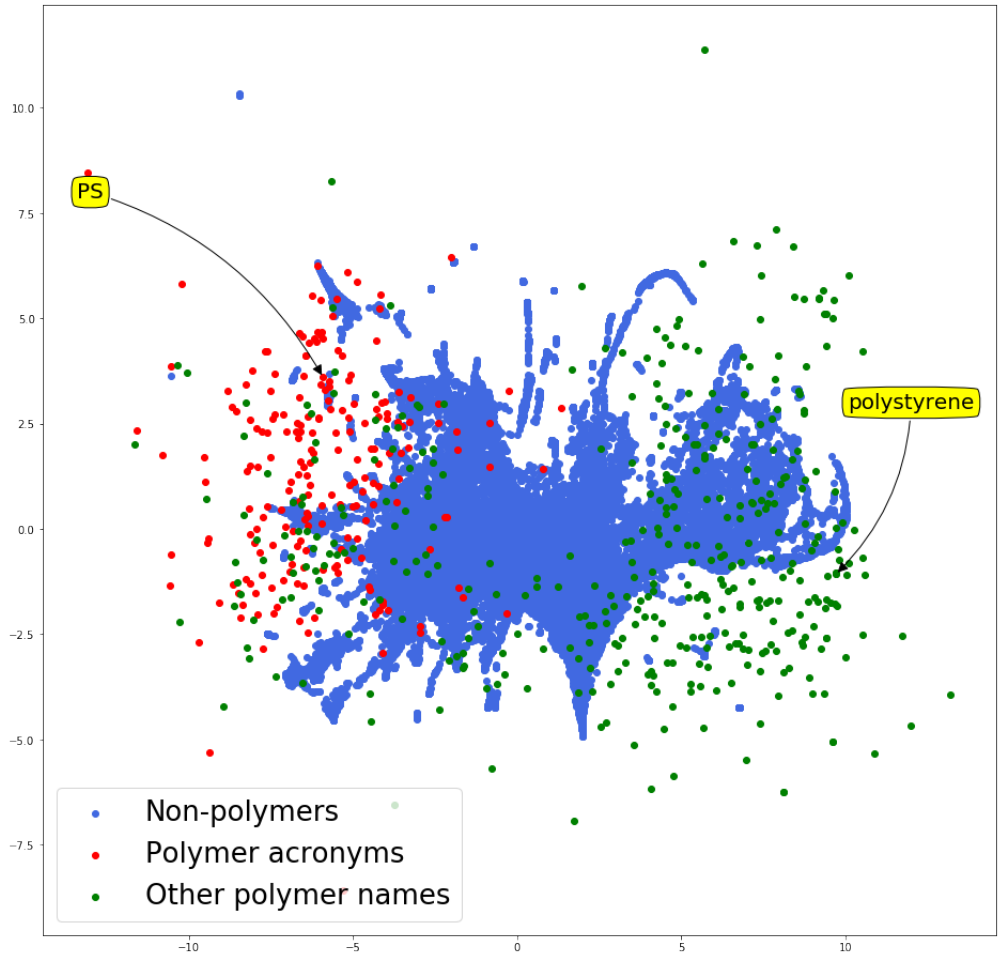


Figure 6.5: A two-dimensional representation of all words in P100, generated with the scikit-learn implementation of t-distributed Stochastic Neighbor Embedding (t-SNE) [86]. Of the words identified by experts as polymers, we show acronyms in red and non-acronyms in green; all other words are blue. We label our two representative words. (The t-SNE plot, a dimensionality reduction technique used to graphically simplify large datasets, reduces the 120-dimensional vectors to two-dimensional data points. The axes have no "global" meaning.)

unique candidates: 500 for CG2 and 466 from CG1, of which 69 overlapped. (We do not apply the rule-based synonym finder here.) Then we employ an expert to label as polymer or non-polymer each of those 897 candidates, producing a new dataset that we refer to as P50-labeled. Note that this task is quick work for the expert, as only 897 words need to be evaluated: the total time required was two hours. This expert review identifies 260 (29.0%) of the 897 as polymers.

6.3.2.1 Training and validating the classifiers:

We next use the 897 expert-labeled words to train our three classifiers. We use 90% (807) for training and hold out 10% (90) for validation.

The left-hand side of Table 6.2 shows the performance of the different trained classifiers when applied to the P50 hold-out words. Note that performance here is defined with respect to how well the classifier does at predicting the expert labels assigned to the polyNER-generated candidates—not how well the classifier identifies *all* polymer referents in P50, as we do not have the latter information.

All three classifiers obtain between 66.7% and 100.0% precision and between 28.6% and 57.1% recall. SVC achieves the highest recall and F_1 score. The lower part of the table ("combined classifiers") shows that combined classifiers can improve performance. The 3-of-3 method achieves the highest precision (100.0%) but lowest recall, as one might expect. The $\geq$2 method also achieves 100.0% precision but with a higher recall (42.3% vs 28.6%). The $\geq$1 method has the lowest precision but the highest recall at 57.1%.

6.3.2.2 Testing the trained classifiers:

We test the trained classifiers by applying each to all 9,656 NLP-filtered nouns extracted from P100 and comparing the resulting polymer/non-polymer labels against our ground truth of polymer names extracted from P100 by experts. Results are in the right-hand side of Table 6.2. RF achieves the highest F_1 score (46.1%) with 51.0% precision and 42.1% recall.

Table 6.2: Results when various classifiers (trained on expert-labeled P50 candidates) are applied to P50 holdouts (left) and P100 (right). The results in the bottom two rows are copied from Table 6.1 for ease of comparison.

Classifier	Validation on P50 holdouts			Testing on P100		
	Precision	**Recall**	**F_1**	**Precision**	**Recall**	**F_1**
KNN	75.0%	42.8%	54.5%	9.5%	77.8%	16.9%
SVC	66.7%	57.1%	61.5%	16.8%	76.5%	27.5%
RF	100.0%	28.6%	44.4%	51.0%	42.0%	46.1%
Combined						
3-of-3	100.0%	28.6%	44.4%	52.7%	39.1%	44.9%
≥2	100.0%	42.3%	60.0%	22.8%	66.6%	34.0%
≥1	57.1%	57.1%	57.1%	9.1%	90.7%	16.5%
CDE				8.7%	74.5%	15.6%
CDE+				42.2%	68.3%	52.2%

While recall is relatively low (fewer entities retrieved), precision is significantly better than that achieved by CDE+. We observe also that combined classifiers can improve precision (52.7%) at the expense of recall, or significantly increase recall (90.1%) at the expense of precision. Users can thus trade off precision and recall, in each case exceeding those achieved by the rule-based CDE+ system.

These results are based on only limited training data: just 897 labeled words, of which 260 are polymers. We view the effectiveness of the classifiers trained with these limited data as demonstrating the feasibility of using small amounts of expert-labeled data to bootstrap context-aware word-vector classifiers. Importantly, this whole process was both inexpensive and generalizable to other domains. Candidate generation was fully automated and involved no domain knowledge besides the two representative words, polystyrene and PS. Labeling required just two hours of an expert's time. Classifier training was again automated and involved no domain knowledge.

6.3.3 Active Learning Evaluation

We first report on a study in which we evaluate the generation of candidate entities using vector distances from representative (frequently used) entities. We then discuss the results of

initial classification and subsequent four rounds of active learning using multiple word vector classifiers and our three sampling strategies: random, UBS, and Distance UBS. Finally, we experiment with word representations enhanced with character-level information using FastText [16, 68].

We evaluate extraction accuracy in terms of precision and recall.

6.3.3.1 Dataset

We work with a corpus of 1690 full-text publications in HTML format from *Macromolecules*, a relevant journal in polymer science. These documents comprise 381 947 sentences and 9 229 417 (253 195 unique) words or "tokens," of which 23 205 pass the NLP filter of Section 6.2.4.2. We use the same P100 documents used in Section 6.3, from which six experts have manually 467 unique one-word polymer names. We use these 467 names as a gold standard in subsequent subsections; we automatically label all NLP-filtered strings from the P100 test set using these manually extracted names.

6.3.3.2 Seed Entities

Recall from Section 6.2.4.1 that polyNER uses the Word2Vec word embedding tool to compute a word vector for each word. In order to maximize the number of actual entities in the dataset—and the ratio of target to non-target entities—in the initial set of labels, we explore how the choice of seed entities impact the number of target entities retrieved. While we cannot expect meaningful classification using only positive examples, given the imbalance in the whole dataset, we aim to select the Word2Vec parameters that yield the highest ratio of polymers in this initial batch of candidates.

In the experiments that follow, we use the 467 gold standard polymer names identified by experts in our P100 test set to evaluate performance with different seed entities. Specifically, for each choice of seed entities that we want to evaluate, we determine the 10 000 NLP-filtered

words with vectors closest to the seed entity vectors, and report what fraction of the 467 gold standard names are included in that 10 000. We use lower-case exact string matching between the gold standard polymer names and the proposed distance candidate strings to determine if a candidate is a polymer.

In Section 5.2.4, we built a dictionary of polymer names by using a rule-based approach and aggregating synonyms across ChemDataExtractor records. (A record consists of all information found about a chemical entity in a document.) Here we use this dictionary to identify the 10 most frequently occurring polymers in our corpus and their acronyms. We assume that frequent polymers provide a large number of sentences that illustrate context in which polymers are commonly used. Hence, we first test the most frequent, the three most frequent, and the ten most frequent polymers as seed entities. We also experiment with including and excluding their acronyms as additional seed entities. (Note that this modest set of 1, 3 and 10 seed entities could also be suggested by an expert.)

Rows 1–6 of Table 6.3 shows the results for these experiments. When using *polystyrene* (the most frequently used name) as a seed entity, the candidates contained 33.6% of the 467 gold standard polymers. We note a 2% increase in the fraction of polymers retrieved when using both *polystyrene* and *PS*, when compared to using *polystyrene* alone. The fraction of polymers increases by 10% when we use three representative entities (the three most frequent polymers in our datasets are *polystyrene*, *poly(methyl methacrylate)*, and *polyethylene*), but by less than 1% when using 10 instead of three entities. These results suggest that there is little value to using more than a few seed entities.

To further explore whether using larger numbers of seed entities may increase the fraction of polymers retrieved, we conducted a second set of experiments. We have built a database of polymer properties (χDB) in Chapter 4. Our corpus of 1690 publications included 111 out of 175 χDB polymers. We also scraped CrowDB, which lists some polymers and their properties at `http://polymerdatabase.com/` for polymer names; 32 out of 295 scraped polymer names were found in our corpus. We measure how many of our gold standard

Table 6.3: Fraction of gold standard polymer names in the 10 000 entities that are closest, by word vector distance, to various sets of seed entities.

#	Seed entities	Fraction of polymers extracted
1	Polystyrene	35.6%
2	Polystyrene, with acronym PS	37.7%
3	Three most frequent polymer names	46.9%
4	Three most frequent polymer names, with acronyms	48.0%
5	10 most frequent polymer names	46.5%
6	10 most frequent polymer names, with acronyms	48.4%
7	χDB polymer names	46.7%
8	crowDB polymer names	36.4%

polymers are identified when these 111 and 32 polymer names are used as seed entities, with results shown in rows seven and eight of Table 6.3. These results confirm that using more entities does not increase the yield of polymers. Thus, in all subsequent experiments, we use the three most frequent polymers and their acronyms as seeds.

6.3.3.3 Labeling

We conduct some experiments to estimate labeling time. We ask two polymer scientists to label 200 candidates from a subset of our corpus. One expert reports 30 minutes, the other 45 minutes. We overestimate the time to label 200 candidates at one hour of expert time. Based on user feedback, we also improve the labeling Web interface after the above-mentioned test rounds to further facilitate and speed up labeling. For example, we increase the number of example-sentences available to provide context, to decrease the number of occurrences in which experts have to look up original publications for candidates. We also increase the size of checkmarks to make it easier to reject erroneous candidates. In the initial labeling round, we perform crosschecking for 10% of the batch of labels. We confirm agreement between labels for all but one of 20: an agreement rate of 95%.

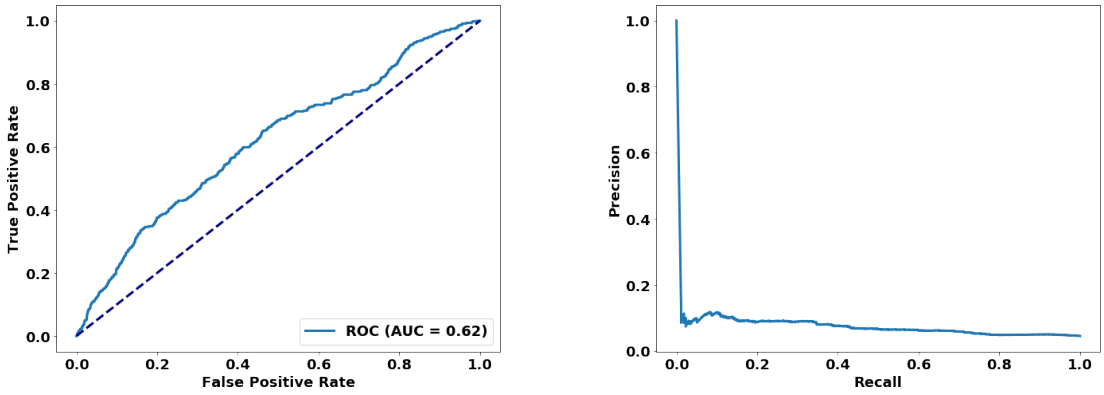


Figure 6.6: ROC (left) and PR (right) curves for KNN model for the initial classifier. The PR curve shows that precision is low regardless of recall, indicating that we need more data.

The Initial Classifier Recall from Section 6.2.4.5 that we use an initial set of 200 NLP-filtered entities that are close to seed entities to bootstrap our labeling process. Once those data are labeled by our experts, we use 80% to train a KNN classifier, validating on the remaining 20%. We then test this *initial classifier* against all 9656 NLP-filtered candidates in the 100-document test set, which as noted in Section 6.3.3.1 contain 467 polymers. Results are shown in Figure 6.6. Its Receiver Operating Characteristic (ROC) curve shows better-than-random behavior, with an area under curve (AUC) of 0.62. However, in our application, correctly predicting non-polymers is not as important as correctly identifying our targets. Therefore, we also plot the Precision Recall (PR) curve to show the tradeoff between precision and recall. While the AUC for the initial classifier is above random performance (0.5) its PR curve shows poor precision, regardless of recall. In Section 6.3, we found that we could achieve better performance with more labels (897), suggesting that a KNN classifier begins to learn with more data [129]. However, 160 labels (80% of 200) is not yet enough.

6.3.3.4 Comparing Sampling Strategies

After the initial round of labeling, we experiment with the three sampling strategies described in Section 6.2.3: Random, UBS, and Distance UBS, performing four rounds of active learning for each. Results, in Table 6.4, show no improvement in the first two rounds for any strategy:

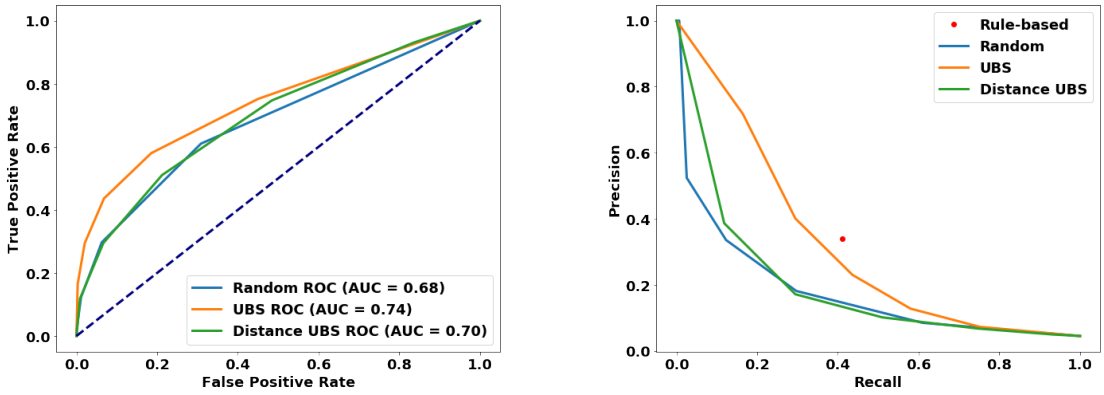


Figure 6.7: ROC (left) and PR (right) curves after four rounds. The ROC curves for two active learning strategies, UBS and Distance UBS, show significant improvements, achieving AUCs of 0.74 and 0.70, respectively: significantly better than the 0.62 achieved by the initial classifier in Figure 6.6. Random achieves an AUC of 0.68. PR curves also show improvements relative to the initial classifier, with all three strategies lifting away from the bottom corner, indicating discriminative capacities. In both types of plots, UBS outperforms Distance UBS and approaches rule-based performance based solely on context information and under five hours of expert input.

precision remains under the initial precision of 6.5% for all. However, in the third round, we observe increases in precision, an improvement that is sustained in the fourth round for UBS. Figure 6.7 shows ROC and PR curves for the three strategies after four rounds. The AUC for UBS is 0.74 and that of Distance UBS is 0.70. The PR curves for both are improved (lifting away from the lower left corner of the graph) over the first round, with active learning performing better with UBS than with Distance UBS. When tested against our gold standard of 467 one-word polymer names, the KNN classifier achieves 18.2% precision and 45.6% recall. We notice that even the random strategy PR curve is improved (away from the initial PR curve and close to the Distance UBS curve), indicating that the NLP-filtering alone is enough to enable the learning process after 1000 labels. We also note that KNN was most-often selected across strategies and iterations, likely due to its inherent nature to optimize locally and our direct focus on finding similarities between observations.

We conclude that while the sampling step helps ensure that the classes are balanced in the initial batch of labels, restricting ourselves to just distance candidates, as is done by Distance UBS, does not yield better results than using active learning with all NLP-filtered

candidates (UBS). Intuitively, basic UBS can find *useful* instances (target and non-targets) to be labeled from the entire word embedding space, while examples from Distance UBS are clustered around the seed entities that may be collocated in that space.

Table 6.4: Precision and recall relative to the gold standard for the initial classifier (round 0) and the classifiers trained also with the increased data obtained in each of four learning rounds, 1–4.

Round	Metric	Strategy		
		Random	**UBS**	**Distance UBS**
0	Precision	6.5%		
	Recall	19.1%		
1	Precision	3.8%	3.2%	5.3%
	Recall	0.29%	93.6%	56.8%
2	Precision	1.5%	3.8%	5.4%
	Recall	1.5%	46.4%	10.1%
3	Precision	6.0%	**21.2%**	3.9%
	Recall	44.6%	**40.0%**	84.3%
4	Precision	12.3%	**18.2%**	7.2%
	Recall	33.3%	**45.6%**	51.9%

We selected seed entities based on their frequency in our corpus. This observation suggests that we could also study how the choice of seed entities impact of the performance of the classifier during the active learning process. Note that with limited training data and based solely on context, the classifier retrieves 45.6% (more than one third) of the gold standard polymers with a precision of 18.2%, after five hours of expert labeling. For comparison, an attempt to extract polymer names using the rule: *if the name contains "poly" extract it as a polymer*, gets recall of 41% and precision of 34% on the same dataset. We conclude that with our relatively small dataset (based on context-only information from entire documents of unstructured and uncurated text), we are able to achieve close to rule-based performance, using active learning and little labeling.

6.3.3.5 Active Learning Labels + Character-Level Embeddings

After just 1000 labels, the context-based classifier using active learning applied to NLP-filtered candidates achieved performance comparable to rule-based performance, but not quite as good as the polymer-enhanced CDE+. With the goal of further improving polyNER performance, we experiment with the use of an alternative word embedding model, FastText, which uses word representations enriched with sub-word (character-level) information. Because FastText considers sub-word information as well as context, it can consider word morphology differences, such as prefixes and suffixes. Sub-word information is especially useful for words for which context information is lacking, as words can still be compared to morphologically similar existing words. We set the length of the sub-word used for comparison—FastText's `n_gram` parameter—to five characters, based on our intuition that many polymers begin with the prefixes "poly" or "poly(." Therefore, we generate a FastText word embedding model, and generate character-enhanced vectors for our UBS-labeled candidates.

Next, we train a KNN classifier using vectors for the candidates labeled through active learning from the NLP-filtered candidates (the active learning strategy identified as best-performing in Figure 6.7). We test the classifier against NLP-filtered nouns from our 100-document test set. KNN classifier performance improves when using these word vectors, achieving 29.7% precision and 81.9% recall, comparable to those achieved by CDE (see Section 3.4.1). CDE's recall is high at 74.5%, but its precision for polymers is, as expected, low at 8.7%, as it does not incorporate polymer knowledge. In Figure 6.8, we show the PR curve for the FastText vector classifier and also results for the polymer-enhanced CDE+ which achieve 42.2% precision and 68.3% recall on the same test set. We achieve higher recall than CDE and CDE+ using labels from UBS and FastText vectors and in-between (higher than CDE and lower than CDE+) precision.

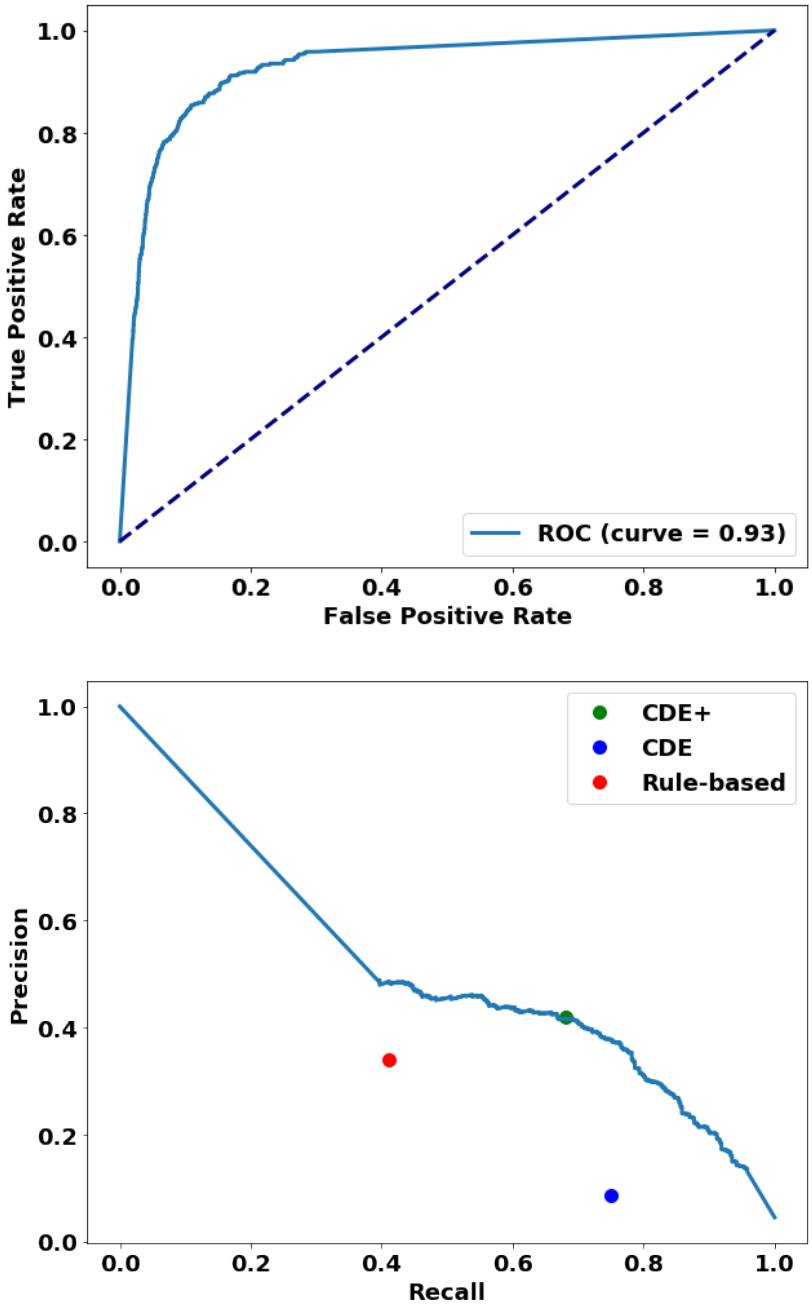


Figure 6.8: ROC curve for KNN model trained using active learning labels and word representations enriched with character-level information(top). At the bottom, PR curve for KNN model trained with active learning labels and word representations enriched with character-level information. Results for CDE+ are also shown. Note: PR curves, like ROC curves, are obtained by varying the threshold of probability that separates classes; straight lines occur when several points have similar probabilities and changing the threshold yields identical precision to recall ratio.

6.3.3.6 Impact of Quantity and Quality of Data

After achieving CDE+ performance, we investigate the impact of the quantity and the "quality" of data, where higher quality data are more likely to contain scientific entities. We recreate the Active Learning + Character Embedding classification experiments with two different word embedding models:

- The first dataset is re-used from Section 5.3.1. It is a set of 6090 full-text publications in HTML format from *Macromolecules*. The hypothesis here is that with more data, we increase the number of sentences that contain polymers, capture more context information, and hence generate better-defined embedding vectors for previously annotated entities. Note that the 4500-document increase from the dataset used in Section 6.3.3.1 adds a considerable number of sentences that do not contain polymers (see below).

- The second dataset is generated using a subset of the sentences from the 1690 full-text publications used in Section 6.3.3.1. We run CDE+ on each sentence from this dataset and keep only sentences in which CDE+ has identified a polymer (12% of the sentences contain polymers). This step of course assumes a domain-specific technique to decrease noise (non-relevant sentences) and refine the dataset to create the embeddings.

Results show that increasing the dataset size increases performance surpassing CDE+ performance as shown in Figure 6.9 (top). The classifier achieved 35.2% precision and 86.1% recall when evaluated against the 100-document test set, noticeably improving from the previous 29.7% precision and 81.9%.

We observe that performance also increases when using a smaller, higher-quality sentences(only 12% of the sentences selected using domain-specific NLP software). Using this dataset, the classifier reaches 40.1% precision and 89.1% recall. These results confirm the intuition that the quality of training data is as important, if not more important, than the quantity of training data. We achieve 89.1% recall with more than one third of proposed

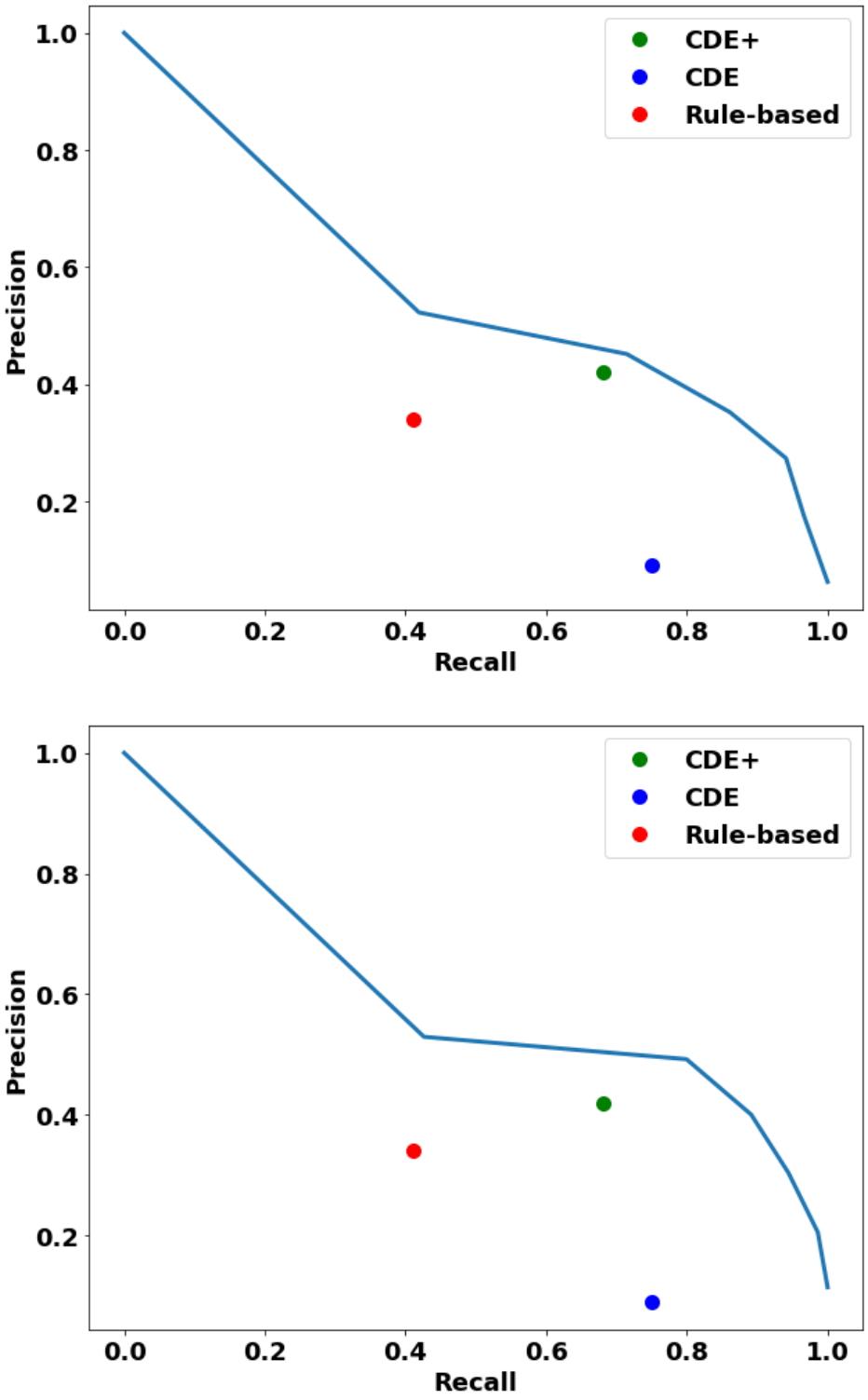


Figure 6.9: Precision Recall curves for KNN model trained with active learning labels and word representations enriched with character-level information generated from a larger corpus (additional 4500 documents, top) and CDE+-filtered sentences (bottom).

entities being actual polymers, significantly facilitating final human curation and validation process.

An important remark is the fact that a relatively high number of false positives persists. Upon examination of the results, we attribute this phenomenon mainly to the limitations of our tokenizer and our initial focus on one-word polymer names. For example, some examples of false positives include: *P(OEGA* and *(PtBS*, which all contain partial polymer names—and were leniently considered positives during our expert annotations—but were not an exact match to any gold standard polymer names. A better tokenizer would separately identify PtBS as a polymer in itself and a multi-word approach would attach the missing subsequent words to each of these examples. Other examples of false positives raise an important question as to the quality and nuances of human annotations. For instance, *PtBS/τa*, *χi—P3HT*, *(PM)m* all accurately contain polymer names. The actual polymers have been altered and are described with extra components (characters attached) that do not appear in the list of gold standard polymer names. Both types of errors can be partially addressed by a lenient evaluation (allowing partial matches between candidates and true polymers) along with the ability to recover missing parts by exposing text around certain candidates via an interface for further curation. Finally, other errors are more difficult to completely eliminate as they include terms that often precede polymers such as concentrations, polydispersity index and prefixes (*Mono-2*, *Sulfonated*, *(10K/ns)*, *(PDI = 1.2)*, *(NPS = 163, NP3HT = 105*), etc.) or plurals (PSSs, Poly(lactide)s) that have purposely been left out by expert annotators. We revisit the topic of an ideal human-computer IE system and interface to further curate these false positives in Chapter 8.

6.3.3.7 Discussion and Future Work

PolyNER achieves CDE performance using labels obtained via active learning (UBS sampling strategy) and FastText vectors. We attribute the increased performance to the character embedding enhancement, which not only recognizes "poly" (and yields more names based on

this n-gram comparison), but also filters out more anomalous candidates (preceding or following polymer names) generated during tokenization and missed by the filtering steps, such as "A_mB_n" and "*Mw/Mn=1.36*." In other words, the classifiers of character and (context-based) word embedding vectors perform better than classifiers of only context-based word embedding vectors. Given this result, one may wonder whether the active learning process itself could benefit from using this enhanced vector embedding. To determine whether this is the case, we repeated the active learning experiment using the entire corpus of NLP-filtered candidates and classifying FastText (enhanced) vectors instead of Gensim vectors at each round. However, performance was worse than random.

These results suggest that character-level information enhances classifier performance only once a certain threshold of context information has been captured by the embedding. We explain this observation as follows. In FastText, the portion of the word embedding vector generated by using context varies depending on how much context is available in the entire corpus. For words deemed to have enough context, vectors do not include any character-level information. At the other extreme, for previously unseen words, the embedding is generated based solely on character n-gram information and comparison to other words in the corpus. During the active learning process, candidates to be labeled by experts are selected by using maximum-entropy-based uncertainty sampling: that is, words for which prediction probability is similar for target and non-target. Such candidates are more likely to lack context and thus have vectors that use mostly character-level information. As a result, the expert is often presented with nearly identical candidates (e.g., PS13k/PMMA12k, PS214k/PMMA12k, PS31.6k/PMMA12k), which hinders the learning process as these candidates are located in close vicinity in terms of the full (character and context) word embedding space. In other words, in this full space, while their uncertainty measure is comparable, these examples are not *diverse*, where diversity is a measure of the distance of the examples to each other or previously labeled instances [21]. One solution to explore in future work would be to impose a diversity constraint on the candidates, for example by using batch active learning [121].

6.4 Conclusion

Despite much progress in NLP, scientific named entity recognition (NER) remains a research challenge. The lack of access to large amount of expert-annotated training data impedes the adoption of recent machine learning techniques in certain scientific applications. PolyNER is a generalizable system that can efficiently retrieve and classify scientific named entities. It uses word representations and minimal domain knowledge (a few representative entities) to produce a small set of candidates for expert labeling; labeled candidates are then used to train named entity classifiers. We show that using natural language processing techniques, we can bootstrap a word vector classifier of scientific entities. PolyNER can achieve either 52.7% precision or 90.7% recall when combining classifiers: a 10.5% improvement in precision or 22.4% in recall over a well-performing hybrid NER model (CDE+) that combines a dictionary, expert created rules, and machine learning algorithms. PolyNER's architecture allows users to tradeoff precision and recall by selecting which classifiers are used for discrimination. One out of every four candidates identified by our current polyNER prototype is in fact a polymer. This enrichment relative to the relative paucity of polymers in publications significantly reduces the effort required by experts.

Using more labels obtained via uncertainty-based active learning, and word embedding containing character-level information in addition to context, polyNER achieves performance comparable to CDE+. PolyNER exceeds CDE+ performance when processing more data (additional 4500 articles), or high-quality data (pre-selecting sentences more likely to contain chemical compounds) and reaches. PolyNER was trained on data annotated using approximately five hours of expert time and minimal untrained crowd input obtained via active learning. While we note that CDE+ is intended to extract polymers and their properties, our work highlights the potential of using minimal amount of data and focused expert input in order to enable machine learning techniques for previously unmined scientific entities.

CHAPTER 7

GENERALIZABILITY OF APPROACHES

In this chapter, we discuss how our various approaches represent necessary steps to bridge the gap between state-of-the-art NLP software and the extraction needs of certain scientific applications. We describe how these approaches are generalizable and applicable in other fields.

7.1 Scientific Crowdsourcing

Generalizing the χDB work involves training crowds in particular topics for assisted manual extraction of scientific facts. While χDB is focused on the Flory-Huggins parameter, the steps required to collect a new property are straightforward. The crawler must be configured to use a different keyword. Domain experts provide a schema for a new target property, which can be used to design new Web forms and corresponding database tables. Our results show that guiding manual extraction and making it more efficient has several benefits: (1) trained crowds can be less expensive than experts even when done outside classroom settings. In other words, requiring expert knowledge for supervision vs. for extraction saves costs, (2) partial manual extraction is unavoidable in cases such as χ where a minimum of six metadata fields are required and free-text is needed to give context and meaning to extracted values, (3) preliminary manual extraction is needed to guide future (semi-) automated extraction. For example, in our work, the χDB class, showed that the number of values encoded solely in Figures was negligible and that of the 4 temperature-dependent forms, only 2 were widely used in the literature. Such efforts can also serve as guidelines for the community at large. For example, as a follow-up to the χDB class, we are exploring ways in which authors can submit χ values to our website in a straightforward manner. More broadly, χDB is part of a growing body of work that aims to address the challenges posed by the rapidly increasing amount of scholarly literature via hybrid crowd-machine integration platforms [27, 40, 32].

SciCrowd, for instance, emphasizes the challenges and motivations for designing a crowd-enabled, self-organizing system towards achieving a higher level of engagement by researchers and citizen scientists [32]. Such hybrid approach can be particularly fruitful in crowd science projects to refine machine-extracted metadata while providing evidence on demand by using automatic classification techniques enabled by human crowd workers who can process, filter, classify, and verify the information [31].

7.2 Supplementing NLP Software

ChemDataExtractor is only one of several chemical extraction software (OSCAR, TmChem, ChemSpot [64, 79, 113]) that may require human and computer modules to be adapted to different subfields. Similar software has emerged from the BioNLP workshop [71, 97, 143, 96], a CHEMDNER-like challenge in biology, that may benefit from supplementary modules to improve accuracy of such NLP software. The automated search and identification of entities associated with properties in the previous sentence, can be extended to N previous sentences, or to the current paragraph. The human modules for matching labels to their corresponding entity is also easily applicable to any domains, in which names can be long and complex. Given the significant cost of manual curation, we can also investigate more advanced methods to prioritize where human effort should be used such as validation of extracted data via machine learning models. For example, for properties like T_g, which are common and frequently reported, there are a large number of machine learning models for predicting T_g values [38, 91, 77, 139]. These models could be retrained, 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. Similar efforts can be applied to prioritize human review using existing dictionaries and databases in bioinformatics, which benefits from wider access to databases for example.

7.3 Generalizable Scientific NER

To illustrate the generalizability of polyNER we turn to a very different named entity extraction problem: identifying dataset names in free text. This is an important problem because while data plays an essential role in research, a lack of standards for data citation makes discovering relationships between publications and datasets challenging [89]. We apply polyNER to the task of automatically identifying free-form text references of datasets in social science publications and evaluate its accuracy against the manually curated relationships available from the Inter-university Consortium for Political and Social Research (ICPSR).

To illustrate the aim of this task, consider the following paragraph which states that "Longitudinal Study of American Youth" dataset was used in a study.

> "With this deficit looming, efforts to understand the factors that contribute to degree attainment have redoubled. This article is a piece of that effort. By analyzing data from 3,279 individuals who participated in the **Longitudinal Study of American Youth**, this study examines the relative importance of personal, peer, and parent educational expectations in the degree attainment of students 15 years after high school graduation." [17]

We first use the Python Natural Language ToolKit (NLTK) `sent_tokenize` tool to perform sentence extraction [14]. For candidate generation, we apply a slightly different approach than we did for polymer names as we do not have a canonical dataset name with which we can compare. Instead, we rely on the regularity of dataset names in social science: dataset names are usually sequences (i.e., n-grams) of upper case and proper nouns. As evaluating all n-grams in a sentence ($n \in [1, \text{number of words in the sentence}]$). would result in an exponentially large number of candidates, we apply the simple heuristic of selecting consecutive capitalized words. (Note that polyNER ignores non-capitalized stopwords, such as "of" and "and," even if they are part of a proper noun.) For each candidate we use

PolyNER's Gensim Candidate Generation method (Word2Vec CBOW model) to compute word vectors. In the final candidate discrimination phase, we train polyNER's KNN classifier to identify dataset names based on their word vectors. Unlike identifying polymer names, we consider here multiword dataset names. However, the polyNER classifier expects input vectors to have the same length. Thus, we average the individual words in a vector to compute a vector of the same length for every candidate phrase.

ICPSR provides a large corpus of manually curated papers and associated datasets in the field of sociology. It has indexed over 72,000 papers that reference 60,566 unique datasets. We sampled this dataset to obtain the full text of 6,368 papers that were published by Elsevier and for which full text was available via an API. We split these 6,368 papers such that 80% are used for training and 20% are reserved for testing. We first evaluate the accuracy of sentence classification. Initially, we found the performance to be poor (precision: 14.2%, recall: 51.9%, F_1: 22.3%), however we noted that the training dataset was significantly unbalanced with 18,682 positive examples and 1,268,239 negative examples. After randomly downsampling the negative examples by a factor of 10, a common approach for such problems, we observed precision of 58.8%, recall of 90.3%, and F_1 score of 71.2%. We use the hand curated mapping from papers to dataset names provided by ICPSR as the gold standard to evaluate the candidate generation and discrimination. Automatic evaluation is difficult because of possible variations of dataset names so instead we manually compared our extraction result from 300 papers against the gold standard from ICPSR and observed precision of 59.6% and recall of 58.7%. As before, we also evaluated the performance of the classifier in isolation, when using only labeled candidates, and observed precision of 90.8% and recall of 97.2%. This result suggests that even though the classifier had high accuracy in discriminating dataset names from other candidate phrases, the ultimate accuracy of our model was limited by the candidates that were fed into the classifier (as was the case for polymer NER). For example, in a paper, a dataset may be referenced as "US Quarterly Census of Wages," but the manually annotated gold standard dataset name provided by ICPSR

would state the full title "Quarterly Census of Wages (2010Q1) – United States Bureau of Labor Statistics." In such cases, polyNER is not able to extract the full title of the dataset because it was simply not present in the paper. Nevertheless, these results (precision of 59.6% and recall of 58.7%) demonstrate the great potential of using polyNER to identify scientific entities in a previously unexplored field using context-based vectors and limited training data.

CHAPTER 8

FUTURE WORK

In this chapter, we discuss future work as well as some open-ended ideas that emerged from designing and implementing our various scientific fact extraction approaches.

8.1 Scientific Crowdsourcing

In the case of χDB, we can not only repeat the experiment for new complex relations to save cost and guide subsequent automated extraction, but also investigate ways to guide community text mining efforts. We can promote our Polymer Prediction and Property Database (PPPDB) website as a community resource from which scientists can easily and freely load extracted properties for machine learning models for example. The more interest PPPDB generates, the more incentive it provides for scientists to submit newly measured properties to our database. Hence, providing easy-to-use APIs to upload data onto the website is an important, straightforward follow-up task to χDB. We are also providing data to the Materials Data Facility (MDF) [15], which aims to create a thriving materials and machine learning ecosystem. In the case of the hybrid T_g IE pipeline and as discussed in the previous chapter, our human and computer modules aren't specific to the domain and only aim to increase the accuracy of existing NLP software. Interesting next steps would apply the pipeline to a different domain, and add automated modules such as model-based ranking of extracted values to prioritize human review for example. A more end-to-end evaluation of added value through computer and human modules would more formally illustrate the concept of the hybrid IE pipeline. We discuss this idea in more details in Section 8.4.

8.2 Scientific Entity Relations

We have shown that polyNER achieves performance comparable to CDE for one-word scientific entities. While we have explored averaging context-based vectors for multi-word entities

to identify datasets in the social science literature, one could implement a similar labeling and classifying process to detect two, then three-word polymer names for example. As previously mentioned, with our scientific entity tagger, machine learned relations extraction is achievable using software such as Snorkel and Snuba [133]. While there are only a handful of rule-based descriptions of a polymer, there are multiple ways to define a property. The most obvious rule for tagging polymers is to search for "*poly*". When looking for co-polymers, some additional IUPAC rules may be implemented as we have in the polymer dictionary module (e.g. find "—co—", —graft—), but this only helps for a subset of polymer names and cannot alone be used to automatically label text for polymer NER. When it comes to properties however, we do have examples of regular expressions that could be translated into rules for systems like Snorkel. For example, if a temperature is in the same sentence as the words "melting point", "melting pt.", "T_m", "m. pt." etc., then extract it as a potential polymer–melting point temperature pair. Such rules provide Snorkel with multiple user defined functions to identify melting points which it can use to generate training data, learn the accuracy of each definition and extract the pairs.

8.3 Generalizable Named Entity Recognition via Word Embedding and Language Modeling

In polyNER, we have used 6,090 papers, and we were able to train a classifier of context and character word embedding vectors to identify vectors for scientific entities with a better performance than CDE. We have begun to explore how the amount of training data and the quality of sentences impact performance. We could also use more advanced word embedding vectors generated using pre-trained language representations such as Glove, ELMo and BERT [51, 52, 104, 105, 37]. Previous work has demonstrated improved performance in language understanding, text classification and other NLP tasks using language model pre-training [108, 60]. There are two main strategies for applying pre-trained language rep-

resentations to downstream tasks: feature-based and fine-tuning; ELMo uses task-specific architectures that include the pre-trained representations as additional features [105] OpenAI GPT introduces minimal task-specific parameters, and is trained on the downstream tasks by simply fine-tuning the pretrained parameters [108]. The main improvement in BERT, besides the sheer amount of training data (800M words from the BooksCorpus [73] and 2,500M words from English Wikipedia), is the use bi-directional language representations. While the question remains as to how much information about scientific entities (e.g., polymers) could be found in Wikipedia, we can use transfer learning from non-scientific corpora to acquire informative vectors for all non-target entities and better recognize targeted scientific entities amongst them. Other improvements specific to polyNER include learning multi-word context by exploring and classifying multi-word embedding vectors. We are currently exploring the use of polyNER-labeled data to annotate text for other NER approaches, such as bidirectional long short-term memory (LSTM) models, or other neural networks with *attention* mechanism [51, 101], which would capture local (sentence level) context and also address the current limitation of one-word named entity recognition. An attention mechanism allows the network to automatically select key words and/or sentences, which is likely describing the extraction target. Such an approach seems especially helpful when using noisy labels, i.e. labels acquired via weak supervision [81, 65] In doing so, we may consider the different types of names for scientific entities as different classes (long names, acronyms, labels, family names) to identify; hence, we can either explore classifying one type of names, or multiple entities simultaneously with multi-task learning.

8.4 Hybrid IE Experiment Design

In all three approaches we have used humans with different levels of expertise from trained undergraduate students, untrained graduate students, and domain experts. In both the hybrid IE T_g pipeline and polyNER, we have experimented with multi-phase human reviews. In the former, a consensus of untrained graduate students resolved 78.6% of missing labels in

polymer–T_g pairs, leaving the 21.4% of missing information to be extracted by experts. In polyNER, labels for the random strategy were reduced by up to 80% via a two-phase process in which an untrained graduate student rejected obvious non-entities. A similar process could be implemented even in cases where labels are not randomly selected and still decrease the load for domain experts. A major challenge however remains as to determine 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. Even in current best-performing NLP software there are some errors and as the number of publications processed increases, careful scrutiny of the data will become costly and eventually unworkable. Hence, an open-ended question for this type of hybrid machine-computer endeavors remains: how to maximize accuracy while minimizing cost of different crowd input? In future work we can explore this question as a formal optimization problem. In other words, we will work on a more rigorous approach to automatic partitioning and assignment of extraction tasks by applying techniques from *optimal experiment design* [45, 42, 44] to maximize the accuracy of extracted data while minimizing the time and cost of human involvement.

- 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 entities or relations and considering the rarity and completeness of those values, and then measure how dataset value changes with each automated and crowdsourced task.

- Assign levels of difficulty to tasks based on estimated completeness, accuracy of data to be curated. Such measures should also consider the amount and type of the data to be processed and/or the information needed to complete the task, to help decide where to crowdsource various tasks. For example, more data to be collected indicate a more complex property, hence a more difficult task; free text means more work to

extract and review and should be assigned a higher level of difficulty than extracting numbers.

- 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.

A hybrid human-computer approach to extract scientific entities requires replication or access to crowdsourcing platforms such as Amazon Mechanical Turk for simple tasks as well as access to multiple state-of-the-art domain specific NLP toolkits. It also requires mechanisms to train crowds and request expert input. Such approaches would involve consider of human-computer interaction, as user interfaces for untrained crowds have to be straightforward, while experts may require the ability to request more information and access to the original publications from which value were extracted. An ultimate interface, would also serve as a validation tool by showing both, what has been extracted and how/where it was extracted from to further alleviate the task on the expert. Such an interface would partially address the very interesting question as to how accurate the human gold standard annotations really are. As discussed in Section 6.3.3.6, we suspect some of the candidates proposed by polyNER are correct but differ slightly from what experts have reported. A partial-matching comparison, along with surrounding text for each candidate, will assist human in recognizing and correcting potential entities. This step would remain essential for verification regardless of the final level of automation.

CHAPTER 9

SUMMARY OF THE RESULTS AND LESSONS LEARNED

Motivated by the long-term goal to create a digital handbook of polymer properties, we have investigated three hybrid crowd-machine approaches that represent varied levels of automation and involvements from human with different levels of expertise. In this final section, we summarize our results and discuss lessons learned more broadly.

9.1 Summary of the Results

We first designed and implemented χDB, a hybrid human computer-system that extracts the Flory-Huggins (or χ) parameter from scientific literature. Our work to date has extracted 237 measured χ values for blends of 63 unique polymers. We trained students during two academic quarters to extract more values than the 134 χ values for blends of 41 unique polymers found in the *Physical Properties of Polymers Handbook* [41]. We were able to collect values reported after the 2007 publication of the *Handbook* (84 of our χ values are from 2010 to 2015). Our more exhaustive search also led us to find earlier values not reported in the *Handbook*. Using publications marked *relevant* and machine learning software, we were able to improve the publication selection process considerably, decreasing the number of reviewed publications that do not contribute to the χ database from 61.5 % to 13.1 %. While a more systematic comparison of cost and accuracy remains to be conducted, our results emphasized the potential for using an machine-assisted crowdsourcing approach to create and maintain a digital database of complex scientific relations. Such automatically guided crowd extraction is necessary in cases which require accute expertise to identify novel measurement methods for example or to distinguish between entities that may refer to related but not identical concepts (polymer family name vs. family instance name).

It also serves to discover how previously unexplored scientific relations are described in the literature in order to guide subsequent partially automated extraction. Finally, some

version of machine guided crowd extraction will remain necessary in cases where there is a lack of consensus around a particular relation and/or part of the information to be extracted is subjective (e.g., free text to describe method).

The second approach, the T_g IE pipeline, which extracts the glass transition temperature of polymers, takes a different, more automated approach to hybrid machine-human scientific information extraction. In this case, the motivating idea was to implement a system around existing domain-specific natural language processing and machine learning toolkits. Studies have shown that such toolkits are not always easily transferable to new domains [74], prompting other fields to create their own combination of dictionary-based, rule-based, and machine learning approaches, which in turn require training data. One solution then was to augment available domain-specific NLP software with a series of human and computer modules to adapt and apply them to a new domain. In particular, we showed how one could build upon ChemDataExtractor, which aims to extract all chemical entities from unstructured text, to extract polymers. We added new polymer-identification rules and aggregate ChemDataExtractor output to build an accurate dictionary of polymer names. We designed a polymer proximity search module that correctly identified missing polymers for half of those T_g values without polymers, the idea being to simply search for one element of a relation in the proximity of the automatically detected element. 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 fourths (78.6 % labels resolved by non-experts using consensus method). Our initial results on automated ranking for prioritized human review showed that even a simple method for assessing the quality of extracted data can effectively increase the impact of human curation. The hybrid T_g pipeline offered 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 re-

sults demonstrated the considerable potential of combining automated and crowdsourcing modules to extract scientific facts from literature in an efficient and cost-effective manner.

The third hybrid human-computer approach, polyNER, was motivated by challenges encountered in χDB and the T_g IE pipeline. Indeed, we found that identifying polymer names was a major challenge in extracting polymer properties. This challenge was exacerbated by the lack of training data, which prohibited the direct application of state-of-the-art machine learning and natural language processing extraction techniques. Hence, we designed and implemented polyNER, a generalizable system that can efficiently retrieve and classify scientific named entities. PolyNER used word representations and minimal domain knowledge (a few representative entities) to produce a small set of candidates for expert labeling; labeled candidates are then used to train named entity classifiers. We showed that using natural language processing techniques, we could bootstrap a word vector classifier of scientific entities. Using an ensemble of classifiers and word embedding software, polyNER allowed users to trade off precision for recall, and achieved performance comparable to the state-of-the-art chemistry-aware natural language processing software. Using additional labels obtained via uncertainty-based active learning, polyNER exceeded the performance of a hybrid NER model (CDE+) that combines a dictionary, expert created rules, and machine learning algorithms and was trained on the CHEMDNER corpus: a collection of 10,000 PubMed abstracts that contain a total of 84,355 chemical entity mentions labeled manually by expert chemistry literature curators, following annotation guidelines specifically defined for this task [75]. PolyNER was trained on data annotated using about five hours of expert time and minimal untrained crowd input obtained via active learning. This work demonstrated that using minimal amount of data and focused expert input, we can successfully extract previously unmined scientific entities from unstructured text.

We envision that the work presented in this thesis, combined with other work in related areas will ultimately lead to the integration of all three approaches in a single platform that will automatically assess the value of data, cost of crowds and levels of expertise to assign

tasks and subtasks to multiple crowds and computer modules to extract scientific facts from the literature.

9.2 Lessons Learned

In order to guide such future endeavors, we summarize lessons learned from this thesis. In our experience, scientific fact extraction began and ended with the experts and there were five main types of tasks to be performed by machines and humans including crowds with different levels of expertise:

1. Schema design: The first step in scientific fact extraction was to clearly define the facts (e.g., named entities or entity relations along with metadata information) in order to select an appropriate extraction approach. Schema design was a task better suited for the domain experts, who not only had in-depth prior knowledge about the information to be extracted, but also a well-defined application in mind for the expected output data. Our work demonstrated that trained crowds can efficiently assist experts in confirming and refining initial schemas by discovering how the information has historically been published in the literature. This preliminary crowdsourcing work also served in assembling useful datasets to be later used as training data for automated IE.

2. Document classification: Here the task is to identify publications or sections in publications that contain the entity to be extracted. Since machines are capable of processing volumes of text faster than their human counterparts, the classification of relevant publications, (or paragraphs such abstracts, results sections etc.) is a fitting computer task. We have shown that such classification can be used to efficiently prioritize human review and considerably decrease the amount of non-relevant publications to be processed.

3. Extraction: In our work, this task was performed by both humans and machines. Based on our experience, we observed three different types of scientific facts, each potentially

requiring additional metadata.

(a) Named entities: The goal here is the identification of names of scientific entities such as polymers, proteins and diseases for examples. Ideally, this task would be performed by computer techniques that capture information based mainly on context and word morphology when applicable. However, in our experience, due to the nature of scientific articles (long sentences, long multi-word names with special characters etc.), knowledge cannot always be easily transferred from one domain to another without generating new training data. Our work suggested that untrained crowds can efficiently recover missing information (parts of entity names) and establish links between entities, labels and other referents. Experts could then be called upon to decipher controversial data only as crowds encountered difficulties. With more training data and the current advances in NLP techniques such as hierarchical attention neural networks and new language models, scientific NER is moving towards more automation and minimal expert input.

(b) Relations: Similarly, while there exist machine learned approaches to entity linking and relations extraction, these methods aim to teach the computer relations that are often straightforward to human brains (*married to*, *employed by*, *located in* etc.) In science, relations may be more complex and require some expertise, however trained crowds and even untrained crowds can recognize and correct relationships such as "T_g *of*" or "*melting point of*". In our application for example, while computer techniques were efficient at extracting such relations in straightforward context contained in single sentences, extraction failed when entities spread across long sentences and paragraphs, or when sentences included discussions and comparisons between multiple entity relations. Our work suggested that crowds can successfully establish missed connections without expert knowledge.

(c) Processes: Here, we make a distinction between processes and the joint extraction and linking of multiple named entities. Scientists may be interested in complex

procedures that combines varying numbers and types of entities and require relatively high numbers of metadata fields rather than one or two parameters. For example, in some articles, the newly designed polymer was not named; instead its synthesis was described in a paragraph. While NLP techniques may identify the relevant paragraph in publications, even extract parts of the synthesis procedure, there would likely remain gaps between the automatically extracted data and the desired output information. Such cases are more suitable for exploring a more complex combination of automation with untrained and trained crowd input. Once, the correct section of the article has been identified via text classification, untrained crowd can be directed to extract temperatures and other concise facts; trained crowd workers may review this initial data, correct and/or add metadata information, leaving only the extraction of nuanced information to be performed by experts.

4. Labeling: In this case, both automated and human input can be used. Weak supervision methods can automatically annotated some data, while experts may only provide labels for ambiguous cases to improve classification performance. Untrained and trained crowds can simply aim to decrease the amount of information to be examined by the experts—thereby decreasing the highest labeling costs—by eliminating obvious erroneous data proposed during the weak supervision process.

5. Validation: Validation of extraction models is best performed by experts until satisfactory classification accuracies of entities and relations are achieved. For example, based on budget and costs discussions, expert can review a modest percentage of the automatically extracted results and verify that models achieve their desired accuracy in practice. High accuracy predictive models can then help prioritize human review as the volume of publications and extracted data increases. The need for experts is not totally eliminated but greatly reduced by such models as outliers may not always

indicate errors but results from interesting, novel scientific studies. In the case of the glass transition temperature for examples, while there exist mathematical models to predict this property, an extreme temperature may indicate novel results such as new materials, or special measurement conditions.

In summary, the work presented in this thesis illustrates how to lower the costs, increase the throughput and accelerate the pace of state-of-the-art scientific facts extraction—which currently includes fully-manual extraction from the literature to populate databases, due to its challenging nature and the occasional lack of training data—by combining computer tasks with crowd input. We propose that human expertise is an essential, unavoidable piece of the extraction of scientific facts. This human input, while costly, is necessary for extracting data that require attention to details (large amounts of metadata) and pointed expertise. As accuracy is a serious concern, humans are also required to validate results from fully-automated models. Computers can greatly improve the discovery and presentation of this data to facilitate the manual extraction when applicable. They can also efficiently capture expert knowledge to alleviate the burden on labeling data for machine learned extraction and validation machine models. Untrained and trained crowds can further decrease expert costs by performing the extraction and providing labels to these models as well. To mitigate losses in accuracy, these crowds can be supervised and assigned concise extraction or labeling sub-tasks.

In terms of distribution of tasks across crowds, one would ideally give the highest numbers of tasks to the cheapest work force: untrained crowds, and progressively funnel the remaining of the tasks towards experts. However, while there exist platforms such as *Amazon Mechanical Turk* and *Figure Eight*[1] for untrained crowds, scaling trained crowds beyond the classroom setting is more challenging. One solution could be community-maintained portals and databases, in which volunteers would occasionally provide data. Another solution

1. Figure Eight is a crowdsourcing platform designed to provide training data (website at `https://www.figure-eight.com/`)

would replicate existing practices of using trained crowds to populate scientific databases during paid internships for example. As the performance of machine learning models improve, the IE systems may move towards bypassing the trained crowds and distributing tasks only between large untrained crowds and a limited number of experts. Such endeavor will require experts to temporarily focus on a new schema design subtask: properly breaking down scientific IE tasks into well-thought out micro-tasks for amateur crowd workers. Once, automated models and untrained crowd performances reach desired accuracy, expert focus can ultimately be narrowed down only to schema design and validation of controversial data.

REFERENCES

[1] Macrostrat. `http://macrostrat.org`. Accessed Sep, 2017.

[2] medkat. `http://ohnlp.sourceforge.net/MedKATp`, 2017. Accessed Sep, 2017.

[3] Paleodb. `http://paleodb.org`, 2019. Accessed Sep, 2019.

[4] Charu C Aggarwal and ChengXiang Zhai. *Mining text data*. Springer Science & Business Media, 2012.

[5] Eugene Agichtein and Luis Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 85–94. ACM, 2000.

[6] Ankit Agrawal and Alok Choudhary. Perspective: Materials informatics and big data: Realization of the "fourth paradigm" of science in materials science. *APL Materials*, 4(5):053208, 2016.

[7] Debra J Audus and Juan J de Pablo. Polymer informatics: Opportunities and challenges. *ACS Macro Letters*, 6(10):1078–1082, 2017.

[8] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

[9] Suneel Bandi and David A Schiraldi. Glass transition behavior of clay aerogel/poly (vinyl alcohol) composites. *Macromolecules*, 39(19):6537–6545, 2006.

[10] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction for the web. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence (IJCAI)*, pages 2670–2676, 2007.

[11] Sugato Basu, Arindam Banerjee, and Raymond J Mooney. Active semi-supervision for pairwise constrained clustering. In *Proceedings of the 2004 SIAM international conference on data mining*, pages 333–344. SIAM, 2004.

[12] Joel R Bell, Kwanho Chang, Carlos R López-Barrón, Christopher W Macosko, and David C Morse. Annealing of cocontinuous polymer blends: Effect of block copolymer molecular weight and architecture. *Macromolecules*, 43(11):5024–5032, 2010.

[13] D. A. Benson, I. Karsch-Mizrachi, D. J Lipman, J. Ostell, B. A Rapp, and D. L. Wheeler. Genbank. *Nucleic Acids Research*, 28(1):15–18, 2000.

[14] Steven Bird and Edward Loper. NLTK: The natural language toolkit. In *42nd Annual Meeting of the. Association for Computational Linguistics*, page 31. Association for Computational Linguistics, 2004.

[15] B Blaiszik, K Chard, J Pruyne, R Ananthakrishnan, S Tuecke, and I Foster. The Materials Data Facility: Data services to advance materials science research. *JOM*, 68(8):2045–2052, 2016.

[16] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

[17] Katarina Boland, Dominique Ritze, Kai Eckert, and Brigitte Mathiak. Identifying references to datasets in publications. In *International Conference on Theory and Practice of Digital Libraries*, pages 150–161. Springer, 2012.

[18] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM, 2008.

[19] J. Brandrup, Edmund H. Immergut, and E. A. Grulke, editors. *Polymer Handbook*. Wiley-Interscience, 4th edition, 1999.

[20] Sergey Brin. Extracting patterns and relations from the world wide web. In *International Workshop on The World Wide Web and Databases*, pages 172–183. Springer, 1998.

[21] Klaus Brinker. Incorporating diversity in active learning with support vector machines. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 59–66, 2003.

[22] M. Buhrmester, T. Kwang, and S. D. Gosling. Amazon's Mechanical Turk: A new source of inexpensive, yet high-quality, data? *Perspectives on Psychological Science*, 6(1):3–5, 2011.

[23] Erik Cambria and Bruce White. Jumping nlp curves: a review of natural language processing research [review article]. *Computational Intelligence Magazine, IEEE*, 9(2):48–57, 2014.

[24] Colin Campbell, Nello Cristianini, Alex Smola, et al. Query learning with large margin classifiers. In *ICML*, volume 20, page 0, 2000.

[25] Gabino A Carriedo, J Luis García Álvarez, FJ García Alonso, A Presa Soto, M Pilar Tarazona, Francisco Mendicuti, and Gema Marcelo. Thermal atropisomerization and photoluminiscence spectra of very high glass transition temperature chiral poly (binaphthoxyphosphazenes) with a secondary helicoidal structure. *Macromolecules*, 37(14):5437–5443, 2004.

[26] Lifeng Chen and Carol Friedman. Extracting phenotypic information from the literature via natural language processing. *Studies in Health Technology and Informatics*, 107(2):758–762, 2004.

[27] Justin Cheng and Michael S Bernstein. Flock: Hybrid crowd-machine learning classifiers. In *Proceedings of the 18th ACM conference on computer supported cooperative work & social computing*, pages 600–611. ACM, 2015.

[28] Laura Chiticariu, Yunyao Li, and Frederick R Reiss. Rule-based information extraction is dead! Long live rule-based information extraction systems! In *Conference on Empirical Methods in Natural Language Processing*, pages 827–832, 2013.

[29] Jinho D Choi, Joel Tetreault, and Amanda Stent. It depends: Dependency parser comparison using a web-based evaluation tool. In *53rd Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 387–396, 2015.

[30] Aaron M Cohen and William R Hersh. A survey of current work in biomedical text mining. *Briefings in Bioinformatics*, 6(1):57–71, 2005.

[31] António Correia, Daniel Schneider, Benjamim Fonseca, and Hugo Paredes. Crowdsourcing and massively collaborative science: a systematic literature review and mapping study. In *International Conference on Collaboration and Technology*, pages 133–154. Springer, 2018.

[32] António Correia, Daniel Schneider, Hugo Paredes, and Benjamim Fonseca. Scicrowd: Towards a hybrid, crowd-computing system for supporting research groups in academic settings. In *International Conference on Collaboration and Technology*, pages 34–41. Springer, 2018.

[33] J. Cowie and W. Lehnert. Information extraction. *Communications of the ACM*, 39(1):80–91, 1996.

[34] Sanjoy Dasgupta and Daniel Hsu. Hierarchical sampling for active learning. In *Proceedings of the 25th international conference on Machine learning*, pages 208–215. ACM, 2008.

[35] J. J. de Pablo, B. Jones, C. L. Kovacs, V. Ozolins, and A. P. Ramirez. The Materials Genome Initiative, the interplay of experiment, theory and computation. *Current Opinion in Solid State and Materials Science*, 18(2):99–117, 2014.

[36] Christopher De Sa, Alex Ratner, Christopher Ré, Jaeho Shin, Feiran Wang, Sen Wu, and Ce Zhang. DeepDive: Declarative knowledge base construction. *ACM SIGMOD Record*, 45(1):60–67, 2016.

[37] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*, 2018.

[38] AT 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*, 25(9):1949–1969, 1987.

[39] A. Doan, R. Ramakrishnan, and A. Y. Halevy. Crowdsourcing systems on the world-wide web. *Communications of the ACM*, 54(4):86–96, 2011.

[40] Zhaoan Dong, Jiaheng Lu, Tok Wang Ling, Ju Fan, and Yueguo Chen. Using hybrid algorithmic-crowdsourcing methods for academic knowledge acquisition. *Cluster Computing*, 20(4):3629–3641, 2017.

[41] H. B. Eitouni and N P. Balsara. Thermodynamics of polymer blends. In *Physical Properties of Polymers Handbook*, pages 339–356. Springer, 2007.

[42] Ashley F Emery and Aleksey V Nenarokomov. Optimal experiment design. *Measurement Science and Technology*, 9(6):864, 1998.

[43] E. Estellés-Arolas and F. González-Ladrón-de Guevara. Towards an integrated crowdsourcing definition. *Journal of Information Science*, 38(2):189–200, 2012.

[44] Valerii Fedorov. Optimal experimental design. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(5):581–589, 2010.

[45] Valerii Vadimovich Fedorov. *Theory of optimal experiments*. Elsevier, 1972.

[46] P. J. Flory. Thermodynamics of high polymer solutions. *The Journal of Chemical Physics*, 10(1):51–61, 1942.

[47] Michael J Franklin, Donald Kossmann, Tim Kraska, Sukriti Ramesh, and Reynold Xin. CrowdDB: Answering queries with crowdsourcing. In *ACM SIGMOD International Conference on Management of Data*, pages 61–72, 2011.

[48] Carol Friedman, Philip O Alderson, John HM Austin, James J Cimino, and Stephen B Johnson. A general natural-language text processor for clinical radiology. *Journal of the American Medical Informatics Association*, 1(2):161–174, 1994.

[49] Carol Friedman, George Hripcsak, Lyuda Shagina, and Hongfang Liu. Representing information in patient reports using natural language processing and the extensible markup language. *Journal of the American Medical Informatics Association*, 6(1):76–87, 1999.

[50] Carol Friedman, Pauline Kra, Hong Yu, Michael Krauthammer, and Andrey Rzhetsky. GENIES: A natural-language processing system for the extraction of molecular pathways from journal articles. In *ISMB (supplement of bioinformatics)*, pages 74–82, 2001.

[51] Shang Gao, Michael T Young, John X Qiu, Hong-Jun Yoon, James B Christian, Paul A Fearn, Georgia D Tourassi, and Arvind Ramanthan. Hierarchical attention networks for information extraction from cancer pathology reports. *Journal of the American Medical Informatics Association*, 25(3):321–330, 2017.

[52] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. Allennlp: A deep semantic natural language processing platform. *arXiv preprint arXiv:1803.07640*, 2018.

[53] Anita Greenhill, Kate Holmes, Chris Lintott, Brooke Simmons, Karen Masters, Joe Cox, and Gary Graham. Playing with science: gamised aspects of gamification found on the Online Citizen Science Project–Zooniverse. pages 15–24. EUROSIS, 2014.

[54] L. Hawizy, D. M Jessop, N. Adams, and P. Murray-Rust. ChemicalTagger: A tool for semantic text-mining in chemistry. *Journal of Cheminformatics*, 3(1):17, 2011.

[55] Kristina M Hettne, Rob H Stierum, Martijn J Schuemie, Peter JM Hendriksen, Bob JA Schijvenaars, Erik M van Mulligen, Jos Kleinjans, and Jan A Kors. A dictionary to identify small molecules and drugs in free text. *Bioinformatics*, 25(22):2983–2991, 2009.

[56] Joanne Hill, Gregory Mulholland, Kristin Persson, Ram Seshadri, Chris Wolverton, and Bryce Meredig. Materials science with large-scale data and informatics: Unlocking new opportunities. *MRS Bulletin*, 41(05):399–409, 2016.

[57] Scott Himmelberger and Alberto Salleo. Engineering semiconducting polymers for efficient charge transport. *MRS Communications*, 5(3):383–395, 2015.

[58] Roger C Hiorns, Ray J Boucher, Rumen Duhlev, Karl-Heinz Hellwich, Philip Hodge, Aubrey D Jenkins, Richard G Jones, Jaroslav Kahovec, Graeme Moad, Christopher K Ober, et al. A brief guide to polymer nomenclature. *Polymer*, 54(1):3–4, 2013.

[59] Eric Horvitz. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 159–166. ACM, 1999.

[60] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[61] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

[62] Keiichi Imato, Atsushi Takahara, and Hideyuki Otsuka. Self-healing of a cross-linked polymer with dynamic covalent linkages at mild temperature and evaluation at macroscopic and molecular levels. *Macromolecules*, 48(16):5632–5639, 2015.

[63] A. Jain, S. Ping Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, and G. Ceder. Commentary: The Materials Project: A materials genome approach to accelerating materials innovation. *APL Materials*, 1(1):011002, 2013.

[64] David M Jessop, Sam E Adams, Egon L Willighagen, Lezan Hawizy, and Peter Murray-Rust. OSCAR4: A flexible architecture for chemical text-mining. *Journal of Cheminformatics*, 3(1):41, 2011.

[65] Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. Distant supervision for relation extraction with sentence-level attention and entity descriptions. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[66] Rob Johnson, Anthony Watkinson, and Michael Mabe. The stm report. *An overview of scientific and scholarly publishing. 5th edition October*, 2018.

[67] Richard G Jones, Edward S Wilks, W. Val Metanomski, Jaroslav Kahovec, Michael Hess, Robert Stepto, and Tatsuki Kitayama, editors. *Compendium of Polymer Terminology and Nomenclature.* The Royal Society of Chemistry, 2009.

[68] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[69] S. Kandel, J. Heer, C. Plaisant, J. Kennedy, F. van Ham, N. Henry Riche, C. Weaver, B. Lee, D. Brodbeck, and P. Buono. Research directions in data wrangling: Visualizations and transformations for usable and credible data. *Information Visualization*, 10(4):271–288, 2011.

[70] Justin G Kennemur, Marc A Hillmyer, and Frank S Bates. Synthesis, thermodynamics, and dynamics of poly (4-tert-butylstyrene-b-methyl methacrylate). *Macromolecules*, 45(17):7228–7236, 2012.

[71] Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. Overview of bionlp'09 shared task on event extraction. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, pages 1–9. Association for Computational Linguistics, 2009.

[72] Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Nigel Collier. Introduction to the bio-entity recognition task at JNLPBA. In *International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, pages 70–75. Association for Computational Linguistics, 2004.

[73] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302, 2015.

[74] Martin Krallinger, Florian Leitner, Obdulia Rabal, Miguel Vazquez, Julen Oyarzabal, and Alfonso Valencia. Overview of the chemical compound and drug name recognition (CHEMDNER) task. In *BioCreative Challenge Evaluation Workshop*, volume 2, page 2, 2013.

[75] Martin Krallinger, Florian Leitner, Obdulia Rabal, Miguel Vazquez, Julen Oyarzabal, and Alfonso Valencia. CHEMDNER: The drugs and chemical names extraction challenge. *Journal of Cheminformatics*, 7(1):S1, 2015.

[76] Michael Krauthammer and Goran Nenadic. Term identification in the biomedical literature. *Journal of Biomedical Informatics*, 37(6):512–526, 2004.

[77] Tu Le, V. Chandana Epa, Frank R. Burden, and David A. Winkler. Quantitative structure-property relationship modeling of diverse materials properties. *Chemical Reviews*, 112(5):2889–2919, may 2012.

[78] Robert Leaman and Graciela Gonzalez. BANNER: An executable survey of advances in biomedical named entity recognition. In *Biocomputing*, pages 652–663. World Scientific, 2008.

[79] Robert Leaman, Chih-Hsuan Wei, and Zhiyong Lu. tmChem: A high performance approach for chemical named entity recognition and normalization. *Journal of cheminformatics*, 7(1):S3, 2015.

[80] David D Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Machine learning proceedings 1994*, pages 148–156. Elsevier, 1994.

[81] Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2124–2133, 2016.

[82] C. J. Lintott, K. Schawinski, A. Slosar, K. Land, S. Bamford, D. Thomas, M. J. Raddick, R. C. Nichol, A. Szalay, D. Andreescu, et al. Galaxy Zoo: morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey. *Monthly Notices of the Royal Astronomical Society*, 389(3):1179–1189, 2008.

[83] Yingdong Luo, Damien Montarnal, Sangwon Kim, Weichao Shi, Katherine P Barteau, Christian W Pester, Phillip D Hustad, Matthew D Christianson, Glenn H Fredrickson, and Edward J Kramer. Poly (dimethylsiloxane-b-methyl methacrylate): A promising candidate for sub-10 nm patterning. *Macromolecules*, 48(11):3422âĂŞ3430, 2015.

[84] Y Lussier and C Friedman. BiomedLEE: A natural-language processor for extracting and representing phenotypes, underlying molecular mechanisms and their relationships. *ISMB: 2007*, 2007.

[85] Jean-François Lutz. Aperiodic copolymers. *ACS Macro Letters*, 3(10):1020–1023, 2014.

[86] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[87] M. D. Mailman, M. Feolo, Y. Jin, M. Kimura, K. Tryka, R. Bagoutdinov, L. Hao, A. Kiang, J. Paschall, L. Phan, et al. The NCBI dbGaP database of genotypes and phenotypes. *Nature Genetics*, 39(10):1181–1186, 2007.

[88] Mónica Marrero, Julián Urbano, Sonia Sánchez-Cuadrado, Jorge Morato, and Juan Miguel Gómez-Berbís. Named entity recognition: fallacies, challenges and opportunities. *Computer Standards & Interfaces*, 35(5):482–489, 2013.

[89] Brigitte Mathiak and Katarina Boland. Challenges in matching dataset citation strings to datasets in social science. *D-Lib Magazine*, 21(1/2):23–28, 2015.

[90] Jason Mattia and Paul Painter. A comparison of hydrogen bonding and order in a polyurethane and poly (urethane- urea) and their blends with poly (ethylene glycol). *Macromolecules*, 40(5):1546–1554, 2007.

[91] Brian E Mattioni and Peter C Jurs. Prediction of glass transition temperatures from monomer and repeat unit structure using computational neural networks. *Journal of Chemical Information and Computer Sciences*, 42(2):232–240, 2002.

[92] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[93] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed

representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[94] I. Muslea. Extraction patterns for information extraction tasks: A survey. In *The AAAI-99 Workshop on Machine Learning for Information Extraction*, pages 1–6, 1999.

[95] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[96] Masatoshi Nagata, Kohichi Takai, Keiji Yasuda, Panikos Heracleous, and Akio Yoneyama. Prediction models for risk of type-2 diabetes using health claims. In *Proceedings of the BioNLP 2018 workshop*, pages 172–176, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[97] Claire Nédellec, Robert Bossy, Jin-Dong Kim, Jung-Jae Kim, Tomoko Ohta, Sampo Pyysalo, and Pierre Zweigenbaum. Overview of bionlp shared task 2013. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 1–7, 2013.

[98] Hieu T Nguyen and Arnold Smeulders. Active learning using pre-clustering. In *Proceedings of the twenty-first international conference on Machine learning*, page 79. ACM, 2004.

[99] Nicola Nosengo. Can artificial intelligence create the next wonder material? *Nature*, 533(7601):22–25, may 2016.

[100] Chandramouli Nyshadham, Corey Oses, Jacob E Hansen, Ichiro Takeuchi, Stefano Curtarolo, and Gus LW Hart. A computational high-throughput search for new ternary superalloys. *Acta Materialia*, 122:438–447, 2017.

[101] Takeshi Onishi, Takuya Kadohira, and Ikumu Watanabe. Relation extraction with weakly supervised learning based on process-structure-property-performance reciprocity. *Science and technology of advanced materials*, 19(1):649–659, 2018.

[102] Shingo Otsuka, Isao Kuwajima, Junko Hosoya, Yibin Xu, and Masayoshi Yamazaki. PoLyInfo: Polymer database for polymeric materials design. In *International Conference on Emerging Intelligent Data and Web Technologies*, pages 22–29. IEEE, 2011.

[103] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[104] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[105] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Conf. of the North American Chapter of the Association for Computational Linguistics*, 2018.

[106] Shanan E Peters, Ce Zhang, Miron Livny, and Christopher Ré. A machine reading system for assembling synthetic paleontological databases. *PLoS One*, 9(12):e113523, 2014.

[107] M. J. Raddick, G. Bracey, P. L. Gay, C. J. Lintott, C. Cardamone, P. Murray, K. Schawinski, A. S. Szalay, and J. Vandenberg. Galaxy Zoo: Motivations of citizen scientists. *arXiv preprint arXiv:1303.6886*, 2013.

[108] Alec Radford, Karthik Narasimhan, Time Salimans, and Ilya Sutskever. Improving language understanding with unsupervised learning. Technical report, Technical report, OpenAI, 2018.

[109] Juan Ramos. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, 2003.

[110] Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. Data programming: Creating large training sets, quickly. In *Advances in Neural Information Processing Systems*, pages 3567–3575, 2016.

[111] Radim Rehurek and Petr Sojka. Software framework for topic modelling with large corpora. In *Workshop on New Challenges for NLP Frameworks*. Citeseer, 2010.

[112] Fabio Rinaldi, Gerold Schneider, Kaarel Kaljurand, Michael Hess, Christos Andronis, Ourania Konstandi, and Andreas Persidis. Mining of relations between proteins over biomedical scientific literature using a deep-linguistic approach. *Artificial Intelligence in Medicine*, 39(2):127–136, 2007.

[113] Tim Rocktäschel, Michael Weidlich, and Ulf Leser. ChemSpot: A hybrid system for chemical named entity recognition. *Bioinformatics*, 28(12):1633–1640, 2012.

[114] David E Rumelhart. Learning internal representations by back-propagating errors. *Parallel distributed processing: Explorations in the microstructure of cognition*, 1:318–362, 1986.

[115] Andrey Rzhetsky, Ivan Iossifov, Tomohiro Koike, Michael Krauthammer, Pauline Kra, Mitzi Morris, Hong Yu, Pablo Ariel Duboué, Wubin Weng, W John Wilbur, et al. GeneWays: A system for extracting, analyzing, visualizing, and integrating molecular pathway data. *Journal of Biomedical Informatics*, 37(1):43–53, 2004.

[116] Philip N Sabes and Michael I Jordan. Reinforcement learning by probability matching. In *Advances in Neural Information Processing Systems*, pages 1080–1086, 1995.

[117] Erik F Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *7th Conference on Natural Language Learning*, pages 142–147, 2003.

[118] Guergana K Savova, James J Masanz, Philip V Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C Kipper-Schuler, and Christopher G Chute. Mayo clinical text analysis and knowledge extraction system (cTAKES): Architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513, 2010.

[119] Ariel S Schwartz and Marti A Hearst. A simple algorithm for identifying abbreviation definitions in biomedical text. In *Pac. Symp. Bio.*, pages 451–462. 2002.

[120] C. Seifert, M. Granitzer, P. Höfler, B. Mutlu, V. Sabol, K. Schlegel, S. Bayerl, F. Stegmaier, S. Zwicklbauer, and R. Kern. Crowdsourcing fact extraction from scientific literature. In *Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data*, pages 160–172. Springer, 2013.

[121] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

[122] H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294. ACM, 1992.

[123] Arfon M Smith, Stuart Lynn, and Chris J Lintott. An introduction to the Zooniverse. In *First AAAI Conference on Human Computation and Crowdsourcing*, 2013.

[124] Matthew C Swain and Jacqueline M Cole. ChemDataExtractor: A toolkit for automated extraction of chemical information from the scientific literature. *Journal of Chemical Information and Modeling*, 56(10):1894–1904, 2016.

[125] Jaana Takis, AQM Islam, Christoph Lange, and Sören Auer. Crowdsourced semantic annotation of scientific publications and tabular data in PDF. In *11th International Conference on Semantic Systems*, pages 1–8. ACM, 2015.

[126] Javier Tamames and Alfonso Valencia. The success (or not) of HUGO nomenclature. *Genome biology*, 7(5):402, 2006.

[127] Roselyne B Tchoua, Kyle Chard, Debra J Audus, Logan T Ward, Joshua Lequieu, Juan J De Pablo, and Ian T Foster. Towards a hybrid human-computer scientific information extraction pipeline. In *13th International Conference on e-Science*, pages 109–118. IEEE, 2017.

[128] Roselyne B Tchoua, Zhi Hong, Ashuti Ajith, Kyle Chard, Debra J Audus, Logan T Ward, Shrayesh Patel, Juan J De Pablo, and Ian T Foster. Active learning yields better training data for scientific named entity recognition. To appear at the 2019 IEEE 15th International Conference on e-Science (e-Science).

[129] Roselyne B Tchoua, Zhi Hong, Kyle Chard, Logan Ward, Alexander Belikov, , Debra Audus, Shrayesh Patel, Juan de Pablo, and Ian Foster. Creating training data for scientific named entity recognition with minimal human effort. In *International Conference on Computational Science*, 2019.

[130] Roselyne B. Tchoua, Jian Qin, Debra J. Audus, Kyle Chard, Ian T. Foster, and Juan de Pablo. Blending education and polymer science: Semiautomated creation of a thermodynamic property database. *Journal of Chemical Education*, 93(9):1561–1568, 2016.

[131] K. M. Tolle, D. S. W. Tansley, and A. J. G. Hey. The fourth paradigm: Data-intensive scientific discovery. *Proceedings of the IEEE*, 99(8):1334–1337, Aug 2011.

[132] Dirk Willem Van Krevelen and Klaas Te Nijenhuis. *Properties of polymers: their correlation with chemical structure; their numerical estimation and prediction from additive group contributions*. Elsevier, 2009.

[133] Paroma Varma and Christopher Ré. Snuba: automating weak supervision to label training data. *Proceedings of the VLDB Endowment*, 12(3):223–236, 2018.

[134] Max Völkel, Markus Krötzsch, Denny Vrandecic, Heiko Haller, and Rudi Studer. Semantic wikipedia. In *Proceedings of the 15th international conference on World Wide Web*, pages 585–594. ACM, 2006.

[135] Luis Von Ahn. Games with a purpose. *Computer*, 39(6):92–94, 2006.

[136] Byron C Wallace, Anna Noel-Storr, Iain J Marshall, Aaron M Cohen, Neil R Smalheiser, and James Thomas. Identifying reports of randomized controlled trials (RCTs) via a hybrid machine learning and crowdsourcing approach. *Journal of the American Medical Informatics Association*, 2017.

[137] Yiming Yang and Xin Liu. A re-examination of text categorization methods. In *22nd Annual International ACM SIGIR Conference*, pages 42–49. ACM, 1999.

[138] Donald G York, J Adelman, John E Anderson Jr, Scott F Anderson, James Annis, Neta A Bahcall, JA Bakken, Robert Barkhouser, Steven Bastian, Eileen Berman, et al. The Sloan Digital Sky Survey: Technical summary. *The Astronomical Journal*, 120(3):1579, 2000.

[139] Xinliang Yu. Support vector machine-based QSPR for the prediction of glass transition temperatures of polymers. *Fibers and Polymers*, 11(5):757–766, 2010.

[140] Zhiqiang Zeng, Hua Shi, Yun Wu, and Zhiling Hong. Survey of natural language processing techniques in bioinformatics. *Computational and mathematical methods in medicine*, 2015, 2015.

[141] Ce Zhang, Vidhya Govindaraju, Jackson Borchardt, Tim Foltz, Christopher Ré, and Shanan Peters. GeoDeepDive: Statistical inference using familiar data-processing languages. In *ACM SIGMOD International Conference on Management of Data*, pages 993–996, 2013.

[142] Deyu Zhou and Yulan He. Extracting interactions between proteins from the literature. *Journal of Biomedical Informatics*, 41(2):393–407, 2008.

[143] Kaiyin Zhou, Sheng Zhang, Xiangyu Meng, Qi Luo, Yuxing Wang, Ke Ding, Yukun Feng, Mo Chen, Kevin Cohen, and Jingbo Xia. Crf-lstm text mining method unveiling the pharmacological mechanism of off-target side effect of anti-multiple myeloma drugs. In *Proceedings of the BioNLP 2018 workshop*, pages 166–171, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[144] Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, 2017.

[145] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005.