Ensemble Labeling Towards Scientific Information Extraction (ELSIE)

Erin Murphy[0000−0002−3767−3925], Alexander Rasin, Jacob Furst, Daniela Raicu, and Roselyne Tchoua

DePaul University, Chicago IL 60604, USA
emurph35@depaul.edu

Abstract. Extracting scientific facts from unstructured text is difficult due to challenges specific to the complexity of the scientific named entities and relations to be extracted. This problem is well illustrated through the extraction of polymer names and their properties. Even in the cases where the property is a temperature, identifying the polymer name associated with the temperature may require expertise due to the use of complicated naming conventions and by the fact that new polymer names are being "introduced" into the lexicon as polymer science advances. While domain-specific machine learning toolkits exist that address these challenges, perhaps the greatest challenge is the lack of—time-consuming, error-prone and costly—labeled data to train these machine learning models. This work repurposes Snorkel, a data programming tool, in a novel approach as a way to identify sentences that contain the relation of interest in order to generate training data, and as a first step towards extracting the entities themselves. By achieving 94% recall and an F1 score of 0.92, compared to human experts who achieve 77% recall and an F1 score of 0.87, we show that our system captures sentences missed by both a state-of-the-art domain-aware natural language processing toolkit and human expert labelers. We also demonstrate the importance of identifying the complex sentences prior to extraction by comparing our application to the natural language processing toolkit.

Keywords: Information Extraction · Data Labeling · Relations Extraction · Snorkel · Data Programming · Polymers.

1 Introduction

Extracting scientific facts from esoteric articles remains an important natural language processing (NLP) research topic due to the particularity of the entities and relations to be extracted. The challenges involved include the fact that entities can be described by multiple referents (synonymy), one word can refer to different concepts depending on context (polysemy), and other domain-specific nuances. These issues arise in many fields as evidenced by NLP tools that rely on domain-specific grammar and ontologies.

Perhaps the most significant challenge in scientific Information Extraction (IE) is the lack of readily available labeled training data. The process of creating

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_60

well-balanced, manually-labeled datasets of scientific facts is difficult due in part to the aforementioned challenges, but also due to the scarcity of entities and relations in scientific articles. For instance, it is not uncommon for scientists to write an article about a single newly synthesized material. To annotate sentences in such publications not only requires time and attention, but is also costly as it requires time from domain experts and cannot be easily crowdsourced.

Our ultimate goal is to alleviate the burden of expert annotators and facilitate extraction of scientific facts. Towards achieving this goal, we repurpose a data programming software [14] to identify sentences that contain scientific entities and relations automatically. Typically, data programming relies on existing entity *taggers* in order to identify and label relations. The key novelty of our approach lies in the identification of sentences containing the target entities and relations without identifying the entities through the use of dictionaries nor through complicated hard-coded rules. Instead, we use data programming to describe and combine approximate descriptions of the relations and the entities involved. Not only are we able to identify sentences of interest accurately (94% recall), but our combination of weak, programmed rules retrieves sentences that were missed by human experts and state-of-the-art domain-specific software.

The rest of this paper is organized as follows. In Section 2, we briefly discuss the application motivation for this work. Section 3 discusses related work. Section 4 presents the architecture of our system. Section 5 presents the results or our approach, followed by a conclusion in Section 6.

2 Motivation

The initial motivation for this work is polymer science. Polymers are large molecules composed of many repeating units, referred to as monomers. Partly due to their large molecular masses, polymers have a variety of useful properties (elasticity, resistance to corrosion and more). Given such properties, polymers are ubiquitous and gathering information about their properties is an essential part of materials design [1]. One specific property with a profound impact on their application, and what this work specifically targets, is the glass transition temperature (T_g): the temperature at which a polymer transitions from a solid, amorphous, glassy state to a rubbery state as the temperature is increased. As the properties between the two states are drastically different, it is crucial to identify polymers with the appropriate T_g for different applications. For example, plexiglass (poly(methyl methacrylate)), 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 [2].

3 Related Work

The medical community has long been invested in applying information extraction methods to medical publications [5, 6, 9, 16]. These tools are designed to extract clinical information from text documents and translate entities and terms

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_60

to controlled ontologies and vocabularies. Other communities have followed, such as biology, where MedLEE, a tool used to extract clinical information from medical documents [5,6], led to the development of GENIES [7] and BioMedLEE [3] which extract biomolecular substances and phenotypic data from text. However, developing NLP tools for such specialized ontologies can be error-prone, time consuming and hard to maintain, and requires a knowledge of the domain.

Scientific IE models remedy the above challenges by learning from data. Statistical models include Conditional Random Field (CRF), which are graph-based models used in NLP to capture context by learning from sequences of words; long short-term memory (LSTM) networks, which are recurrent neural networks that also capture context by learning relationships between a word and its preceding word; and bidirectional LSTM (Bi-LSTM) networks, which exploit information about the words that come before and after a given word. These models have shown great promise when applied to scientific IE [3, 8, 10, 15, 17]. Another example, ChemDataExtractor (CDE)—to which we compare our work and refer to as the state-of-the-art tool—implements an extensible end-to-end text-mining pipeline that can process common publication formats and produces machine-readable structured output data (chemicals and their properties) [17]. While machine learning techniques do not require the implementation of rich domain ontologies and grammars, they do rely heavily on labeled training data to achieve high accuracy, especially when focusing on specialized entities/relations.

While tagging entities and identifying relations between them may be crowdsourced to the general public for general IE, labeling esoteric scientific articles requires domain knowledge [18–20] and can be costly. Distant supervision [11] circumvents the need for expensive annotation by leveraging available databases or semi-structured text. Deep learning tools like PaleoDeepDive[1] use advanced statistical inference approaches to extract paleontological data from text, tables, and figures in scientific texts by mapping entities and their relations from a large database to text [4, 12]. Unfortunately, many fields do not have access to large databases of entities and relations, especially if new data is constantly added.

Snorkel, for example, uses weak programmed rules called labeling functions (LFs) to describe relations between known entities; it can learn and model accuracies and conflicts between LFs to approximately and quickly create labels on unlabeled data [13, 14]. However, as mentioned, scientific entities and relations are complex and difficult to extract automatically; while many relations extraction work focuses on relations between two entities, scientific relations may consist of multiple entities and relations or include additional metadata [18, 20].

Our work uses Snorkel in a novel manner to address these crucial scientific IE challenges: 1) many NLP tools assume access to costly carefully labeled, balanced datasets, while scientific entities can be scarce in publications; 2) our entities are not always known a priori and are continuously being created or discovered; 3) relations identification is not dependent on first identifying the entities; and 4) our relations are complex and may contain entities with multiple relations, hence requiring further expert scrutiny to be extracted.

[1] PaleoDB at http://paleodb.org

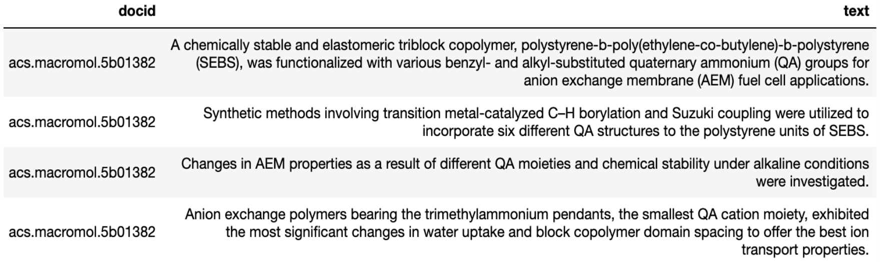
ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_60

4 Architecture

Databases that contain information about polymers and their properties are not readily and freely available, thereby creating a need to be able to extract polymers and their properties without relying on external databases to supply known information. A tool is needed to not only extract polymer names from text without knowing them a priori, but can also extract information about the polymer's properties. The particular property this work targets is a polymer's T_g. We have therefore built a tool which aims to identify the three entities of a polymer's T_g, or a polymer-T_g pair: 1) polymers and/or their abbreviations, 2) temperatures and 3) glass transition-mentions.

4.1 Input Dataset

The input dataset was made up of 9,518 unique text sentences (data points) from 31 journal articles containing "Tg" from a keyword search from the journal, *Macromolecules*, a prominent journal in polymer science, during the years 2006-2016 [19]. The full text version of each article was downloaded in HTML format, and split into sentences (Fig. 1) so that each data point was tied to a document (journal article) identifier [19]. The sentences were not preprocessed nor altered in any way prior to this work.



docid	text
acs.macromol.5b01382	A chemically stable and elastomeric triblock copolymer, polystyrene-b-poly(ethylene-co-butylene)-b-polystyrene (SEBS), was functionalized with various benzyl- and alkyl-substituted quaternary ammonium (QA) groups for anion exchange membrane (AEM) fuel cell applications.
acs.macromol.5b01382	Synthetic methods involving transition metal-catalyzed C–H borylation and Suzuki coupling were utilized to incorporate six different QA structures to the polystyrene units of SEBS.
acs.macromol.5b01382	Changes in AEM properties as a result of different QA moieties and chemical stability under alkaline conditions were investigated.
acs.macromol.5b01382	Anion exchange polymers bearing the trimethylammonium pendants, the smallest QA cation moiety, exhibited the most significant changes in water uptake and block copolymer domain spacing to offer the best ion transport properties.

Fig. 1. Example of Input database [2]

4.2 The Snorkel System and Its Built-In Functionalities

Snorkel is a system developed at Stanford University with the objective to "...programmatically [build] and [manage] training datasets without manual labeling" [13]. It applies user-defined programmed rules as weak learners to label data points in a dataset and avoids having to manually assign each data point. The weak learners, or rules programmed in a computer language such as Python, in the Snorkel system are known as labeling functions (LFs). Multiple LFs can be created, and their logic can often be in opposition to each other. After applying LFs to the input data, Snorkel can determine if a data point should be

[2] Extracted from: Mohanty, Angela D., Chang Y. Ryu, Yu Seung Kim, and Chulsung Bae. "Stable Elastomeric Anion Exchange Membranes Based on Quaternary Ammonium-Tethered Polystyrene-B-Poly (Ethylene-Co-Butylene)-B-Polystyrene Triblock Copolymers." Macromolecules 48, no. 19 (2015): 7085-95.

labeled or not. Part of the motivation to use Snorkel was to leverage its speed and ease-of-use of the LFs rather than rely on hard-to-maintain hard-coded rules.

Snorkel Preprocessors and the Uniqueness of Polymer Data The Snorkel preprocessor [13] allows for each data point to be preprocessed in a user-defined manner. This is important because polymer names do not always follow the same textual rules. For example, it is common for polymer names to be represented throughout polymer texts by abbreviations, consisting largely of uppercase alpha character strings. Applying a preprocessing function to make all text lowercase before applying the LFs would result in missing many abbreviations. On the other hand, there are times when the same sentence containing an abbreviation needs to be made lowercase in order to look for a different entity (i.e. a glass transition-mention). Consider the following sentence:

> Bacterial polyhydroxy alkanoates such as poly(3-hydroxybutyrate) (P3HB), poly(3-hydroxyvalerate) (P3HV), or higher hydroxy acids and their copolymers display decreasing melting points from about 180 °C (Tg = 1–4 °C) for P3HB to 112 °C (Tg = –12 °C) for P3HV.[3]

To find a glass transition-mention, a conversion to lowercase and a search for “tg” (a transformation of “TG” or “Tg” or “tg”) can be performed. If this conversion were permanent, then polymer abbreviations like P3HB and P3HV in the above sentence would never be identified. Finding different entities may require numerous, impermanent preprocessing applications on the same data point, which are easily accommodated by Snorkel preprocessor functions.

Three preprocessors are built for this work: *makeTextLower()*, *makeCharUniform()* and *removeSpacesInParentheses()*. *makeTextLower()* is self-explanatory: it converts input sentences into lowercase text. *makeCharUniform()* converts special characters, such as dashes and apostrophes (which can appear throughout polymer texts as different characters) to a uniform character. For example, a dash can be represented by the following characters: - – — ―. Uniformizing these characters is important, especially if they are used in LF logic. Polymer names can often contain multiple character tokens within parentheses. Consider the polymer name: poly(tetrafluoroethylene). Although this is the common spelling for this polymer, it is possible the polymer could be referred to as: poly(tetrafluoro ethylene). If so, it would be important that a computer program knows that both poly(tetrafluoroethylene) and poly(tetrafluoro ethylene) are the same polymer. Therefore, *removeSpacesInParentheses()* was built to remove spaces only within parentheses.

Labeling Functions (TRUE, JUNK, ABSTAIN) When Snorkel LFs are applied to data points, each LF returns a value of 1, 0 and -1 indicating a TRUE, FALSE (JUNK) or ABSTAIN label, respectively. Attention must be paid to what value is returned since it can greatly impact labeling a data point as TRUE or FALSE. For example, if three LFs are applied to a sentence and their output

[3] Extracted from: Petrovic, Zoran S, Jelena Milic, Yijin Xu, and Ivana Cvetkovic. "A Chemical Route to High Molecular Weight Vegetable Oil-Based Polyhydroxyalkanoate." Macromolecules 43, no. 9 (2010): 4120-25.

renders [1, 0, 1], 2 of 3 LFs deemed the sentence to be TRUE (1), if using a majority voting system. If the LF outputs are [1, 0, 0], the sentence would be deemed FALSE (0) since 2 of 3 LFs returned 0. If the LF outputs are [1, 0, -1], this is equivalent to saying that only two LFs voted (1 and 0) and one abstained (-1), resulting in a 50% chance the data point is TRUE or FALSE, and a majority does not exist.

Labeling Functions to Identify Different Entities For a sentence to be labeled TRUE, it must contain three different entity types: a polymer name or abbreviation, a temperature and a glass transition-mention. Snorkel combines the output of all LFs to label an entire datapoint with a value of 1, 0 or -1. Our work modifies this functionality by first having each LF look for (or lack thereof) one of the three entities within a sentence, where multiple LFs, though expressing different logic, can look for the same type of entity. As a result, each LF is designed to either identify a polymer name or abbreviation entity, a temperature entity or a glass transition-mention entity, thereby grouping LFs by the type of entity for which they are looking. After the LF group determines if the respective entity is present, an ensemble labeler applies a label of 1 (TRUE) or 0 (FALSE) the sentence; if all three entities are present in a sentence, it receives a 1, else it receives a 0. This ensemble labeler will be discussed in section 4.3.

The following code examples illustrate the logic for two different LFs that target temperature entities: $tempUnits()$ and $JUNKnoNumbers()$. The rationale for their logic will be discussed in the following sections, but for now we shall illustrate the architecture of LFs.

```
@labeling_function()
def tempUnits(x):
  return TG if "°" in x.text else JUNK
```

Listing 1.1. tempUnits() Labeling Function

```
@labeling_function(pre=[makeTextLower])
def JUNKnoNumbers(x):
      regexp = re.compile(r"[0-9]")
      return ABSTAIN if regexp.search(x.text) else JUNK
```

Listing 1.2. JUNKnoNumbers() Labeling Function

In the above examples, @$labeling\_function()$ signals to Snorkel that a LF is to be defined [13], x refers to the input datapoint which consists of a document ID (*docid*) and a sentence (*text*) (see Fig. 1), while the variables TG, $JUNK$ and $ABSTAIN$ are assigned values of 1, 0 and -1, respectively. In Listing 1.1, a 1 is returned if a degree sign (°) is found within *text* indicating the LF found a temperature entity within *x.text*, otherwise a 0 is returned indicating that a temperature entity was not found. In Listing 1.2, the regular expressions (re) module [21] allows for a regular expression search to be performed on *x.text* in that if any numeric digits exist, a -1 is returned, otherwise a 0 is returned. It should also be noted that the preprocessor, $makeTextLower()$, is also applied to *x.text* prior to applying $JUNKnoNumbers()$.

The following sections describe the three groupings of LFs, lists the individual LFs for the group, and describes their logic.

Labeling Polymer Entities Only four LFs are required to identify sentences with polymer entities without a priori knowledge, external reference dictionaries, or writing rules which use extensive REGEX functions.

1. **abbreviation_in_sentence**: This LF looks for a token that consists only of uppercase alpha characters, numbers and special characters. Only 40% or less of the token can consist of special and numeric characters. For example, P3HB is considered an abbreviation, whereas 270°C is not since 100% of characters in the latter token are numbers and special characters. If the criteria is met, the LF returns 1, otherwise it returns a -1. It would not be appropriate to return a 0 if the logic is not met because polymers do not always have abbreviations, and a sentence should not be penalized for not containing an abbreviation.
2. **keyword_poly**: This LF looks for the character string, "poly" in a sentence. If it exists, a 1 is returned, otherwise a -1 is returned.
3. **keyword_polyParen**: Similar to keyword_poly(), if a sentence contains, "poly(", then a 1 is returned, otherwise a -1 is returned.
4. **keyword_copolymer**: There are naming conventions applied to certain types of polymers known as copolymers. This LF accounts for those rules in that if any of these character strings are found in a sentence, a 1 is returned, else a -1 is returned. Examples of character strings found in copolymers are: "-co-", "-stat-", "-per-", "-ran-", "-grafted-", "-trans-", and "-alt-".

Labeling Temperature Entities It is simple to identify numbers in a sentence, but it is more difficult to discern what those numbers represent. The below lists the LFs used to identify sentences with and without temperature entities.

5. **tempUnits**: This LF simply looks for a degree (°) symbol. If found, it returns 1, otherwise it returns -1.
6. **tempUnitsAfterNumber**: If a number is followed by a unit of temperature such as C (Celsius), F (Farenheit) or K (Kelvin), then a 1 is returned, otherwise a -1 is returned.
7. **tempUnitsAfterDegree**: If a degree (°) symbol is followed by a C, F, or K, then a 1 is returned, otherwise a -1 is returned.
8. **equalSignBeforeNumber**: If an equal ($=$) sign exists before numbers (with or without special characters like $-$ or $\sim$), then a 1 is returned, otherwise a -1 is returned.
9. **circaSignBeforeNumberDegree**: If the tokens "circa" or "ca" or "about" precede a number (with or without special characters like $-$ or $\sim$), then a 1 is returned, otherwise a -1 is returned.
10. **tempRange**: Glass transition temperatures can be reported as a temperature range. This LF returns a 1 if more than 40% of a token's characters consists of numbers, such as in the case of "-2-1" which could read, "negative 2 to negative 1." Otherwise a -1 is returned.
11. **JUNKtempUnitsAfterNumber**: If a number exists and is not followed by a degree (°) symbol, C, F, or K, the number is assumed to not be a temperature and a 0 is returned, otherwise a -1 is returned.
12. **JUNKtempUnitsAfterDegree**: If a degree (°) symbol exists and is not followed by a C, F, or K, it is assumed the sentence does not contain a temperature and a 0 is returned, otherwise a -1 is returned.
13. **JUNKnoNumbers**: If there are no numbers in a sentence, a 0 is returned, otherwise a -1 is returned. A 1 is not returned because that assumes a temperature exists. Since not all numbers represent temperatures, it can only be assumed that a sentence containing numbers is at, best, not a JUNK sentence.

Labeling Glass Transition-Mentions There are a discrete number of ways that a glass transition mention can be expressed through text, which is either by spelling out "glass transition" (with varying forms of capitalization), shortening it to "glass trans" or "glass-trans," or abbreviating it to simply "tg." Ultimately, this search can be streamlined to searching for: "glass t" or "glass-t" or "tg."

In polymer texts there is a technique called thermogravimetric analysis, which is sometimes abbreviated as, "TGA." Therefore, additional LFs are needed to to distinguish sentences that contain "TGA" vs just "TG" to avoid labeling sentences that only refer to TGA as containing a glass transition-mention entity.

14. **keyword_tg**: If the (lowercase) character strings “glass t” or “glass-t” or “tg” are found in a sentence a 1 is returned, otherwise a −1 is returned.
15. **JUNK_tga**: If the character string “TGA” is found in a sentence a 0 is returned, otherwise a −1 is returned.
16. **JUNK_tgAndTGA**: This is considered a “tie-breaker” LF for sentences containing “TGA.” If this LF did not exist, sentences with “TGA” would return LF output arrays as [1, 0] and would need to be resolved with a tie-breaker (i.e. randomly assigning the glass transition mention entity as 1 or 0). Therefore, if “TG” is found in a sentence with no other alpha characters following it, a 1 is returned; if the character string “TGA” is found, then a 0 is returned; otherwise a −1 is returned.

4.3 Majority Ensemble Labeler and ELSIE

There are a total of 16 LFs used in this work. The first four LFs, highlighted below in yellow, aim to identify polymer names and abbreviations, the next nine LFs, highlighted in green, aim to identify temperatures, and the last three, in blue, aim to identify glass transition-mentions. Applying LFs is demonstrated in the below sentences and respective output arrays, where characters and/or words are color-coded to indicate the entity identified by a particular LF group, and the LF output values are listed in the corresponding order in the output array as outlined below. The values of the output arrays correspond to the LFs as enumerated in Sec. 4.2—Labeling Polymer Entities, Labeling Temperature Entities and Labeling Glass Transition-Mentions, such that the output array values are designated by the following LFs: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16].

Sentence 1 Bacterial polyhydroxy alkanoates such as poly(3-hydroxybutyrate) (P3HB), poly(3-hydroxyvalerate) (P3HV), or higher hydroxy acids and their copolymers display decreasing melting points from about 180 °C (Tg = 1–4 °C) for P3HB to 112 °C (Tg = -12 °C) for P3HV.[4]

Output array: [1, 1, 1, -1, 1, 1, 1, 1, -1, 1, -1, -1, -1, 1, -1, 1] → [3/3, 5/5, 2/2]

Of the LFs that did not abstain in Sentence 1 (where the output was either a 1 or 0, but not a -1), all three entities of interest were identified; 3 of 3 LFs found a polymer name or abbreviation (yellow), 5 of 5 LFs found a temperature (green), and 2 of 2 LFs found a glass transition-mention (blue).

Sentence 2 Although the corresponding copolymers were afforded with perfectly alternating nature and excellent regiochemistry control, only glass-transition temperatures of around 8.5 °C were observed in the differential scanning calorimetry (DSC) curve, demonstrating that the polymers are completely amorphous (see Supporting Information Figure S3).[5]

Output array: [1, 1, -1, -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, 1, -1, -1] → [2/2, 3/3, 1/1]

Similar to sentence 1, LFs applied to Sentence 2 also identified all three entities of interest, however, more LFs abstained in this sentence than in Sentence 1.

Sentence 3 (repeated to show how Snorkel can correctly label tricky sentences):

- The TGA scans indicated that APNSi has 5% decomposition in air of 340 °C and in argon of 450 °C (Figure 1).
- The TGA scans indicated that APNSi has 5% decomposition in air of 340 °C and in argon of 450 °C (Figure 1).

[4] Extracted from: Petrovic, Zoran S, Jelena Milic, Yijin Xu, and Ivana Cvetkovic. "A Chemical Route to High Molecular Weight Vegetable Oil-Based Polyhydroxyalkanoate." Macromolecules 43, no. 9 (2010): 4120-25.

[5] Extracted from: Yue, Tian-Jun, Wei-Min Ren, Ye Liu, Zhao-Qian Wan, and Xiao-Bing Lu. "Crystalline Polythiocarbonate from Stereoregular Copolymerization of Carbonyl Sulfide and Epichlorohydrin." Macromolecules 49, no. 8 (2016): 2971-76.

- The TGA scans indicated that APNSi has 5% decomposition in air of 340 °C and in argon of 450 °C (Figure 1).[6]

Output array: [1, -1, -1, -1, 1, -1, 1, -1, 1, -1, 0, -1, -1, 1, 0, 0] → [1/1, 3/4, 1/3]

An abbreviation of "TGA" was identified in sentence 3, and polymer LFs cannot determine if the abbreviation represents a polymer or not. The LFs also found the string, "TG" in the sentence. The LFs are able to determine that a glass transition-mention is not present (refer to 4.2 Labeling Glass Transition-Mentions for further clarification), and that only abbreviation and temperature entities were found. As a result, the sentence was labeled as 0. This final example demonstrates the power of LFs and how the combination of weak learners allow the system to carve out entities of interest while ignoring entities not of interest from the sentence by picking up on nuances to discern which sentences to label, even tricky ones.

Once the output arrays of the LFs are generated, the majority ensemble labeler determines which entities exist in a sentences by using a simple majority of LF outputs per entity group; polymer (yellow), temperature (green) and glass transition-mention (blue). The majority ensemble labeler will label a sentence as 1 if and only if all three entities are present in a sentence. This process of considering the output of all LFs per entity group and determining if all entities are present is being called, ensemble labeling toward scientific information extraction, or ELSIE.

5 Results and Analysis

First, we discuss how the initial gold standard dataset—determined by human experts and the state-of-the-art tool—was generated in order to be compared to sentences labeled by ELSIE. Next, we discuss how the initial gold standard dataset was updated after ELSIE identified true positive sentences that were missed by human experts and the state-of-the-art tool. Finally, the state-of-the-art tool's and ELSIE's outputs are both compared to the updated gold standard (hereafter referred to as the "gold standard") dataset. The state-of-the-art tool's performance against the gold standard is discussed as a matter of comparison to ELSIE's performance and labeling abilities.

5.1 Training Dataset and its Labels

The intention of the initial gold standard dataset was to label extracted polymer entities and their T_g; this differs from the current intention of ELSIE which aims to label sentences containing the three entities of interest. The motivation behind our approach of labeling sentences before extracting entities is that scientific entities and relations can be too complex to be immediately extracted and may require additional human attention or additional passthroughs of the data. To

[6] Extracted from: Finkelshtein, E Sh, KL Makovetskii, ML Gringolts, Yu V Rogan, TG Golenko, LE Starannikova, Yu P Yampolskii, VP Shantarovich, and T Suzuki. "Addition-Type Polynorbornenes with Si (Ch3) 3 Side Groups: Synthesis, Gas Permeability, and Free Volume." Macromolecules 39, no. 20 (2006): 7022-29.

align the initial gold standard dataset with ELSIE-labeled data, metadata about sentences and polymer-T_g pairs extracted by experts and the state-of-the-art tool was used to determine the sentences from which the entities were extracted. Data extracted by the state-of-the-art tool was previously validated by experts [19]. If the state-of-the-art tool extracted a polymer-T_g pair correctly, the sentence(s) from which the information was obtained by the state-of-the-art tool were labeled as 1; if the state-of-the-art tool extracted an incorrect polymer-T_g pair (i.e. a polymer was paired with an incorrect T_g), sentences containing the correct polymer name/abbreviation and the T_g were both labeled as 0 [19]. Sentences identified by the human experts which contained polymer-T_g pairs were labeled as 1. If a polymer-T_g pair existed in the corpora, and the human experts and/or the state-of-the-art tool did not extract the pair, the sentence was labeled as 0.

5.2 Updated Gold Standard Labels

After running ELSIE on unlabeled data, new polymer-T_g pairs that were not in the initial gold standard dataset (i.e. missed by human experts and/or the state-of-the-art tool) were discovered. We considered these to be false "false positives" from the initial gold standard dataset. More details of these sentences are provided in section 5, but as a result of these findings, the initial gold standard dataset was updated, and the sentences with previously missed polymer-T_g were labeled as 1. It is this updated dataset—the gold standard—to which the state-of-the-art tool and ELSIE are compared.

5.3 Results

The final document corpora contained 9,518 sentences (data points), representing 31 unique scientific journal articles. Overall, the state-of-the-art tool labeled 15 sentences as positive cases, ELSIE labeled 67 sentences, and human experts identified 49; the gold standard dataset contained 64 positive cases. Positive cases represent less than 1% of the data, illustrating the highly unbalanced nature of the dataset, and accuracy alone does not convey each application's performance. Precision and recall results, along with accuracy and F1-scores, are reported in Table 1.

Table 1. Performance Compared to the Gold Standard.

	Gold Standard	**Human Experts**	**State-of-the-Art Tool**	**ELSIE**
Total Cases	9,518			
Total Positive Cases	64	49	15	67
Accuracy		99.84%	99.49%	99.88%
Precision		100%	100%	90%
Recall		77%	23%	94%
F1 score		0.87	0.38	0.92

The analyses were run on a personal laptop using Python 3.8 in Jupyter Notebook. The total processing time to process all 9,518 sentences through ELSIE, including Snorkel preprocessors, was 0:01:03, compared to the state-of-the-art tool's processing time which took approximately 0:26:00 to process 31 documents.

5.4 Analysis

The F1 score of the state-of-the-art's performance compared to the gold standard (0.38) versus the F1 score of ELSIE's performance to the gold standard (0.92) overall demonstrates that ELSIE is better at labeling sentences correctly.

It is more important to capture all true labels than it is to miss true labels, and is therefore acceptable for precision to be compromised in order to obtain high recall. ELSIE identified new polymer-T_g pairs that human experts and the state-of-the-art tool missed (see Table 2). With recall for the human experts (77%) being lower than ELSIE (94%), and the need to update the gold standard dataset to include new polymer-T_g pairs that were previously missed, this demonstrates that a high level of attention is required by humans (even experts) when reading texts, otherwise important information can get missed. This finding also highlights ELSIE's robustness and reliability in labeling scientific [polymer] sentences for training data over human experts and state-of-the-art tools aiming to perform the same function.

Table 2. Sentences Missed by Human Experts, Labeled by ELSIE.

Text	Gold Standard	Human Experts	State-of-the-Art Tool	ELSIE
Upon 10 wt % clay loading, the glass transition of the PTMO:MDI–BDO PU nanocomposites shifts slightly from −44.7 to −46.6 °C.[7]	1	0	0	1
The functionalized polycarbonate exhibited a lower Tg of 89 °C compared to its parent (108 °C).[8]	1	0	0	1

The state-of-the-art tool's recall (23%) is much lower than ELSIE's recall (94%) because the state-of-the-art tool missed labeling more positive cases (n=49) than ELSIE (n=4). Given the state-of-the-art tool's objective to extract entities and not label sentences, when the state-of-the-art tool extracted an incorrect polymer-T_g pair, it was penalized and the sentences were not labeled. The state-of-the-art tool would have achieved higher recall (88%) had we focused only on rules-based extraction of the T_g, as opposed to the polymer-T_g pair. However, due to the nuances in complex sentences and complicated polymer naming, it often linked the T_g to an incorrect polymer name [19].

[7] Extracted from: James Korley, LaShanda T, Shawna M Liff, Nitin Kumar, Gareth H McKinley, and Paula T Hammond. "Preferential Association of Segment Blocks in Polyurethane Nanocomposites." Macromolecules 39, no. 20 (2006): 7030-36.

[8] Extracted from: Darensbourg, Donald J, Wan-Chun Chung, Andrew D Yeung, and Mireya Luna. "Dramatic Behavioral Differences of the Copolymerization Reactions of 1, 4-Cyclohexadiene and 1, 3-Cyclohexadiene Oxides with Carbon Dioxide." Macromolecules 48, no. 6 (2015): 1679-87.

Precision for the state-of-the-art tool was higher than ELSIE's because ELSIE labeled false positive sentences. ELSIE looks for entities within a sentence (even if the entities are not related to one another), whereas the state-of-the-art tool looks for related entities. The number of sentences labeled by the state-of-the-art tool was much smaller (n=15) than ELSIE (n=67). The state-of-the-art tool did not label any false positives, whereas ELSIE labeled 7 of the 67 sentences as false positives. An example of a false positive sentence labeled by ELSIE is shown in Table 3; polymer name and glass transition-mention entities were identified, but the temperature entity in the sentence is a melting temperature and not a T_g. Though it is a false positive, reporting this sentence can be beneficial because it could contain important metadata either about the entities of interest, or other polymer characteristics.

Table 3. False Positive Sentence (Labeled as TRUE by ELSIE).

Text[9]	**Gold Standard**	**Human Experts**	**State-of-the-Art Tool**	**ELSIE**
Two or three thermal transitions are expected for SEBS: (1) a low glass transition temperature (Tg1) corresponding to the ethylene-co-butylene block, (2) a high glass transition temperature (Tg2) corresponding to the styrene block, and (3) a broad endothermic transition at the melting temperature (Tm) near 20 °C, depending on the degree of crystallinity of the ethylene-co-butylene block.	0	0	0	1

ELSIE missed labeling sentences that contained entities of a polymer-T_g pair if all three entities were not contained within a single sentence. This demonstrates how and why the problem of finding polymers and their respective T_g is hard for computers and easier for humans. Table 4 shows two consecutive sentences in a text. The first sentence only contains a polymer entity (which ELSIE identified), but did not contain temperature nor glass transition-mention entities; the human identified this sentence and received credit. The other two entities are found in the next sentence, to which the human experts received credit. The state-of-the-art tool extracted the T_g mention from the second sentence, but paired it to the wrong polymer, and did not receive credit for either sentence. Since all three entities were spread among multiple sentences, ELSIE was not able to label either sentence with a 1.

6 Conclusion

This work presented ELSIE, a system that leverages data programming to process scientific articles—specifically in materials science—and identify sentences containing target entities such as polymers, temperatures and glass transition-mentions. We demonstrated that a collection of simple and easy to understand

[9] Extracted from: Mohanty, Angela D., Chang Y. Ryu, Yu Seung Kim, and Chulsung Bae. "Stable Elastomeric Anion Exchange Membranes Based on Quaternary Ammonium-Tethered Polystyrene-B-Poly (Ethylene-Co-Butylene)-B-Polystyrene Triblock Copolymers." Macromolecules 48, no. 19 (2015): 7085-95.

Table 4. True Positive Sentences Missed by LFs.

Text[10]	**Gold Standard**	**Human Experts**	**State-of-the-Art Tool**	**ELSIE**
The azo-polymer material, poly[4‘-[[2-(acryloyloxy)ethyl]ethylamino]-4-nitroazobenzene], often referred to as poly(disperse red 1 acrylate) (hereafter pdr1a), was synthesized as previously reported.	1	1	0	0
The prepared material was determined to have a molecular weight of 3700 g/mol, and a corresponding Tg in the range 95-97 °C.	1	1	0	0

programmed rules are able to detect entity-containing sentences without having to identify the target entities themselves. ELSIE does not use distant supervision—nor a priori known entities—and it does not look for relationship-type words. Instead it determines whether the entities that form the target relationship are present in a sentence. We achieved a recall of 94% when compared to the gold standard, mostly due to an assumption that the three entities are related if they existed in a single sentence. Future work will aim to 1) identify and isolate sentences of interest with their surrounding sentences (e.g., sentences containing 2 out of 3 target entities), and 2) extract polymer entities and their properties.

ELSIE found sentences missed by a best of breed domain-specific toolkit and human experts, whether due to sentences being complicated, such as a sentence containing multiple polymer-T_g pairs, or fatigue/lack of attention paid by human experts. Since ELSIE outperformed a domain-specific toolkit as well as human annotators, this work demonstrates the need for software that can reliably and quickly process polymer texts.

References

1. Audus, D.J., de Pablo, J.J.: Polymer informatics: opportunities and challenges (2017)
2. Brandrup, J., Immergut, E.H., Grulke, E.A., Abe, A., Bloch, D.R.: Polymer handbook, vol. 89. Wiley New York (1999)
3. Chen, L., Friedman, C.: Extracting phenotypic information from the literature via natural language processing. In: Medinfo. pp. 758–762. Citeseer (2004)
4. De Sa, C., Ratner, A., Ré, C., Shin, J., Wang, F., Wu, S., Zhang, C.: Deepdive: Declarative knowledge base construction. ACM SIGMOD Record **45**(1), 60–67 (2016)
5. Friedman, C., Alderson, P.O., Austin, J.H., Cimino, J.J., Johnson, S.B.: A general natural-language text processor for clinical radiology. Journal of the American Medical Informatics Association **1**(2), 161–174 (1994)
6. Friedman, C., Hripcsak, G., Shagina, L., Liu, H.: 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)

[10] Extracted from: Yager, Kevin G, and Christopher J Barrett. "Photomechanical Surface Patterning in Azo-Polymer Materials." Macromolecules 39, no. 26 (2006): 9320-26.

7. Friedman, C., Kra, P., Yu, H., Krauthammer, M., Rzhetsky, A.: Genies: a natural-language processing system for the extraction of molecular pathways from journal articles. In: ISMB (supplement of bioinformatics). pp. 74–82 (2001)
8. Hong, Z., Tchoua, R., Chard, K., Foster, I.: Sciner: Extracting named entities from scientific literature. In: International Conference on Computational Science. pp. 308–321. Springer (2020)
9. Jagannathan, V., Elmaghraby, A.: Medkat: multiple expert delphi-based knowledge acquisition tool. In: Proceedings of the ACM NE Regional Conference. pp. 103–110 (1985)
10. Jessop, D.M., Adams, S.E., Willighagen, E.L., Hawizy, L., Murray-Rust, P.: Oscar4: a flexible architecture for chemical text-mining. Journal of cheminformatics **3**(1), 1–12 (2011)
11. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP. pp. 1003–1011 (2009)
12. Peters, S.E., Zhang, C., Livny, M., Ré, C.: A machine reading system for assembling synthetic paleontological databases. PLoS one **9**(12), e113523 (2014)
13. Ratner, A., Bach, S.H., Ehrenberg, H., Fries, J., Wu, S., Ré, C.: Snorkel: Rapid training data creation with weak supervision. In: Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases. vol. 11, p. 269. NIH Public Access (2017)
14. Ratner, A.J., De Sa, C.M., Wu, S., Selsam, D., Ré, C.: Data programming: Creating large training sets, quickly. In: Advances in neural information processing systems. pp. 3567–3575 (2016)
15. Rocktäschel, T., Weidlich, M., Leser, U.: Chemspot: a hybrid system for chemical named entity recognition. Bioinformatics **28**(12), 1633–1640 (2012)
16. Savova, G.K., Masanz, J.J., Ogren, P.V., Zheng, J., Sohn, S., Kipper-Schuler, K.C., Chute, C.G.: 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)
17. Swain, M.C., Cole, J.M.: Chemdataextractor: a toolkit for automated extraction of chemical information from the scientific literature. Journal of chemical information and modeling **56**(10), 1894–1904 (2016)
18. Tchoua, R.B., Chard, K., Audus, D., Qin, J., de Pablo, J., Foster, I.: A hybrid human-computer approach to the extraction of scientific facts from the literature. Procedia computer science **80**, 386–397 (2016)
19. Tchoua, R.B., Chard, K., Audus, D.J., Ward, L.T., Lequieu, J., De Pablo, J.J., Foster, I.T.: Towards a hybrid human-computer scientific information extraction pipeline. In: 2017 IEEE 13th International Conference on e-Science (e-Science). pp. 109–118. IEEE (2017)
20. Tchoua, R.B., Qin, J., Audus, D.J., Chard, K., Foster, I.T., de Pablo, J.: Blending education and polymer science: Semiautomated creation of a thermodynamic property database. Journal of chemical education **93**(9), 1561–1568 (2016)
21. Van Rossum, G.: The Python Library Reference, release 3.8.6. Python Software Foundation (2020)