Machine extraction of polymer data from tables using XML versions of scientific articles

Hiroyuki Oka a, b, Atsushi Yoshizawa a, Hiroyuki Shindo b, c, Yuji Matsumoto d and Masashi Ishii a

aResearch and Services Division of Materials Data and Integrated System, National Institute for Materials Science (NIMS), Tsukuba, Ibaraki, Japan; bDivision of Information Science, Nara Institute of Science and Technology (NAIST), Ikoma, Nara, Japan; cCenter for Advanced Intelligence Project, RIKEN, Chuo-ku, Tokyo, Japan

ABSTRACT
In this study, we examined machine extraction of polymer data from tables in scientific articles. The extraction system consists of five processes: table extraction, data formatting, polymer name recognition, property specifier identification, and data extraction. Tables were first extracted in plain text. XML versions of scientific articles were used, and the tabular forms were accurately extracted, even for complicated tables, such as multi-column, multi-row, and merged tables. Polymer name recognition was performed using a named entity recognizer created by deep neural network learning of polymer names. The preparation cost of the training data was reduced using a rule-based algorithm. The target polymer properties in this study were glass transition temperature (Tg), melting temperature (Tm), and decomposition temperature (Td), and the specifiers were identified using partial string matching. Through these five processes, 2,181 data points for Tg, 1,526 for Tm, and 2,316 for Td were extracted from approximately 18,000 scientific articles published by Elsevier. Nearly half of them were extracted from complicated tables. The F-scores for the extraction were 0.871, 0.870, and 0.841, respectively. These results indicate that the extraction system created in this study can rapidly and accurately collect large amounts of polymer data from tables in scientific literature.

1. Introduction
Polymers are materials that are present in plastics, fibers, and resins. Considering their abundance, they are important in modern society. Many groups have conducted experiments and used polymer informatics to research polymers actively and to enhance their performance or discover new polymers. Recently, machine learning has been applied to polymer informatics, and this has produced remarkable results [1–5]. Thus, the progress of machine learning algorithms is important for polymer informatics; however, the availability of systematic datasets for machine learning is more important [6]. Such datasets are managed by databases, and our institute (National Institute for Materials Science (NIMS)) has produced some material databases. One such polymer database, PoLyInfo [7,8], contains data such as the names, structures, physical properties, molecular weights, and method for preparation of polymers. These datasets have already been used in machine learning assisted polymer research [2] and in the creation of a benchmark database for polymer informatics [9]. However, to perform a wide variety of polymer research by informatics, larger and denser datasets are desirable. Because the data in PoLyInfo have been collected from scientific publications by polymer

CONTACT
Hiroyuki Oka OKA.Hiroyuki@nims.go.jp Research and Services Division of Materials Data and Integrated System, National Institute for Materials Science (NIMS), 1-1 Namiki, Tsukuba, Ibaraki 305-0044, Japan

© 2021 The Author(s). Published by National Institute for Materials Science in partnership with Taylor & Francis Group
This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial License (http://creativecommons.org/licenses/by-nc/4.0/), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
experts for over 20 years, data collection by machines rather than by human labor will be necessary to increase the amount of polymer data considerably and rapidly. In addition, the quantity of scientific publications has exponentially increased, making it quite difficult to check all of them manually. Therefore, we started to develop a system for extracting polymer data using machines. This was also motivated by the rapid progress of natural language processing (NLP) techniques. Machine learning, particularly deep neural network (DNN) learning, has been actively used for NLP and has led to the development of excellent NLP techniques such as Word2Vec [10], character-based convolutional neural networks (CNNs) [11], bidirectional long short-term memory (BiLSTM) [12], and bidirectional encoder representations from transformers (BERT) [13]. Parsing scientific articles enabled the utilization of such techniques.

Polymer data in scientific articles are described in text, figures, or tables. We first studied the extraction of polymer data from tables, because major components of polymer data are numerical values such as glass transition temperature and the values are frequently condensed in tables using numerical characters. In addition, tabular forms systematically manage the data, which is very convenient for creating algorithms for machine extraction. Although text and figures also bear physical values, processes such as relation extraction and image processing are not easy. However, we have been also researching these objects, because articles do not always include tables. In addition, it will be necessary to relate table data to information in text or figures to perform informatics. Among these objects, tables are the most convenient for starting to develop a data extraction system. However, extracting tabular forms in plain text from PDF files by machines is not easy, because the character positions in PDFs are lost in the process of conversion into plain text. Therefore, data extraction from tables by machines is not easy and the corresponding research has not been intensive. ChemDataExtractor [14], which is an automated chemical data extraction system described in literature, is the only system researched for this purpose thus far. However, this system does not cover complicated tables such as multi-column, multi-row, and merged tables, because such table extraction is considerably more difficult than that of simple tables. Complicated tables are frequently used in scientific articles, which also prevents the extraction of scientific data from tables by machines. One method of solving these problems is to use XML. Although PDF has been a typical electronic format for scientific articles, XML has recently become available. XML is convenient for information extraction by machines, because the XML tags systematically manage the content. By referencing the XML tags, the tabular forms of even the most complicated tables can be accurately extracted in plain text.

The important factor in extracting polymer data from scientific articles by machines is the recognition of polymer names. There are various types of polymer names, such as IUPAC names (e.g. ‘polystyrene’, ‘poly(ethylene oxide)’), abbreviations (‘PS’, ‘SBR’), trade names (‘Nafion’, ‘Kevlar’), common names (‘cellulose’, ‘nylon’), and sample labels (‘P1’, ‘1a’). Additionally, they are varied by joining homopolymer names with hyphens or slashes and including the numerical values of the component ratios and the molecular weights (‘PLA90/NR10/C30B3’, ‘PS(13 kDa)-g-MWNT’). Therefore, the recognition of polymer names is not easy. For such variations, machine learning techniques are effective and have been used for this type of recognition [15,16]. However, because manual preparation of the training data requires high cost, remarkable results have not yet been obtained. In the preparation stage, a rule-based algorithm may be useful for recognizing IUPAC names and their abbreviations, considering the character patterns that usually begin with ‘poly’ in the former and frequently include ‘P’ in the latter. As IUPAC names and their abbreviations are significantly important in polymer names, if they are recognized, then most of the polymer data can be extracted. By reducing the recognition targets, the rule-based algorithm can be helpful.

Based on the above descriptions, our objective in this study was to create a system to extract polymer data from tables, and the corresponding results are reported in this paper. The important factors in creating the system were considered to be table extraction from scientific articles and polymer name recognition. Table extraction was examined using XML versions of scientific articles to enable accurate extraction of not only simple tables, but also complicated tables. Because publishers have started to provide XML files for text and data mining research recently, we could obtain many such documents. Polymer name recognition was examined using DNN learning. Because the polymer names in tables frequently include information such as the component ratios for blends and composites as well as synthetic conditions of temperatures, solvents, and catalysts, DNN learning is useful. The training data were prepared using a rule-based algorithm to avoid the high cost of manual preparation. Using the rule-based algorithm, mainly the IUPAC names and their abbreviations in polymer text were annotated. To extract polymer data, specifiers of polymer properties must be identified. In this study, glass transition temperature (Tg), melting temperature (Tm), and decomposition temperature (Td) were selected as the target properties. The specifiers were identified by partial string matching using the index terms
and stop words previously prepared by polymer experts. Consequently, the target polymer data can be identified by combining the identification of the specifiers with the recognition of polymer names. This ensures that the full names for polymer abbreviations and sample labels, the synthetic procedures and the molecular weights of the polymer samples, the measurement methods to obtain the property values, etc., were not extracted in association with the polymer data. Although these data are important for performing informatics, extracting the relations between them and table data will significantly increase the difficulty of extraction. As such, this study is focused on extracting raw polymer data from tables as the first stage of development.

### 2. Experiment

#### 2.1. XML versions of scientific articles

In this study, XML versions of scientific articles published between 2015–2017 by Elsevier were used. The journal titles and the numbers of articles are summarized in Table 1. These XMLs were purchased for text data mining.

#### 2.2. System overview

An overview of the data extraction system in this study is shown in Figure 1. The system consists of five processes: table extraction, data formatting, polymer name recognition, property specifier identification, and data extraction. Polymer name recognition was performed using a named entity recognizer created in the DNN learning of polymer names. The training data were prepared using a rule-based algorithm. The details of each process are described below.

#### 2.3. DNN learning for polymer name recognition

##### 2.3.1. Preparation of training data by rule-based algorithm

In this study, the training data for DNN learning of polymer names were prepared using a rule-based algorithm. The procedure is as follows. The algorithms for IUPAC names and their abbreviations were created based on their character patterns, i.e. IUPAC names usually begin with 'poly' while their abbreviations frequently include 'P'. Because parentheses and curly square brackets frequently follow immediately after 'poly' in IUPAC names, this regularity was used. Besides the two types of names, algorithms were created for the sample labels described in parentheses or square brackets following immediately after polymer full names (usually IUPAC, common, or trade names), and for copolymers and blends, which are named by joining homopolymer names by slashes or hyphens. This was done using their regularities. Additionally, an algorithm for analyzing 'copolymer of A and B' and 'A and B copolymer' was also created. The typical polymer names listed in polymer books, such as 'SBR' (styrene butadiene rubber) and 'cellulose,' and frequently used polymer names were registered and identified by string matching. Stop words were also registered to avoid incorrect annotation of words including 'poly' and 'P' other than polymer names. Because polymer names occasionally include modifiers, such as 'isotactic' and 'doped,' an algorithm for identifying such modifiers was also created by registering them or using regular expressions. Using the rule-based algorithm, polymer names in 737 polymer articles published between 2000–2008 were annotated.

![Figure 1. Overview of data extraction system in this study.](image_url)
The rule-based algorithm was used for tokenized text prepared using PDFBox [17] for plain text conversion and Stanford Core NLP [18] for tokenization from the PDF files of polymer articles. Sequence labeling was applied to the annotation output using B, I, E, S, and O. S stands for a single-token polymer name; B, I, and E represent the beginning, intermediate, and end tokens in a polymer name; and O stands for a token unrelated to the polymer name. The output files consisted of two tab-separated columns for tokens and labels with a blank line after a period for sentence splitting.

### 2.3.2. DNN learning

Using the annotated training data prepared as described in the previous section, DNN learning was performed to create a named entity recognizer for polymer names. In this study, the DNN architecture studied by Dos Santos and Zadrozny [11] was used. This architecture was created for part-of-speech tagging using the word- and character-level representations for a word extracted by CNNs. The architecture has been reproduced on Merlin [19,20], which is the DNN learning framework created by the Nara Institute of Science and Technology and written in Julia. The architecture can perform calculations more rapidly on Merlin than on other frameworks such as Theano and Chainer [20] and can also perform BIESO sequence labelling in addition to part-of-speech tagging. Thus, the DNN learning in this study was performed on Merlin. The details of the architecture are described in references [11,20]. The processing of the input text in the architecture can be summarized as follows. Firstly, given a sentence of words, every word is converted into a 100-dimensional dense vector. The dense vectors, also known as the word embeddings, have been pre-trained by gensim Word2vec [21] using approximately 102 M words in chemical domain extracted from the journals of the American Chemical Society published between 2016–2017. In addition, a character-level CNN and max-pooling operation are applied to each word to extract the character-level representations. After that, the character-level vectors and the word embeddings are concatenated, and a word-level CNN is applied to the feature vectors. Finally, an affine transformation is applied to the feature vectors to predict the BIESO labels. The kernel sizes of both character- and word-level CNNs are set to be five. The model is trained to predict the correct BIESO labels for each sentence. In this study, the size of the training dataset varied (i.e. 279, 588, and 737 articles) to check the sizing effect. Each dataset was divided in a ratio of 5:1 for training and testing, respectively. The training was performed for up to 20 epochs, following which the model (named entity recognizer) for each training was created. The best recognizer for data extraction was used in this study.

### 2.4. Data extraction system

#### 2.4.1. Table extraction

To extract polymer data from tables, the tabular forms were first extracted from XML versions. Because Elsevier’s XMLs describe tables using `table` elements, they were extracted from these elements. First, the tabular framework was constructed by identifying the numbers of columns and rows. These numbers were identified from the attributed value in the `tgroup` elements and by counting the number of `row` elements, respectively. For rows, the number of the header rows was also identified, which is useful for property specifier identification. In XMLs, the table header and body parts are clearly described in the `thead` and `tbody` elements, respectively. The table cell data were then extracted from the `entry` elements and embedded into the framework. Here, all XML tags for character styles, such as italics, bold, superscript, and subscript, were removed owing to their inconvenience for string matching. Additionally, for a blank cell datum, ‘non’ was embedded as a dummy datum, because a blank causes an execution error in polymer name recognition. The multi-column and multi-row cells were split, and then the table cell datum was embedded into all split cells; however, the duplicate embedding was not done for the multi-column cells in the table body and for the multi-row cells in table header to avoid incorrect data extraction; thus, the table cell datum was embedded into the leftmost or topmost split cell and ‘non’ was embedded into other split cells. The multi-column and multi-row cells were identified by the attributes `name` or `nameend` and `morerows` in the `entry` elements, respectively. The numbers of multi-column and multi-row tables were counted by these attributes. For merged tables, each component table was independently extracted as a table to simplify the extraction of polymer data after the second component tables. The `thead` and `tbody` elements for each component table can be identified in XMLs. This was used to count the number of merged tables.

#### 2.4.2. Data formatting

After table extraction, the table cell data in the tabular form were aligned in a single line for machine readability. Triple sets consisting of each table cell datum in the index columns, header rows, and intersecting table cells, which might be polymer data, were extracted and aligned as follows:

```plaintext
{(2, 1), (1(head), 2), (2, 2)}, {(2, 1), (1(head), 3), (2, 3)}, *, *, *, {(3, 1), (1(head), 2), (3, 2)}, {(3, 1), (1(head), 3), (3, 3)}, *, *, *, *, *, {(2, 2), (1(head), 3),
```
(2, 3), [(2, 2), (1(\text{head}), 4), (2, 4)], \ast, \ast, \ast, \{3, 2\}, (1 (\text{head}), 3), (3, 3\}, [(3, 2), (1(\text{head}), 4), (3, 4)], \ast, \ast, \ast, \ast, \ast.

The numbers in parentheses indicate the row and column addresses, respectively, and ‘head’ and \{\} denote the header row and a triple set, respectively. The leftmost and second columns were regarded as the index columns, because polymer names are generally described there. These triple sets were useful in data extraction.

2.4.3. Polymer name recognition
Polymer names in tables were recognized using the named entity recognizer created in DNN learning. The data formatted in the previous section were run through the recognizer, and then BIESO labeling was applied to the table cell data. The labels almost always resulted in S or O, because each table cell datum was run through without tokenization.

2.4.4. Property specifier identification
After polymer name recognition, property specifiers were identified by partial string matching with the index terms and stop words for $T_g$, $T_m$, and $T_d$. The index terms and stop words were manually investigated and listed. The lists are provided in the supplemental data. As the output, each table cell datum was labeled S (property specifier) or O (others). The resultant labels in polymer name recognition and property specifier identification were given as follows:

- table cell datum: ['PMMA', 'Tg\textdegree C', '60'], \ast, \ast, \ast
- label (polymer name recognition): \{S, O, O\}, \ast, \ast, \ast
- label (property specifier identification): \{O, S, O\}, \ast, \ast, \ast

2.4.5. Data extraction
After property specifier identification, polymer data were extracted by checking the resultant labels from polymer name recognition and property specifier identification. Before checking, the resultant polymer name recognition labels were modified in the following cases:

1) When both resultant labels in polymer name recognition and property specifier identification for a table cell datum were S, then the former resultant label was changed to O, because the datum was usually a property specifier. A property specifier in tables occasionally includes a polymer name (e.g. 'Tm (\textdegree C) of EPDM').

2) When the S labels in polymer name recognition were given to table cell data in both the leftmost and second index columns, then the column with more S labels was selected as the polymer name column, and all labels in another column were changed to O, because the polymer name column is generally single. In the case of the same number of S labels in the leftmost and second index columns, the leftmost column was selected as the polymer name column.

3) When the resultant S labels in polymer name recognition were over 30\% of the table cell data in the column, all labels in the column were changed to S. In such cases, all table cell data in the column were usually polymer names.

After these modifications, the resultant labels in the polymer name recognition and property specifier identification of each table cell datum were first checked to determine if the table cell datum is a polymer name, a property specifier, or other information. When the two resultant labels were combined to S and O, O and S, and O and O, then the table cell datum was identified as a polymer name, a property specifier, and other information, respectively. Then, the triple set including a polymer name (a property specifier), a property specifier (a polymer name), and other information as the first, second, and third table cell data, respectively, was extracted as a polymer data point. However, when the third table cell datum did not include numerical characters, indicating that it is not a property value, then the triple set was not extracted. In the case of the resultant labels shown in Section 2.4.4, the triple set was extracted as a polymer data point.

3. Results and discussion
3.1. DNN learning for polymer name recognition
3.1.1. Polymer name annotation by rule-based algorithm
Using the rule-based algorithm, polymer names in 737 polymer articles were annotated. In the annotated data, many IUPAC polymer names and their abbreviations were included, and most of them were accurately annotated. Author-defined names beginning with ‘poly’, such as ‘poly(1a)’ and ‘Poly-P-a-co-Th’, were also annotated. Sample labels describing in parentheses or square brackets immediately after the polymer full names, the names joining homopolymer names by slashes or hyphens and including numerical values and modifiers, and the names registered in the algorithm were occasionally annotated. Polymer names covered by the rule-based algorithm were essentially annotated. In contrast, incorrect annotations and exclusions of polymer names were significant. Incorrect annotations frequently included chemical abbreviations and formulas including ‘P’ (e.g. ‘PPh3’). For the exclusion of polymer names, the names composed of two or more tokens (‘PVAc : LiClO4’ and ‘multi-walled carbon nanotubes / high-density polyethylene’) were not annotated, and only single-token polymer names were annotated. There were many such names in polymer articles; thus,
there was a significant number of excluded names, which was inevitable in this study. Although there were incorrect annotations and exclusions of polymer names, IUPAC polymer names and their abbreviations were expected to result in good training in DNN learning due to their abundance in the annotated data. The list of polymer names annotated in 103 polymer articles is presented with the evaluation of the annotation in the supplemental data.

3.1.2. **DNN learning**

Following the preparation of annotated data, the DNN learning of polymer names was performed by varying

![Figure 2](image)

**Figure 2.** Training curve in DNN learning of polymer names.

![Table 3](image)

**Table 3.** Extracted multi-column and multi-row table. Only a part of the table is shown because of its large size. This table was extracted from the following XML document: article-DOI: 10.1016/j.polymer.2016.01.041.

![Table 4](image)

**Table 4.** Extracted merged table. These tables were extracted from the following XML document: article-DOI: 10.1016/j.apsusc.2015.06.039. A value in ‘thead = ’ denotes the number of header rows. The values in ‘multi_col = ’, ‘multi_row = ’, and ‘merged = ’, which are 0 or 1, indicate whether the table is a complicated table. ‘(Footnote) non’ is used to denote the absence of a footnote.
the size of the training dataset among 279, 588, and 737 articles to check the sizing effect. The values of precision, recall, and F-score were near or over 0.9 after 20 epochs for all datasets. The training curve for the 588-article dataset is plotted in Figure 2. Loss started from 5.3 and fell smoothly thereafter. Accordingly, the values of precision, recall, and F-score increased and became saturated after 10 epochs. Similar trends were observed for the curves of the differently sized datasets. Among the three datasets, the 588 article dataset attained the best named entity recognizer for data extraction. The ability of the recognizer is described in Section 3.2.2, because it was checked in the data extraction evaluation.

### 3.2. Data extraction system

#### 3.2.1. Table extraction

To extract polymer data, tables were first extracted from XML versions of the scientific journals in Table 1. From 13,185 articles including tables, 30,878 tables were extracted. Among them, the numbers of multi-column, multi-row, and merged tables were 9617, 5754, and 307, respectively, indicating that their numbers were not small; thus, it is important not to exclude the data in these tables for the collection of large amounts of scientific data. The extraction accuracy was manually checked using 160 arbitrarily selected tables; all tables were accurately extracted. Examples of the extracted tables are shown in Figures 3 and 4. Figure 3 shows a multi-column and multi-row table. The multi-column and multi-row cells are indicated by the red and blue frames, respectively. Notably, the datum in the multi-column or multi-row cell is described in all the split cells. Thus, a duplicate description ensures that polymer data are not excluded. For example, three values (‘162,’ ‘157,’ and ‘153’) of glass transition temperature (‘Tg (°C)’) for ‘BA-a’ were extracted from this table. However, the duplicate description is not provided for the multi-row cells in table header; thus, the table cell datum is described only in the topmost split cell, which is indicated by the dotted blue frames. This was done to avoid regenerating the same polymer data. In any case, the four table cells, where ‘245’ is listed as the value of ‘Tmax (°C)’ for ‘BA-a,’ were not originally spanned in the XML. In Figure 4, the component tables of merged table are shown. Because the original merged table had two component tables, two tables were extracted. The separation of each component table ensured the extraction of polymer data after the second component table.

#### 3.2.2. Data extraction

After data formatting of the extracted tables, polymer name recognition, and property specifier identification, polymer data were extracted. From approximately 18,000 articles in the journals in Table 1, 2181, 1526, and 2316 data points for Tg, Tm, and Td were extracted, respectively. The numbers correspond to 3%–9% of those in PolylInfo, which are 60,057, 34,594, and 27,899 data points, respectively. The execution time was approximately three days using a personal computer, which could be reduced to less than a few hours by combining all output files after table extraction into a few files to decrease input/output file operation. Thus, the data extraction system created in this study can rapidly collect large amounts of polymer data. In any case, the polymer data in PolylInfo have been collected for over 20 years. Among the extracted data points, 994 (46%), 611 (40%), and 1008 (44%) data points were extracted from complicated tables, representing nearly half of the extracted data points. Because complicated tables constituted approximately 30% of the extracted tables, as described in the previous section, the results are adequate. The results indicate that the data extraction system does not exclude the data in complicated tables, due to the use of XML versions of scientific articles and accurate table extraction.

Evaluation of the data extraction was performed using the correct answer data manually extracted from 100 articles in Polymer (one polymer journal) and from 100 articles in other journals (non-polymer journals) and compared with the evaluation results in polymer and non-polymer journals. In these articles, tables that do not bear Tg, Tm, and Td data were included to check for incorrect data extraction. The division of polymer and non-polymer journals was taken into consideration with the idea that polymer name recognition would be less accurate in non-polymer journals than in a polymer journal. This is because chemical abbreviations with ‘P’ other than for polymer names are more common in non-polymer journals than in polymer journals. In the evaluation, the numbers of true positive (TP), false positive (FP), and false negative (FN) were counted, and the precision, recall, and F-score were estimated using the following equations: precision = TP/(TP + FP), recall = TP/(TP + FN), and F-score = 2 x precision x recall/(precision + recall). The evaluation results in all journals, one polymer journal, and non-polymer journals are shown in Tables 2–4. All F-scores in Table 2 were over 0.84. The precisions for Tg and Tm in Table 3 were over 0.94, indicating excellent performance. These results were

| Table 2. Evaluation results for data extraction from all journals. |
|-------------------|--------|-----|--------|-----|
|                  | TP/FP/FN | PRECISION | RECALL | F-score |
| Tg               | 824/96/148 | 0.896 | 0.848 | 0.871 |
| Tm               | 959/105/181 | 0.901 | 0.841 | 0.870 |
| Td               | 1143/282/150 | 0.802 | 0.884 | 0.841 |

| Table 3. Evaluation results for data extraction from one polymer journal. |
|-------------------|--------|-----|--------|-----|
|                  | TP/FP/FN | PRECISION | RECALL | F-score |
| Tg               | 540/28/87 | 0.951 | 0.861 | 0.904 |
| Tm               | 675/40/122 | 0.944 | 0.847 | 0.893 |
| Td               | 772/133/102 | 0.853 | 0.883 | 0.868 |
due to good cooperation with polymer name recognition, modification of its resultant labels, property specifier identification, and inclusion of numerical characters as the property values. In polymer name recognition, IUPAC names and their abbreviations could be recognized well. This was due to their abundance in the training data used for the DNN learning of polymer names. The character-level CNN seemed to be very effective, because character patterns in polymer names are peculiar in comparison with general English words. Occasionally, polymer sample labels, such as ‘P1,’ were also recognized. In contrast, incorrect polymer name recognition and exclusion of polymer names also occurred. Therefore, the modification of resultant labels was significantly effective, attaining an increment of 433 for TP, while 135 for FP, in all properties and journals. Polymer names with slightly different expressions showed significantly different results; for example, ‘PU75’ was recognized as a polymer name, while ‘PU60’ in the same column was not. The modification rescued such polymer names.

There was a significant amount of incorrect polymer name recognition in non-polymer journals; thus, the evaluation scores in non-polymer journals (Table 4) are lower than in polymer journals (Table 3). This was because there are many chemical abbreviations that include ‘P’ but are not polymer name abbreviations; thus, they are incorrectly recognized. Examples are ‘BPbP’ (low molecular weight organic compound), ‘LiFePO4/C’ (inorganic composite), and ‘HCP’ (crystal packing, ‘hexagonal close packing’). Owing to such incorrect recognition, the precisions were reduced for all properties. In contrast, incorrect exclusion of polymer names occurred across the three properties in both polymer and non-polymer journals. Polymer abbreviations not including ‘P’ (e.g. ‘SBR41/19’) and in which lowercase characters are dominant (‘Neat PP pellet’), and sample labels were frequently excluded. Such names were not abundantly annotated by the rule-based algorithm; thus, they were not trained in the DNN learning.

The precisions for T_d in Tables 2–4 are generally lower than the other two properties, which is due to incorrect identification of the property specifiers. Owing to the wide variety of expressions, the identification was not easy. For T_d, the numerical values denoting a weight loss percentage of samples in the measurement are frequently used, and the expression and the value were varied (e.g. ‘5% weight loss’ and ‘T10’). Additionally, the specifiers, such as ‘Onset temperature’ and ‘Tmax,’ which are based on features on the measurement curve, are frequently used as they increase the variety for the T_d specifiers. In contrast, the specifiers for T_g and T_m were not as variable; thus, the identification worked well.

The results in this section indicate that the data extraction system created in this study can rapidly collect large amounts of polymer data from tables in scientific articles with high evaluation scores. Although substantial polymer name recognition errors occur, we will address this issue in future research.

### 3.3. Visual representation of extracted polymer data

Because most of the extracted polymer data were considered to be true positive based on the precisions in Table 2, the polymer data were visually represented in graphs for comparison with the data in PoLyInfo and to demonstrate informatics analysis. To conduct the representation, character clearance was performed, except for numerical characters in the property values, such as ‘>,’ and the units were made consistent as degrees Celsius. The details of these processes are not described in this paper.

#### 3.3.1. Comparison with data in PoLyInfo

To compare the extracted polymer data with the data in PoLyInfo, their frequency distributions were investigated in intervals of 50 °C. The results are shown in Figure 5. All distributions are normal distributions, although the frequency at ~49–0 °C for T_g is outstanding. The distributions of the data obtained in this study are similar to those of the PoLyInfo data, indicating that the data points extracted in this study cover a temperature range similar to that of the PoLyInfo data, although the amounts of data from this study are only 3%–9% of the corresponding amounts of PoLyInfo data. The frequency distribution results also indicate that polymer name recognition covering mainly IUPAC names and their abbreviations was adequate to obtain such distributions. However, recognizing all types of polymer names is the best, and we will consider this issue in future research. The data extracted in this study also covered the sub-classes of polymers such as homopolymers, copolymers, blends, and composites as well as PoLyInfo, as indicated by the fact that 103 (37%), 131 (38%), 9 (3%), and 60 (22%) data points among the 303 T_g data points were classified as the respective sub-classes. In any case, the numbers of polymers for the homopolymer, copolymer, blend, and composite sub-classes in PoLyInfo are 17,328 (60%), 6595 (23%), 2145 (8%), and 2649 (9%), respectively.

As another comparison, the pairs of T_g and T_m data points for a polymer sample were plotted in a T_g–T_m plot. PoLyInfo has approximately 8400

---

**Table 4. Evaluation results for data extraction from non-polymer journals.**

| Property | TP/FP/FN | PRECISION | RECALL | F-score |
|----------|----------|-----------|--------|---------|
| T_d      | 284/68/51| 0.807     | 0.823  | 0.815   |
| T_m      | 284/65/59| 0.814     | 0.828  | 0.821   |
| T_d      | 371/149/48| 0.714   | 0.885  | 0.790   |
pairs, and the $T_g$-$T_m$ plot generally exhibits a linear relationship, as shown in Figure 6 (gray circles); thus, the 425 pairs obtained in this study were plotted together (orange circles). The orange circles are mostly distributed in the range of PoLyInfo and show a linear relationship as well as the PoLyInfo data, although there are several outliers and $T_g > T_m$. Because the precision for data extraction of both $T_g$ and $T_m$ is approximately 0.9, as shown in Table 2, most of the pairs of $T_g$ and $T_m$ data points are correct. Thus, it is appropriate to obtain a linear relationship in the $T_g$-$T_m$ plot.

These results indicate that the data extraction system in this study can extract polymer data as well as PoLyInfo data collection. Although significantly fewer data were obtained in this study than in PoLyInfo, the amounts of data can be easily increased by increasing the number of scientific articles considered due to the rapid execution of the data extraction system. The system can instantly process numerous scientific articles, which is a great advantage over manual data collection. In addition, the system can extract polymer data for properties other than $T_g$, $T_m$, and $T_d$ simply by preparing the index terms and stop words to identify the property specifiers. In any case, approximately 100 polymer properties are prescribed in PoLyInfo.

3.3.2. Informatics analysis demonstration

Because the data for polymers including polylactide (PLA) and poly(ethylene oxide) (PEO) frequently

---

**Figure 5.** Frequency distribution of data points extracted in this study and recorded in PoLyInfo in intervals of 50 °C. (a) $T_g$, (b) $T_m$, and (c) $T_d$.
occurred in the extracted data, informatics analysis was roughly demonstrated using these data. The corresponding data points were obtained from the extracted data points by partial string matching with 'PLA,' 'PLLA,' or 'PDLA' for PLA and 'PEO' or 'PEG' for PEO in the polymer names. The numbers for $T_g$, $T_m$, and $T_d$ were 191, 228, and 127 for PLA and 135, 144, and 27 for PEO, respectively. A data ID number was arbitrarily given to each data point, and the data points were plotted using the IDs, as shown in Figure 7. Both Figure 7 (a,b) show clear data lines for $T_g$ and $T_m$, indicating a clear difference between the $T_g$ and $T_m$ values for PLA- and PEO-based polymers, which are approximately 60 and 165 °C for PLA and approximately -45 and 50 °C for PEO, respectively. In contrast, the data points for $T_d$ are widely dispersed over 200 °C in both Figure 7(a,b). One of the reasons for this dispersion is that the definition used to obtain $T_d$ is different in each article, e.g. '5% weight loss' or '10% weight loss.' Nevertheless, two data lines are clearly observed at approximately 340 and 370 °C in Figure 7(a), which is due to the fact that two stages of decomposition were measured in ternal polymer blends. In addition, for $T_m$ in Figure 7(a), two data lines are observed at approximately 165 and 215 °C, although the higher $T_m$ line is short. The lower and higher $T_m$ values are due to differences in the crystal structures.

Such analysis and its causal information will be useful for designing heat-resistant materials. However, further information must be also extracted from text or figures with related table data to perform informatics properly, which corresponds to the next stage of development; thus, the informatics analysis described in this paper is only rough analysis. This study was focused on extracting raw polymer data from tables, and we consider this objective to have been mostly achieved based on the results presented in this paper.

4. Conclusions

Machine extraction of polymer data from tables in scientific articles was examined using XML versions. A system was created using five processes:
table extraction, data formatting, polymer name recognition, property specifier identification, and data extraction. The system extracted 2181 data points for \( T_g \), 1526 for \( T_m \), and 2316 for \( T_d \) from approximately 18,000 articles published by Elsevier. Nearly half of them were extracted from complicated tables; XMLs were used to process these tables easily. The F-scores for the extraction were over 0.84 for all properties. The precisions for \( T_g \) and \( T_m \), from a polymer journal were over 0.94, implying excellent results. This was due to good cooperation between polymer name recognition and property specifier identification. In polymer name recognition, the recognizer created by DNN learning and preparation of the training data using the rule-based algorithm worked well. The extracted polymer data exhibited distributions similar to those of PoLyInfo data. From these results, we consider the first stage of development to have mostly been achieved; however, we will work on improving the polymer name recognition described in this paper in future research. This achievement has already enabled us to start to extract polymer data for properties other than \( T_g \), \( T_m \), and \( T_d \) with increasing scientific articles and to research table extraction from PDFs containing more scientific data than XML files to increase the amount of polymer data. The extracted raw table data must be run through processes such as graphical representation and related to information in text or figures to perform polymer informatics. Performing such processes using machines constitutes the next stage of development.

**Notes**

1. Supplemental data are stored in our repository with the following DOI: [http://dx.doi.org/10.11503/nims.1190](http://dx.doi.org/10.11503/nims.1190). Programs for polymer name annotation and data extraction are stored in the following GitHub repository: [https://github.com/OkaNim/table-polymer-data-extractor/tree/master](https://github.com/OkaNim/table-polymer-data-extractor/tree/master).

**Acknowledgments**

We would like to thank the NIMS Text and Data Mining Platform and the staff for the use of XML versions of scientific articles.

**Author contributions**

H.O. created the rule-based algorithm for polymer name annotation and the data extraction system, prepared correct answer data to evaluate the data extraction, and drafted the manuscript. A.Y. prepared the index terms and stop words for property specifier identification and correct answer data to evaluate the data extraction. H.S. and Y.M. performed the DNN learning of polymer names. M.I. supervised this study and drafted the manuscript. All authors discussed the results, were involved in the development of the study, and contributed to the final manuscript.

**Disclosure statement**

The authors have no conflicts of interest directly relevant to the content of this article.

**ORCID**

Hiroyuki Oka [http://orcid.org/0000-0002-1768-2429](http://orcid.org/0000-0002-1768-2429)
Hiroyuki Shindo [http://orcid.org/0000-0003-1081-9194](http://orcid.org/0000-0003-1081-9194)
Yuji Matsumoto [http://orcid.org/0000-0003-4946-9574](http://orcid.org/0000-0003-4946-9574)
Masashi Ishii [http://orcid.org/0000-0003-0357-2832](http://orcid.org/0000-0003-0357-2832)

**References**

[1] Kim C, Chandrasekaran A, Huan TD, et al. Polymer genome: a data-powered polymer informatics platform for property recognitions. J Phys Chem C. 2018;122(31):17575–17585.
[2] Wu S, Kondo Y, Kakimoto M, et al. Machine-learning-assisted discovery of polymers with high thermal conductivity using a molecular design algorithm. NPJ Comp Mater. 2019;5:Article Number 66.
[3] Hatakeyama-Sato K, Tezuka T, Nishikitani Y, et al. Synthesis of lithium-ion conducting polymers designed by machine learning-based recognition and screening. Chem Lett. 2019;48(2):130–132.
[4] Cravero F, Schustik SA, Martínez MJ, et al. Feature selection for polymer informatics: evaluating scalability and robustness of the PS4RVDD algorithm using synthetic polydisperse data sets. J Chem Inf Model. 2020;60(2):592–603.
[5] Chen G, Shen Z, Iyer A, et al. Machine-learning-assisted de novo design of organic molecules and polymers: opportunities and challenges. Polymers. 2020;12(1):163.
[6] Audus DJ, De Pablo JJ. Polymer informatics: opportunities and challenges. ACS Macro Lett. 2017;6(10):1078–1082.
[7] [https://polymer.nims.go.jp/](https://polymer.nims.go.jp/)
[8] Otsuka S, Kuwajima I, Hosoya J, et al. PoLyInfo: polymer database for polymeric materials design. Proceedings of International Conference on Emerging Intelligent Data and Web Technologies; 2011 Sep 7–9, Tirana (Albania): IEEE Xplore; 2011. p. 22–29.
[9] Ma R, Luo T. PLIM: a benchmark database for polymer informatics. J Chem Inf Model. 2020;60(10):4684–4690.
[10] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality. Proceedings of the 26th International Conference on Neural Information Processing Systems; 2013 Dec 5–10, Lake Tahoe, Nevada, USA. Volume 2. p 3111–3119.
[11] Dos Santos CN, Zadrozny B. Learning character-level representations for part-of-speech tagging. Proceedings of the 31st International Conference on Machine Learning; 2014 Jun 21–26; Beijing, China: JMLR 32(2):1818–1826.
[12] Ma X, Hovy E. End-to-end sequence labeling via bi-directional lstm-cnns-crf. Proceedings of the 54th Annual Meeting of the Association for Computational
Linguistics. 2016 Aug 7–12; Berlin, Germany. Volume 1: Long Papers. p 1064–1074.

[13] Devlin J, Chang M-W, Lee K, et al. BERT: pre-training of deep bidirectional transformers for language understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2019 Jun 2–7; Minneapolis, USA. Volume 1: Long and Short Papers. p 4171–4186.

[14] Swain MC, Cole JM. ChemDataExtractor: a toolkit for automated extraction of chemical information from the scientific literature. J Chem Inf Model. 2016;56(10):1894–1904.

[15] Krallinger M, Leitner F, Rabal O, et al. CHEMDNER: the drugs and chemical names extraction challenge. J Chem Info. 2015;7(Suppl 1):S1.

[16] Tchoua RB, Ajith A, Hong Z, et al. Creating training data for scientific named entity recognition with minimal human effort. Proceedings of International Conference on Computational Science; 2019 Jun 12–14; Algarve, Portugal. p. 398–411.

[17] https://pdfbox.apache.org/

[18] https://stanfordnlp.github.io/CoreNLP/

[19] https://github.com/hshindo/Merlin.jl (Branch: nims)

[20] Shindo H, Sawai Y, Ouchi H, et al. Implementation and evaluation of deep learning library in Julia language. Proceedings of the 30th Annual Conference of the Japanese Society for Artificial Intelligence; 2016 Jun 6–9; Nagoya, Japan. Presentation No. 1E5-1in1.

[21] https://radimrehurek.com/gensim/models/word2vec.html