Article

Predicting Polymers' Glass Transition Temperature by a Chemical Language Processing Model

Guang Chen [1] **, Lei Tao** [1] **and Ying Li** [1,2,*]

1 Department of Mechanical Engineering, University of Connecticut, Storrs, CT 06269, USA; guang.chen@uconn.edu (G.C.); lei.tao@uconn.edu (L.T.)
2 Polymer Program, Institute of Materials Science, University of Connecticut, Storrs, CT 06269, USA
* Correspondence: yingli@engr.uconn.edu

Abstract: We propose a chemical language processing model to predict polymers' glass transition temperature (T_g) through a polymer language (SMILES, Simplified Molecular Input Line Entry System) embedding and recurrent neural network. This model only receives the SMILES strings of a polymer's repeat units as inputs and considers the SMILES strings as sequential data at the character level. Using this method, there is no need to calculate any additional molecular descriptors or fingerprints of polymers, and thereby, being very computationally efficient. More importantly, it avoids the difficulties to generate molecular descriptors for repeat units containing polymerization point '*'. Results show that the trained model demonstrates reasonable prediction performance on unseen polymer's T_g. Besides, this model is further applied for high-throughput screening on an unlabeled polymer database to identify high-temperature polymers that are desired for applications in extreme environments. Our work demonstrates that the SMILES strings of polymer repeat units can be used as an effective feature representation to develop a chemical language processing model for predictions of polymer T_g. The framework of this model is general and can be used to construct structure–property relationships for other polymer properties.

Keywords: polymer informatics; machine learning; glass transition temperature; high-throughput screening; recurrent neural network

1. Introduction

Glass transition temperature (T_g) of polymers is an important physical property, which has been studied extensively in polymer science and engineering [1–6]. T_g characterizes a second-order phase transition over which polymers can change between a rubbery state and a glassy state with Young's modulus ranging from MPa to GPa [7]. Thus, T_g values determine the ease of processing during manufacturing and the application ranges in their deployment. Theoretical studies have provided many chemical and physical insights into T_g of polymers, from thermodynamics to kinetics theories [4,8–10]. It is well known that T_g value is dependent on the chain mobility or free volume of a polymer [9]. Specifically, it depends on the molecular weight, cross-links, side groups, and chain ends of a polymer. Though theoretical studies have offered critical understandings of polymer's glass transition, it is still deficient for accurate predictions of T_g of general polymers and not effective for inverse polymer design.

While experiments and computer simulations, e.g., molecular dynamics [11–14], are feasible approaches to quantify the T_g of polymers, the data sizes, and sample types that can be handled by these methods are usually limited due to the significant cost in experimental or computational measurements. Nonetheless, these measurements have provided a diversified polymer database that can be leveraged by data-driven studies.

In general, data-driven studies try to construct a mapping relation between the polymer's chemical structures to the corresponding T_g or other properties [15–18]. The development of quantitative structure-property relationships (QSPR) of polymers have significantly

benefited quantitative predictions of polymer's T_g [19–21]. This type of studies has also been called polymer informatics [22–25]. Recently, thanks to the advances in computing power and the availability of big data, machine learning (ML), especially deep learning (DL), has attracted enormous attentions in various scientific fields and indeed brought in numerous breakthroughs in material science [17,26–31] and drug discovery [32–35]. However, it is not the case when it comes to polymer science and engineering, such as polymer's T_g prediction and other properties.

The main reason is that the database of polymers with high quality is very limited. In polymer literature, the database in most of previous studies were under a few hundreds or even less [36]. Therefore, DL models were not widely applied in these studies. It is because DL models usually have a large amount of parameters and thus are easy to over fit if trained on a limited amount of data [37]. Nevertheless, there are a few previous studies employing DL for polymer's T_g prediction. For example, the deep neural network (DNN) model [37,38] and convolutional neural network (CNN) model [39] have been recently employed to correlate polymer's chemical structure (monomers) and its T_g, although the data size in these studies are rather limited. Very recently, Nazarova et.al. studied the dielectric property of polymers using the recurrent neural network (RNN) on 1200 polymers, though the model was only tested on 60 samples [40]. Note that DL models have widely been used for another type of tasks without labeled polymer properties, i.e., molecular generation using deep generative models [29,31,41–44]. This kind of tasks is to use deep generative models to learn the conditional probabilities of the SMILES strings [45–47] of organic molecules. The task in this study is a supervised learning of the syntax of SMILES strings for polymer's T_g prediction.

To develop DL models with good performances for polymer T_g prediction, a large amount of polymer data is necessary since DL models usually have a large number of parameters and thus are easy to overfit. Recently, a polymer database, called PolyInfo [48,49], has attracted much attention as it contains about 7000 homopolymers with experimentally measured T_g values. However, since the database uses the SMILES strings of the polymer repeat units for polymer representation, the inclusion of polymerization point '[*]' in the SMILES strings brings several difficulties for common cheminformatics packages to generate molecular descriptors or fingerprints, which have been extensively used in polymer informatics [25,30,50]. For cheminformatics packages like AlvaDesc [51], the SMILES strings with '[*]' cannot be processed. While some other packages such as RDKit [52] can process this type of SMILES strings for descriptor generation, not all of them are available as the symbol '[*]' is an unknown element for them to process, though RDKit can still generate molecular fingerprints for the SMILES with '[*]'. This is probably the reason why the monomers of polymers have been adopted for molecular descriptors/fingerprints generation as they are very easily processed, although it is criticized that monomers are not enough for polymer's morphological representation [25,37,53,54].

Here, in order to avoid this deficiency and use the polymer representation directly, we propose a chemical language processing model which is purely linguistic-based on the SMILES strings. The idea is to consider the polymer's repeat unit (SMILES) as sequential data at the character level. It is then processed by a polymer embedding layer and the RNN for DL model development [55–57]. RNNs have enjoyed great success in, e.g., music processing, and language translation [58,59]. In the field of cheminformatics, they have also been widely applied as deep generative models for molecular generations [29,31,41–43]. A majority of RNN generative methods have been integrated in the generative adversarial network (GAN) and variational autoencoder (VAE) for molecule generation. For example, After Yu, Lantao, et al. [60] have used the RNN variant—LSTM in GAN to generate sequences, Guimaraes, et al. [61] utilized the same strategy to generate molecules with desirable properties. And then based on which Lengeling et al. [62] present their Objective-Reinforced Generative Adversarial Network for Inverse-design Chemistry (ORGANIC)—which is able to generate novel molecules such as with melting points above 800 K. If integrated in VAE, another RNN variant—GRU has also been utilized for molecule

generation. Gø'mez-Bombarelli, et al. [63] have implemented a encoder RNN (GRU) to convert molecules into a latent space vector, and then convert it back to molecule smiles with a decoder RNN (GRU). Operations in latent space allow the decoder RNN to generate novel molecules with optimized properties. To improve the validity rate (valid decoded molecules to the total decoded molecules), Chaochao Yan, et al. [64] have built a VAE model with the bi-directional GRU and uni-directional GRU being the encoder and decoder. Their valid expression rate for the generated molecules is more than 90%. These RNN processing SMILES for molecule generations have been developed extensively, but few studies have been focused on RNN processing SMILES to predict molecule properties [33,41]. To our best knowledge, this work is the first to apply purely linguistic-based (SMILES) DL models for polymer's T_g prediction. The schematic of this model for T_g prediction is given in Figure 1, which will be introduced in detail in the later sections. The results show that this method is a good alternative to the conventional methods based on molecular descriptors or fingerprints.
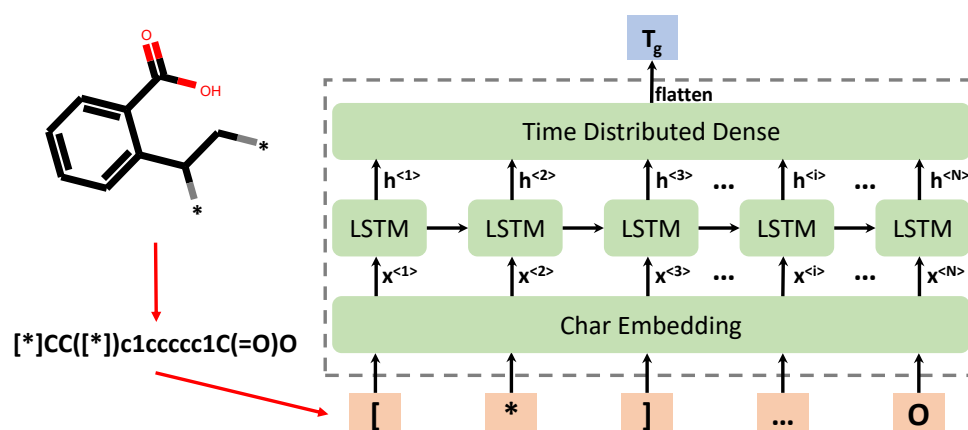

Figure 1. Schematic of the computational framework for chemical language processing model, in which a polymer's repeat unit is first represented by its canonical SMILES string, which is further separated into individual chars as inputs of the RNN model. In the RNN model, light orange color denotes the input chars of SMILES string of polymer's repeat units; green color denotes the intermediate layers, including embedding layer, LSTM layer and dense layer; the light blue color denotes the final output layer of the predicted T_g values of polymers.

The remaining of the paper is organized as follows. The computational methodology of the chemical language processing model is presented in Section 2. Specifically, the database and feature representation of polymers, the char embedding, RNN, and DL models are described in detail. The ultimate architecture of the model and its performance tests are given in Section 3. Several aspects of the chemical language processing model are further discussed in Section 4. Finally, the paper is concluded by remarks in Section 5.

2. Computational Methods

2.1. Database and Feature Representation

There are 7372 polymers in total in the current database. The respective T_g count distribution is presented in Figure 2a. As mentioned previously, the SMILES strings of polymer repeat units are employed for polymer representation. Note, however, that the general SMILES string may not be unique for molecular representation. For example, 'C1=CC=CC=C1' and 'c1ccccc1' are all valid SMILES strings of benzene. To eliminate the inconsistency in the general SMILES representation, all the SMILES strings of polymer's repeat units in the database have been processed to the corresponding canonical SMILES string using the RDKit cheminformatics package [52].
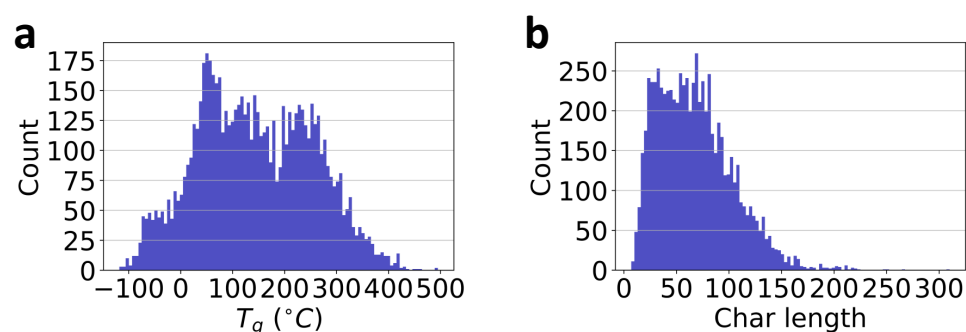
a

b

Figure 2. Database visualization. (**a**): the distribution of T_g values in the database, PolyInfo; (**b**): the distribution of the length of chars in the SMILES strings of the polymer's repeat units.

With this large database of polymers and SMILES string representation for polymer repeat units, the prediction of polymer's T_g is considered as a chemical language processing problem using the RNN. A significant advantage of this method is that no molecular descriptors or fingerprints are generated for ML model development to get around the restrictions on SMILES in descriptor generation.

In the natural language processing field, word-level or char level models can be applied as sentences are composed of words [65,66]. However, for polymer repeat units, only 'word' structure exists, i.e., SMILES strings. Thus, in this work, the char level RNN model is formulated to learn the chemical language of polymers in the SMILES notation. As shown in Figure 1, the pre-processing step is to split the SMILES string into a list of individual chars, which are then tokenized into integers and fed into the embedding layer of the DL model.

2.2. Char Embedding

Generally, in ML model development, the inputs are usually represented in digit numbers so that mathematical models can be constructed [67]. It is the same case for natural language processing. Two methods are usually used for word or char encoding in previous studies, namely one-hot encoding and categorical encoding. In this work, the latter is adopted for char encoding using the position it appears in the char lexicon. The whole list of chars contained in the polymer database is alphabetically as follows:

char lexicon = {'#', '%', '(', ')', '*', '+', '−', '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '=', 'B', 'C', 'F', 'G', 'H', 'I', 'K', 'L', 'N', 'O', 'P', 'S', 'T', 'Z', '[', ']', 'a', 'b', 'c', 'd', 'e', 'i', 'l', 'n', 'o', 'r', 's' }

In the current database, the total number of characters in the list is 45. Consequently, any character in the list can be represented by an integer number in the range of 0 to 44 following the python index rule [68]. Therefore, any SMILES string can be represented by a vector composed of the index number of its individual chars. For example, the numeric representation of polyethylene '[*]CC[*]' is $[32, 4, 33, 19, 19, 32, 4, 33]$. In our polymer database, since the length of the SMILES strings are not the same or uniformly distributed as shown in Figure 2b, to accelerate the ML model development using batch training, a constant length has to be prescribed for the inputs. Another reason is to shorten the sequence length for the next LSTM layer to reduce training difficulties, as longer sequences may result in gradient vanishing or exploding problems during back-propagation. As a result, polymers with longer SMILES strings than the critical length will be truncated; while polymers with short strings will be padded with zeros in the trailing locations. In this database, over 82.1% polymers have shorter SMILES strings than 100; while about 91.2% polymers have shorter SMILES strings than 120. Thus, this number is considered as a hyperparameter in the ML model development to meet the trade-off between accuracy and computational efficiency.

Despite simple and clear, this encoding algorithm may not well represent similarities between words or chars. Therefore, this feature representation alone is not enough for meaningful feature extraction and for ML model development with good performance. In

previous work [69], the authors tested DNN model performance just on integer-encoded vector by ASCII code for SMILES, the accuracy was very poor (accuracy score was about 0.53). It has been shown using word embedding can improve the model performances in natural language processing [70,71]. The objective of word/char embedding is to transform the one-hot or categorical encoding of words/chars into a new shorter yet dense vector with useful language meanings, which is learned by the DL model during model training. Hence, an embedding layer is adopted as the first layer of the chemical language processing model following the input layer, as shown in Figure 1. The purpose is that by applying an embedding layer, meaningful chemical information can be learned and passed to the recurrent neural network so that good performance can be achieved.

2.3. Recurrent Neural Network

The key idea of RNN is to use hidden variables to pass information from early states to later states for sequential data that has temporal dependencies [72]. RNNs have been the driving force in natural language processing, such as language translation and speech recognition. The simplest RNN unit is the so-called vanilla RNN, which suffers from gradient exploding or gradient vanishing problems in practice [72]. Therefore, more advanced units have been developed to build robust models, such as the Long Short-Term Memory (LSTM) unit [73] and the Gated Recurrent Unit (GRU) [74], both of which have been the golden standards of RNNs. The essential improvement is adding a cell state and gates to control the information flow in/out of the unit, in addition to the hidden state variables. In this work, the LSTM unit is employed in the RNN model. An illustrative figure for the LSTM unit is shown in Figure 3.
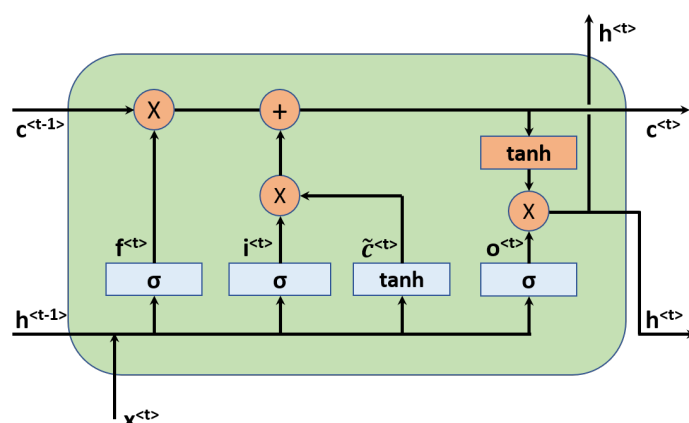

Figure 3. The LSTM unit used in the recurrent neural network.

There are three gates in the LSTM unit, namely, the forget gate, input/update gate, and the output gate. Let the input be denoted by $x^{<t>}$ at the time step t, the hidden state and cell state variables be expressed by $h^{<t>}$ and $c^{<t>}$, respectively. The computational procedure in the LSTM unit is then:

$$
\begin{aligned}
f^{<t>} &= \sigma(W_{fh}h^{<t-1>} + W_{fx}x^{<t>} + b_f) \\
i^{<t>} &= \sigma(W_{ih}h^{<t-1>} + W_{ix}x^{<t>} + b_i) \\
o^{<t>} &= \sigma(W_{oh}h^{<t-1>} + W_{ox}x^{<t>} + b_o) \\
\tilde{c}^{<t>} &= \tanh(W_{ch}h^{<t-1>} + W_{cx}x^{<t>} + b_c) \\
c^{<t>} &= f^{<t>} \times c^{<t-1>} + i^{<t>} * \tilde{c}^{<t>} \\
h^{<t>} &= o^{<t>} \times \tanh(c^{<t>})
\end{aligned}
\tag{1}
$$

where $f^{<t>}$, $i^{<t>}$, and $o^{<t>}$ are respectively the activated vectors for forget, update, and output gate. $\tilde{c}^{<t>}$ and $c^{<t>}$ are the input activated and the updated cell state, respectively.

W_{fh}, W_{fx}, W_{ih}, W_{ix}, W_{oh}, W_{ox}, W_{ch}, W_{cx}, and b_f, b_i, b_o, b_c are trainable weights and biases in the LSTM unit. The symbol '*' denotes element-wise multiplication. σ is the nonlinear activation function such as sigmoid function, and tanh is the hyperbolic activation function.

Note that in addition to the unidirectional LSTM layer, the bidirectional LSTM layer has also been widely applied so that information can be passed from both early chars and later chars. Thus, the unidirectional and bidirectional LSTM networks are also considered for hyperparameter tuning.

2.4. DL Model Development

In this work, the DL model of chemical language processing is developed under the Tensorflow platform [75] mainly using the Keras package [76] to realize the aforementioned layers. To train and validate the chemical language processing model, the total database is split into a training dataset with 90% of the data and a test dataset with the remaining data because of a large database at hand. In the training process, the training dataset is further split into training and validation datasets by an 8:2 ratio to monitor model performance during training. The DL model is first trained on the training dataset and then evaluated on the unseen test dataset. Mathematically, the DL model seeks to find a prediction function $f : \mathbb{R}^d \mapsto \mathbb{R}$, which maps the inputs of chars in d dimensions to the T_g value. The training process is equivalent to finding the optimal weights and biases by solving an optimization problem:

$$\underset{w,b}{\arg\min} \quad \mathcal{L}(w,b) \tag{2}$$

where w and b are the weights and biases in the DL model, which keep updating by gradient descent scheme [77]. $\mathcal{L}(w,b)$ is the loss function, which is defined as:

$$\mathcal{L}(w,b) = \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2 \tag{3}$$

and the evaluation metric of the DL model on the test dataset is

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{4}$$

where m and n are the number of polymer samples in the training and test dataset, respectively. y_i and $\hat{y}_i$ denote the real and predicted values of the T_g of the i-th sample, respectively.

To develop an ML model with good performance, the grid search approach is usually adopted to tune the hyperparameters that lead to a relatively better model. The total hyperparameters considered in this work include:

1. The maximum input length of the SMILES string (100 or 120);
2. The length of the embedded vector (the output of the embedding layer);
3. The type of LSTM layer (unidirectional or bidirectional);
4. The number of LSTM layers;
5. The number of hidden neurons for each LSTM units;
6. The type of intermediate layers (dense layer or time distributed layer), as shown in Figure 4.

In the grid search of the optimal hyperparameters, the Adam optimization scheme [78] is adopted to minimize the loss function for weights and bias updates. In each case, the model is first trained on the training dataset, and then the prediction performance is evaluated on the test dataset using the mean absolute error (MAE) metric, which provides guidance on the selection of the optimal hyperparameters. The early stopping and checkpoints are employed to automatically cease training once comparable model performances are observed on the training and validation datasets.
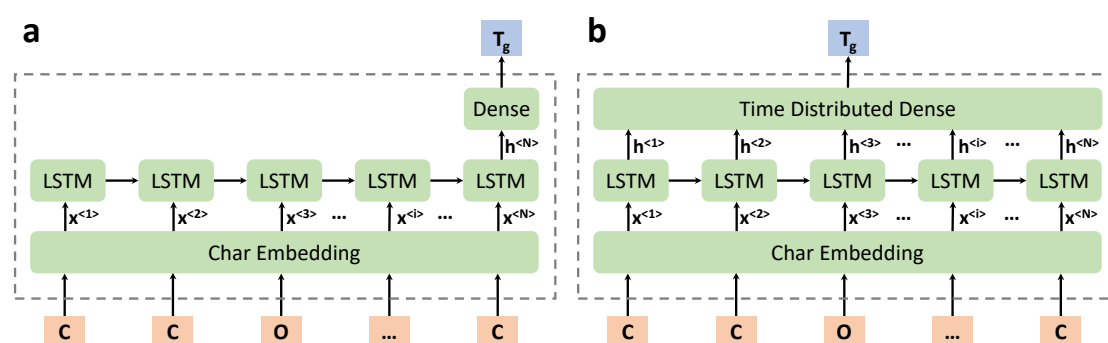
Figure 4. Two different types of DL model architectures. (**a**): Dense layer as an intermediate layer after the last LSTM unit of the previous LSTM layer; (**b**): Time Distributed Dense layer as an intermediate layer (flattened representation).

3. Results

3.1. The architecture of the Chemical Language Processing Model

A series of chemical language processing models with various hyperparameters are developed according to the setup described in Section 2.4. Readers are referred to the Supporting Information for more details. It is observed that the DL model is relatively stable under different hyperparameters, with the MAE metric on the test dataset being in the range of 30∼34 °C. It is also observed that using the Time Distributed Dense layer (Figure 4b) may result in better model performance, which passes information out at each time step. While there is no obvious performance difference in DL models using unidirectional or bidirectional LSTM layers. The architecture of the optimal chemical language processing model is the one shown in Figure 4b.

Specifically, the char embedding layer receives an encoded char vector with a length of 120 and outputs an embedded vector of a length of 15 at each time step. In the next, two bidirectional LSTM layers are implemented with 60 hidden neurons for each layer. A Time Distributed Dense layer with 30 neurons follows the RNN (LSTM) layers subsequently. The final layer is a dense layer with only one neuron which denotes the predicted glass transition temperature T_g. All previous layers use the default activation functions while the final dense layer uses the linear activation function. Unless otherwise stated explicitly, the other parameters are following the default settings in the Keras package.

The learning curve in the training process is shown in Figure 5a. As can be seen from this curve that comparable performances have been achieved on the training and validation dataset. It should be noted that since a patience length of 10 epochs during training is applied, the best model is saved due to early stopping rather than the model trained at the final epoch.
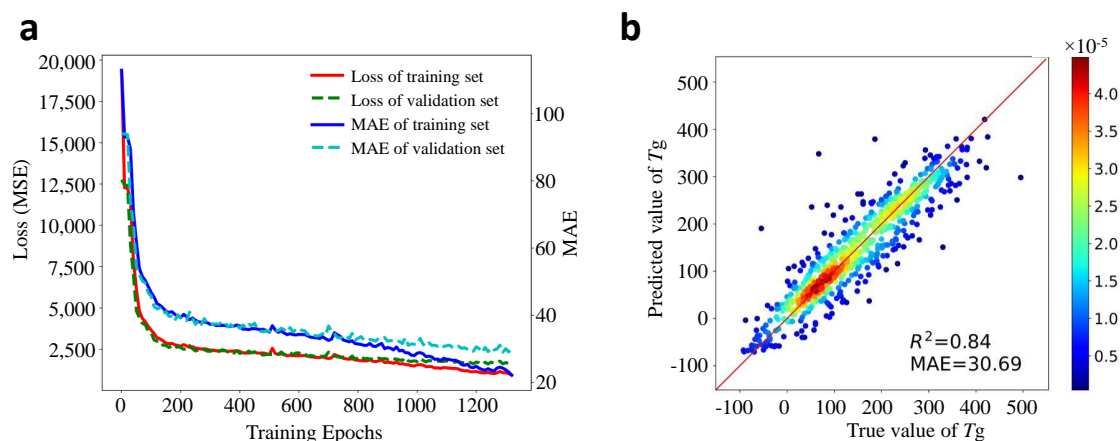

Figure 5. DL model development and evaluation. (**a**): the learning curves of the loss and the MAE with training epochs on the training and validation dataset; (**b**): the performance evaluation of the model on the unseen test dataset (color denotes the density of points).

3.2. Predictions of the Chemical Language Processing Model on Unseen Data

To further validate the trained chemical language processing model, we apply it to predict T_g values of the test dataset. Note that the test dataset is unseen during the training of the DL model. Therefore, the predictability of the DL model can be directly evaluated on this additional dataset, which has 724 polymer samples in total.

After the DL model is well-trained, new predictions can be made easily on the test dataset. The correlation coefficient R^2 score and the MAE can then be calculated based on the predicted and true values of T_g, which is plotted in Figure 5b. One can see that the majority of the scatter points locates in the unity red line, indicating the predicted T_g values are close to their true values. Quantitatively, the developed DL model gives a correlation score $R^2 = 0.84$ and MAE = 30.69 °C. This performance is reasonably well and comparable with many other ML models for T_g prediction in terms of MAE values or R^2 score [24,37–39,79], which confirms the effectiveness of the chemical language processing model. Note that in most previous works, the polymer samples were not large and only certain types of polymers were studied [36], the MAE and R^2 score may be higher. While in this work, the data size is very large and the types of polymers in the database are very general.

3.3. Application of the Chemical Language Processing Model for High-Throughput Screening

To demonstrate the capability of our chemical language processing model, another unlabeled polymer dataset of 5686 samples without reported T_g values are considered for a high-throughput screening task. This dataset collected from earlier work [36] is also from the PolyInfo database [48]. Thus, these two databases are considered similar. It can also be seen from the char length distribution shown in Figure 6a, as compared to the labeled database given in Figure 2b.

To make T_g predictions, the polymer's repeat units in the unlabeled database are first converted into the integer-encoded vector form and then feed into the developed chemical language processing model. The glass transition temperature T_g for those unlabeled polymers can be quickly predicted. Figure 6b presents the distribution of the predicted glass transition temperatures T_g.

For high-throughput screening tasks, the candidates with extreme properties are usually desired and of great value in material discovery [80,81]. As an example, twelve candidates in this unlabeled database with T_g larger than 400 °C are quickly identified, as shown in Figure 6c, although their T_g values have not been reported before. Particularly, we find the chemical structures of these identified polymers share similar features as other high-temperature polymers, such as polyaryletherketone and polyimide. For instance, saturated 4,5 member rings, bridged rings, benzene rings, oxolane groups, amine groups, and halogens had a higher occurrence rate for polymers with high T_g [81–83]. For preliminary validation of ML predictions, we have performed all-atom molecular dynamics (MD) simulations on these polymers, with simulation protocols and detailed results given in the Supporting Information. Overall, the T_g values predicted from molecular dynamics simulations are in good agreement with ML predictions within the range of uncertainty. It indicates that the proposed model can be employed for high-throughput screening tasks if trained well. Besides, the model's prediction ability is evaluated on anther dataset of 32 conjugated polymers with experimentally reported T_g values [84]. A reasonable prediction is demonstrated and can be found in the Supporting Informations. However, note that these examples are mainly adopted for demonstration purposes of the chemical language processing model. If the unlabeled database is significantly different from the training database with reported T_g values, the DL model would do an extrapolation rather than interpolation, which would lead to inaccurate predicted T_g.
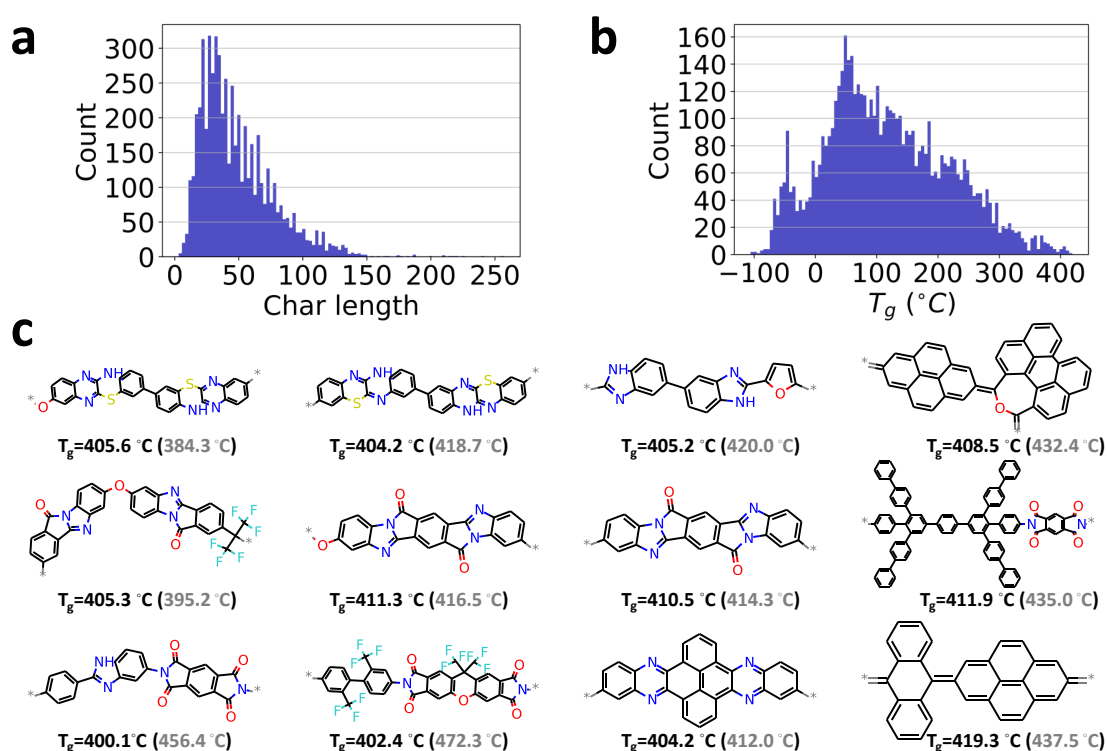
Figure 6. Data visualization for high-throughput screening of high-temperature polymers. (**a**): the distribution of the length of chars in the SMILES strings of polymer's repeat units; (**b**): the distribution of predicted T_g values in the database; (**c**): the 12 candidates with T_g larger than 400 °C in the screening of the unlabeled database. '*' indicates the polymerization point in the repeat unit of polymer. Values in the parentheses by gray color are respective T_g obtained by MD simulations.

4. Discussion

Here, we formulate the forward prediction of polymer's T_g as a chemical language processing problem, leveraging a large polymer database PolyInfo. The utilization of SMILES strings for polymer's repeat unit as feature representation is made to develop DL models. To encode the SMILES strings for DL model development, a lexicon composed of individual characters following alphabetic order is applied. Since feature representation is of great importance for ML models [30], alternative forms of polymer lexicon can be developed to build superior chemical language processing models. For example, an element lexicon can be developed based on the atomic element, e.g., using 'Si' as a lexicon element for silicon instead of 'S' and 'i'.

Additionally, one potential way to improve model performance is to incorporate more chemical domain knowledge into the model. For instance, adding in molecular weight, topological information of polymers, and processing conditions as additional inputs so that the model can reasonably predict T_g with better accuracy. This can be realized by, for example, taking advantage of the hidden vector of the RNN. The additional information can be used to initialize the hidden vector. Alternatively, the information can be added by concatenating to the outputs of RNNs. Moreover, focusing on certain types of polymers, e.g., polyolefin, or polyesters, may also potentially improve the model performances. For example, Pugar et. al. considered polyurethane elastomers and applied ML to extract important physicochemical properties governing T_g [85]. Leveraging these descriptors, such as electrostatic, hydrogen bonding, and microstructures of the hard segments, in the model can improve ML model performances. Furthermore, the sampling method of the training dataset can also impact the model performances, especially for studies with a small database [86].

There are several advantages of the feature representation adopted in this work. The use of polymer repeat units is more reasonable than that of monomers as the former is a building block of the corresponding polymers, though the use of polymer monomers

has been widely adopted in polymer informatics [39,87,88]. This is probably due to the requirements of cheminformatics packages on the SMILES strings that can be processed. Polymer's monomers can be easily processed to generate molecular descriptors or finger-prints to be used as inputs for ML model development, while polymer's repeat units with polymerization point '[*]' may not be feasibly processed in many packages. Besides, there is no additional pre-processing needed before ML model development due to the pure SMILES string used as inputs, in contrast to the use of molecular descriptors or fingerprints. Thus, the formulation of polymer's T_g prediction as a chemical language processing might be more beneficial and efficient. This representation will also benefit the development of generative ML models for the inverse molecular design of polymers.

While the polymers in this study are homopolymers, the framework is general and can be extended to study polymer blends of different typologies. The first step is to prepare the inputs which include the SMILES string of composing polymers and the ratio of them. A model is feasible to build from the perspective of ML model development, but the performance remains to be seen depending on the specific system of interest. For example, when polystyrene under cyclic topological constraint is compared with its linear compartment, a reduced hydrodynamic volume has been reported, leading to higher T_g. Although our RNN model is purely trained on linear polymers, its prediction ability on cyclic architecture is also well demonstrated, as shown in Figure 7. The prediction trend matches well with experiments observation that The cyclic architecture has higher T_g compared with the linear analogue [89]. A positive correlation of RNN T_g prediction to the molecular weight is well recognized too, especially on the linear architecture which is used for our model training.
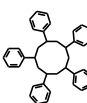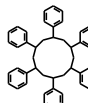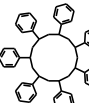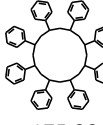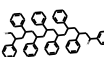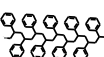
Figure 7. RNN model predictions on various polystyrene architectures. The cyclic architecture and linear architecture of polystyrene being evaluated by the obtained RNN model are accompanied by the T_g prediction (in Celsius). Experimental T_g is taken from [89], with T_g values ranging from −40~100 °C and 65~110 °C form linear and cyclic polystyrene polymers, respectively, depending on the molecular weight.

5. Conclusions

In summary, we proposed a chemical language processing model for predictions of polymer's T_g. The SMILES notation of polymer's repeat unit is adopted as feature representation, which is purely linguistic-based. There are no additional computations needed for pre-processing, in contrast to other conventional polymer informatics models. The key feature of our model is the usage of char embedding and RNN to process the char-based inputs of polymers. Reasonable predictions on polymer's T_g can be achieved using this model. Besides, a high-throughput screening task has been performed on an

unlabeled polymer database to identify promising candidates with high T_g values that can be used in extreme environments. It suggests that the chemical language processing model may be used as an effective approach to developing predictive ML models for other properties of polymers, such as melting temperature, electronic bandgap, dielectric constant, refractive index, and many others.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional neural network
DL	Deep learning
DNN	Deep neural network
LSTM	Long short-term memory
MD	Molecular dynamics
ML	Machine learning
QSPR	Quantitative structure-property relationships
RNN	Recurrent neural network
SMILES	Simplified molecular-input line-entry system

References

1. Weyland, H.; Hoftyzer, P.; Van Krevelen, D. Prediction of the glass transition temperature of polymers. *Polymer* **1970**, *11*, 79–87. [CrossRef]
2. Chow, T. Molecular interpretation of the glass transition temperature of polymer-diluent systems. *Macromolecules* **1980**, *13*, 362–364. [CrossRef]
3. DiBenedetto, A. Prediction of the glass transition temperature of polymers: A model based on the principle of corresponding states. *J. Polym. Sci. Part B Polym. Phys.* **1987**, *25*, 1949–1969. [CrossRef]
4. Dudowicz, J.; Freed, K.F.; Douglas, J.F. The glass transition temperature of polymer melts. *J. Phys. Chem. B* **2005**, *109*, 21285–21292. [CrossRef]

5. Jha, A.; Chandrasekaran, A.; Kim, C.; Ramprasad, R. Impact of dataset uncertainties on machine learning model predictions: The example of polymer glass transition temperatures. *Modell. Simul. Mater. Sci. Eng.* **2019**, *27*, 024002. [CrossRef]
6. Zhang, Y.; Xu, X. Machine learning glass transition temperature of polymers. *Heliyon* **2020**, *6*, e05055. [CrossRef] [PubMed]
7. Mark, J.; Ngai, K.; Graessley, W.; Mandelkern, L.; Samulski, E.; Wignall, G.; Koenig, J. *Physical Properties of Polymers*; Cambridge University Press: Cambridge, UK, 2004.
8. Stutz, H.; Illers, K.H.; Mertes, J. A generalized theory for the glass transition temperature of crosslinked and uncrosslinked polymers. *J. Polym. Sci. Part B Polym. Phys.* **1990**, *28*, 1483–1498. [CrossRef]
9. Gedde, U. *Polymer Physics*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 1995.
10. Hiemenz, P.C.; Lodge, T.P. *Polymer Chemistry*; CRC Press: Boca Raton, FL, USA, 2007.
11. Rigby, D.; Roe, R.J. Molecular dynamics simulation of polymer liquid and glass. I. Glass transition. *J. Chem. Phys.* **1987**, *87*, 7285–7292. [CrossRef]
12. Koehler, M.; Hopfinger, A. Molecular modelling of polymers: 5. Inclusion of intermolecular energetics in estimating glass and crystal-melt transition temperatures. *Polymer* **1989**, *30*, 116–126. [CrossRef]
13. Morita, H.; Tanaka, K.; Kajiyama, T.; Nishi, T.; Doi, M. Study of the glass transition temperature of polymer surface by coarse-grained molecular dynamics simulation. *Macromolecules* **2006**, *39*, 6233–6237. [CrossRef]
14. Xu, Q.; Jiang, J. Molecular simulations of liquid separations in polymer membranes. *Curr. Opin. Chem. Eng.* **2020**, *28*, 66–74. [CrossRef]
15. Yamada, H.; Liu, C.; Wu, S.; Koyama, Y.; Ju, S.; Shiomi, J.; Morikawa, J.; Yoshida, R. Predicting materials properties with little data using shotgun transfer learning. *ACS Cent. Sci.* **2019**, *5*, 1717–1730. [CrossRef]
16. Chandrasekaran, A.; Kim, C.; Ramprasad, R. Polymer Genome: A Polymer Informatics Platform to Accelerate Polymer Discovery. In *Machine Learning Meets Quantum Physics*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 397–412.
17. Batra, R.; Song, L.; Ramprasad, R. Emerging materials intelligence ecosystems propelled by machine learning. *Nat. Rev. Mater.* **2020**, 1–24. [CrossRef]
18. Zhang, Y.; Xu, X. Machine learning glass transition temperature of polyacrylamides using quantum chemical descriptors. *Polym. Chem.* **2021**, *12*, 843–851. [CrossRef]
19. Katritzky, A.R.; Rachwal, P.; Law, K.W.; Karelson, M.; Lobanov, V.S. Prediction of polymer glass transition temperatures using a general quantitative structure-property relationship treatment. *J. Chem. Inform. Comput. Sci.* **1996**, *36*, 879–884. [CrossRef]
20. Liu, H.; Uhlherr, A.; Bannister, M.K. Quantitative structure–property relationships for composites: prediction of glass transition temperatures for epoxy resins. *Polymer* **2004**, *45*, 2051–2060. [CrossRef]
21. Zhang, Y.; Xu, X. Machine learning glass transition temperature of styrenic random copolymers. *J. Mol. Graphics Modell.* **2021**, *103*, 107796. [CrossRef]
22. Adams, N. Polymer informatics. In *Polymer Libraries*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 107–149.
23. Audus, D.J.; de Pablo, J.J. Polymer informatics: Opportunities and challenges. *ACS Macro Lett.* **2017**, *6*, 1078–1082. [CrossRef] [PubMed]
24. Kim, C.; Chandrasekaran, A.; Huan, T.D.; Das, D.; Ramprasad, R. Polymer genome: A data-powered polymer informatics platform for property predictions. *J. Phys. Chem. C* **2018**, *122*, 17575–17585. [CrossRef]
25. Chen, L.; Pilania, G.; Batra, R.; Huan, T.D.; Kim, C.; Kuenneth, C.; Ramprasad, R. Polymer Informatics: Current Status and Critical Next Steps. *arXiv* **2020**, arXiv:2011.00508.
26. Gómez-Bombarelli, R.; Aguilera-Iparraguirre, J.; Hirzel, T.D.; Duvenaud, D.; Maclaurin, D.; Blood-Forsythe, M.A.; Chae, H.S.; Einzinger, M.; Ha, D.G.; Wu, T.; et al. Design of efficient molecular organic light-emitting diodes by a high-throughput virtual screening and experimental approach. *Nat. Mater.* **2016**, *15*, 1120–1127. [CrossRef]
27. Wu, S.; Kondo, Y.; Kakimoto, M.A.; Yang, B.; Yamada, H.; Kuwajima, I.; Lambard, G.; Hongo, K.; Xu, Y.; Shiomi, J.; et al. Machine-learning-assisted discovery of polymers with high thermal conductivity using a molecular design algorithm. *NPJ Comput. Mater.* **2019**, *5*, 1–11. [CrossRef]
28. Jennings, P.C.; Lysgaard, S.; Hummelshøj, J.S.; Vegge, T.; Bligaard, T. Genetic algorithms for computational materials discovery accelerated by machine learning. *NPJ Comput. Mater.* **2019**, *5*, 1–6. [CrossRef]
29. Arús-Pous, J.; Johansson, S.V.; Prykhodko, O.; Bjerrum, E.J.; Tyrchan, C.; Reymond, J.L.; Chen, H.; Engkvist, O. Randomized SMILES strings improve the quality of molecular generative models. *J. Cheminf.* **2019**, *11*, 1–13. [CrossRef] [PubMed]
30. Chen, G.; Shen, Z.; Iyer, A.; Ghumman, U.F.; Tang, S.; Bi, J.; Chen, W.; Li, Y. Machine-Learning-Assisted De Novo Design of Organic Molecules and Polymers: Opportunities and Challenges. *Polymers* **2020**, *12*, 163. [CrossRef] [PubMed]
31. Grisoni, F.; Moret, M.; Lingwood, R.; Schneider, G. Bidirectional molecule generation with recurrent neural networks. *J. Chem. Inf. Model.* **2020**, *60*, 1175–1183. [CrossRef] [PubMed]
32. Lavecchia, A. Machine-learning approaches in drug discovery: Methods and applications. *Drug Discov. Today* **2015**, *20*, 318–331. [CrossRef]
33. Popova, M.; Isayev, O.; Tropsha, A. Deep reinforcement learning for de novo drug design. *Sci. Adv.* **2018**, *4*, eaap7885. [CrossRef]
34. Chen, H.; Engkvist, O.; Wang, Y.; Olivecrona, M.; Blaschke, T. The rise of deep learning in drug discovery. *Drug Discov. Today* **2018**, *23*, 1241–1250. [CrossRef]
35. Vamathevan, J.; Clark, D.; Czodrowski, P.; Dunham, I.; Ferran, E.; Lee, G.; Li, B.; Madabhushi, A.; Shah, P.; Spitzer, M.; et al. Applications of machine learning in drug discovery and development. *Nat. Rev. Drug Discovery* **2019**, *18*, 463–477. [CrossRef]

36. Tao, L.; Chen, G.; Li, Y. Machine Learning Discovery of High-Temperature Polymers. *Patterns* **2021**, in revision. [CrossRef]
37. Ma, R.; Liu, Z.; Zhang, Q.; Liu, Z.; Luo, T. Evaluating Polymer Representations via Quantifying Structure–Property Relationships. *J. Chem. Inf. Model.* **2019**, *59*, 3110–3119. [CrossRef]
38. Miccio, L.A.; Schwartz, G.A. Localizing and quantifying the intra-monomer contributions to the glass transition temperature using artificial neural networks. *Polymer* **2020**, *203*, 122786. [CrossRef]
39. Miccio, L.A.; Schwartz, G.A. From chemical structure to quantitative polymer properties prediction through convolutional neural networks. *Polymer* **2020**, *193*, 122341. [CrossRef]
40. Nazarova, A.L.; Yang, L.; Liu, K.; Mishra, A.; Kalia, R.K.; Nomura, K.i.; Nakano, A.; Vashishta, P.; Rajak, P. Dielectric Polymer Property Prediction Using Recurrent Neural Networks with Optimizations. *J. Chem. Inf. Model.* **2021**, *61*, 2175–2186. [CrossRef] [PubMed]
41. Segler, M.H.; Kogej, T.; Tyrchan, C.; Waller, M.P. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Cent. Sci.* **2018**, *4*, 120–131. [CrossRef] [PubMed]
42. Gao, K.; Nguyen, D.D.; Tu, M.; Wei, G.W. Generative network complex for the automated generation of drug-like molecules. *J. Chem. Inf. Model.* **2020**, *60*, 5682–5698. [CrossRef] [PubMed]
43. Amabilino, S.; Pogány, P.; Pickett, S.D.; Green, D.V. Guidelines for recurrent neural network transfer learning-based molecular generation of focused libraries. *J. Chem. Inf. Model.* **2020**, *60*, 5699–5713. [CrossRef]
44. Ma, R.; Luo, T. PI1M: A Benchmark Database for Polymer Informatics. *J. Chem. Inf. Model.* **2020**, *60*, 4684–4690. [CrossRef]
45. Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inform. Comput. Sci.* **1988**, *28*, 31–36. [CrossRef]
46. Weininger, D.; Weininger, A.; Weininger, J.L. SMILES. 2. Algorithm for generation of unique SMILES notation. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 97–101. [CrossRef]
47. Weininger, D. SMILES. 3. DEPICT. Graphical depiction of chemical structures. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 237–243. [CrossRef]
48. Otsuka, S.; Kuwajima, I.; Hosoya, J.; Xu, Y.; Yamazaki, M. PoLyInfo: Polymer database for polymeric materials design. In Proceedings of the 2011 International Conference on Emerging Intelligent Data and Web Technologies, Tirana, Albania, 7–9 September 2011; pp. 22–29.
49. Tanifuji, M.; Matsuda, A.; Yoshikawa, H. Materials Data Platform-a FAIR System for Data-Driven Materials Science. In Proceedings of the 2019 8th International Congress on Advanced Applied Informatics (IIAI-AAI), Toyama, Japan, 7–11 July 2019; pp. 1021–1022.
50. Chen, G.; Shen, Z.; Li, Y. A machine-learning-assisted study of the permeability of small drug-like molecules across lipid membranes. *Phys. Chem. Chem. Phys.* **2020**, *22*, 19687–19696. [CrossRef] [PubMed]
51. Mauri, A. alvaDesc: A tool to calculate and analyze molecular descriptors and fingerprints. In *Ecotoxicological QSARs*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 801–820.
52. Landrum, G. RDKit: Open-Source Cheminformatics. 2006. Available online: https://www.rdkit.org/ (accessed on 11 February 2021).
53. Wu, K.; Sukumar, N.; Lanzillo, N.; Wang, C.; "Rampi" Ramprasad, R.; Ma, R.; Baldwin, A.; Sotzing, G.; Breneman, C. Prediction of polymer properties using infinite chain descriptors (ICD) and machine learning: Toward optimized dielectric polymeric materials. *J. Polym. Sci. Part B Polym. Phys.* **2016**, *54*, 2082–2091. [CrossRef]
54. Wu, S.; Yamada, H.; Hayashi, Y.; Zamengo, M.; Yoshida, R. Potentials and challenges of polymer informatics: exploiting machine learning for polymer design. *arXiv* **2020**, arXiv:2010.07683.
55. Zaremba, W.; Sutskever, I.; Vinyals, O. Recurrent neural network regularization. *arXiv* **2014**, arXiv:1409.2329.
56. Mikolov, T.; Kombrink, S.; Burget, L.; Černocký, J.; Khudanpur, S. Extensions of recurrent neural network language model. In Proceedings of the 2011 IEEE international conference on acoustics, speech and signal processing (ICASSP), Prague, Czech Republic, 22–27 May 2011; pp. 5528–5531.
57. Jo, J.; Kwak, B.; Choi, H.S.; Yoon, S. The message passing neural networks for chemical property prediction on SMILES. *Methods* **2020**, *179*, 65–72. [CrossRef]
58. Boulanger-Lewandowski, N.; Bengio, Y.; Vincent, P. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. *arXiv* **2012**, arXiv:1206.6392.
59. Auli, M.; Galley, M.; Quirk, C.; Zweig, G. Joint Language and Translation Modeling with Recurrent Neural Networks. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing; Association for Computational Linguistics, Seattle, WA, USA, 18–21 October 2013; pp. 1044–1054.
60. Yu, L.; Zhang, W.; Wang, J.; Yu, Y. Seqgan: Sequence generative adversarial nets with policy gradient. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31.
61. Guimaraes, G.L.; Sanchez-Lengeling, B.; Outeiral, C.; Farias, P.L.C.; Aspuru-Guzik, A. Objective-reinforced generative adversarial networks (ORGAN) for sequence generation models. *arXiv* **2017**, arXiv:1705.10843.
62. Sanchez-Lengeling, B.; Outeiral, C.; Guimaraes, G.L.; Aspuru-Guzik, A. Optimizing distributions over molecular space. An objective-reinforced generative adversarial network for inverse-design chemistry (ORGANIC). *ChemRxiv* **2017**, *2017*. [CrossRef]

63. Gómez-Bombarelli, R.; Wei, J.N.; Duvenaud, D.; Hernández-Lobato, J.M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T.D.; Adams, R.P.; Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Cent. Sci.* **2018**, *4*, 268–276. [CrossRef]

64. Yan, C.; Wang, S.; Yang, J.; Xu, T.; Huang, J. Re-balancing variational autoencoder loss for molecule sequence generation. In Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, Virtual Event USA, 21–24 September 2020; pp. 1–7.

65. Kim, Y.; Jernite, Y.; Sontag, D.; Rush, A.M. Character-aware neural language models. *arXiv* **2015**, arXiv:1508.06615.

66. Pham, T.H.; Le-Hong, P. End-to-end recurrent neural network models for vietnamese named entity recognition: Word-level vs. character-level. In *International Conference of the Pacific Association for Computational Linguistics*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 219–232.

67. Shalev-Shwartz, S.; Ben-David, S. *Understanding Machine Learning: From Theory to Algorithms*; Cambridge University Press: Cambridge, UK, 2014.

68. Van Rossum, G.; Drake Jr, F.L. *Python Tutorial*; Centrum voor Wiskunde en Informatica: Amsterdam, The Netherlands, 1995; Volume 620.

69. Sun, W.; Zheng, Y.; Yang, K.; Zhang, Q.; Shah, A.A.; Wu, Z.; Sun, Y.; Feng, L.; Chen, D.; Xiao, Z.; et al. Machine learning–assisted molecular design and efficiency prediction for high-performance organic photovoltaic materials. *Sci. Adv.* **2019**, *5*, eaay4275. [CrossRef] [PubMed]

70. Zamani, H.; Croft, W.B. Relevance-based word embedding. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Tokyo, Japan, 7–11 August 2017; pp. 505–514.

71. Ruder, S.; Vulić, I.; Søgaard, A. A survey of cross-lingual word embedding models. *J. Artif. Intell. Res.* **2019**, *65*, 569–631. [CrossRef]

72. Goodfellow, I.; Bengio, Y.; Courville, A.; Bengio, Y. *Deep Learning*; MIT Press: Cambridge, UK, 2016; Volume 1.

73. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]

74. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.

75. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. Tensorflow: A system for large-scale machine learning. In Proceedings of the 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16), Savannah, GA, USA, 2–4 November 2016; pp. 265–283.

76. Chollet, F. Keras. 2015. Available online: https://keras.io (accessed on 11 February 2021).

77. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2016**, arXiv:1609.04747.

78. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.

79. Kim, C.; Chandrasekaran, A.; Jha, A.; Ramprasad, R. Active-learning and materials design: The example of high glass transition temperature polymers. *MRS Commun.* **2019**, *9*, 860–866. [CrossRef]

80. Johnson, L.M.; Ledet, E.; Huffman, N.D.; Swarner, S.L.; Shepherd, S.D.; Durham, P.G.; Rothrock, G.D. Controlled degradation of disulfide-based epoxy thermosets for extreme environments. *Polymer* **2015**, *64*, 84–92. [CrossRef]

81. Batra, R.; Dai, H.; Huan, T.D.; Chen, L.; Kim, C.; Gutekunst, W.R.; Song, L.; Ramprasad, R. Polymers for Extreme Conditions Designed Using Syntax-Directed Variational Autoencoders. *Chem. Mater.* **2020**, *32*, 10489–10500. [CrossRef]

82. Mittal, V. *High Performance Polymers and Engineering Plastics*; John Wiley & Sons: Hoboken, NJ, USA, 2011.

83. Fink, J.K. *High Performance Polymers*; William Andrew: Norwich, NY, USA, 2014.

84. Xie, R.; Weisen, A.R.; Lee, Y.; Aplan, M.A.; Fenton, A.M.; Masucci, A.E.; Kempe, F.; Sommer, M.; Pester, C.W.; Colby, R.H.; et al. Glass transition temperature from the chemical structure of conjugated polymers. *Nat. Commun.* **2020**, *11*, 1–8. [CrossRef] [PubMed]

85. Pugar, J.A.; Childs, C.M.; Huang, C.; Haider, K.W.; Washburn, N.R. Elucidating the Physicochemical Basis of the Glass Transition Temperature in Linear Polyurethane Elastomers with Machine Learning. *J. Phys. Chem. B* **2020**, *124*, 9722–9733. [CrossRef] [PubMed]

86. Wen, C.; Liu, B.; Wolfgang, J.; Long, T.E.; Odle, R.; Cheng, S. Determination of glass transition temperature of polyimides from atomistic molecular dynamics simulations and machine-learning algorithms. *J. Polym. Sci.* **2020**, *58*, 1521–1534. [CrossRef]

87. Mattioni, B.E.; Jurs, P.C. Prediction of glass transition temperatures from monomer and repeat unit structure using computational neural networks. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 232–240. [CrossRef]

88. Pilania, G.; Iverson, C.N.; Lookman, T.; Marrone, B.L. Machine-Learning-Based Predictive Modeling of Glass Transition Temperatures: A Case of Polyhydroxyalkanoate Homopolymers and Copolymers. *J. Chem. Inf. Model.* **2019**, *59*, 5013–5025. [CrossRef] [PubMed]

89. Haque, F.M.; Grayson, S.M. The synthesis, properties and potential applications of cyclic polymers. *Nat. Chem.* **2020**, *12*, 433–444. [CrossRef]