CS230

Predicting Ionic Liquid Materials Properties from Chemical Structure

Tyler Quill, Shayta Roy and Yaakov Tuchman
Department of Materials Science & Engineering
Stanford University
{tquill,sroy11,ytuchman}@stanford.edu

Abstract

Ionic liquids, a class of room-temperature molten salts, are of interest in a variety of materials applications. Accurate prediction of their physiochemical properties enables high throughput screening and acceleration of their practical application in tangible technologies. In this work, a published neural-fingerprinting algorithm was adapted to predict melting point and viscosity of ionic liquids based the component anion and cation chemical structures. Both algorithms share a base message passing neural network and successive dense layers to encode and combine information from each ion. Outputs are then passed through property-specific network heads for final prediction. Comparing the true and predicted values, the final viscosity model resulted in an R^2_{dev} = 0.89 and the final melting point model resulted in an R^2_{dev} = 0.64. Transfer learning was also attempted between the viscosity and melting point models, resulting in lower absolute performance but also lower apparent variance.

1 Introduction

Ionic liquids are molecules with a broad range of potential applications ranging from catalysis to energy storage [1]. There are over 1 million ionic liquids which are synthetically feasible, however only a small fraction have been synthesized [2]. Thus, there is interest in predicting the properties of ionic liquids which have yet to be produced or characterized. High throughput screening of these molecules can guide material selection to expedite incorporation into devices [3]. We have specifically chosen to focus on melting point and viscosity of ionic liquids because those parameters are both important factors in the performance of battery electrolytes, one area that ionic liquids are particularly promising [4, 5].

In this work, we demonstrate accurate prediction of melting point and viscosity of an ionic liquid based on the chemical structure of the anion and cation. Each anion and cation is represented in the form of a SMILES string, a standard method for encoding the molecular structure into a unique set of ASCII characters specifying the atoms and bonds of a molecule. Thus the input into the algorithm is a pair of ASCII strings corresponding to the anion and cation structures, respectively (in addition to a temperature, for the case of viscosity prediction). We then use a message-passing neural network followed by a series of feed-forward layers to output a predicted value for either viscosity or melting point.

2 Related work

There exist a multitude of approaches to predict the properties of organic molecules, but quantum chemical methods such as density functional theory (DFT) have been the workhorse of computa-

CS230: Deep Learning, Winter 2020, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

tional chemistry and materials science due to the attractive trade-off between computation time and prediction accuracy [6]. However, these approaches are still too computationally expensive for high throughput screening of molecules because optimized 3D coordinates are required for high-accuracy predictions [6, 7]. These drawbacks make neural networks a promising alternative to enabling high throughput screening of candidate molecules, and such approaches have recently shown prediction accuracy exceeding that of hybrid DFT if given sufficient training data [8].

Several groups have attempted to use machine learning and neural network approaches to predict physical properties of ionic liquids based on chemical structure. Some approaches based on group contribution have been used with relative success, but such techniques require hands-on data labeling and do not lend themselves to generalization across entirely novel structures [9, 10]. Another approach has been to use ISIDA fragment descriptors to encode the varying bond environments across the molecules, however the results are lackluster at best, possibly due to the small (99) dataset used [11]. Yet further approaches use various tree or graph-based frameworks for encoding chemical structure, although such studies tend to focus exclusively on a single family of either anions or cations, and as such may be limited in their generality [12, 13]. Indeed, relatively low mean absolute errors (7.1%) have been recently achieved with perceptron-style networks, but such models rely on a least absolute shrinkage and selection operator (LASSO) to determine the feature space of the model, thereby eliminating the possibility of end to end learning [14].

In molecular property prediction, neural network architectures which operate on graph-based inputs are categorized under the umbrella of Message Passing Neural Networks (MPNNs). Such models encode molecules as graphs with node and edge features which can be updated according to the connectivity of neighboring nodes. These architectures outperform models with a fixed fingerprint because they enable end-to-end learning of each molecule [15]. It has been recently shown that MPNNs can accurately predict the properties of large polymeric molecules without the need for optimized 3D geometries. It was further shown that transfer learning can be implemented between poorly correlated properties, highlighting the ability of MPNNs to extract both high-level and low-level features from input data [7].

3 Dataset and Features

Our dataset contained experimentally measured physicochemical properties of ionic liquids which was compiled and tabulated from hundreds of peer-reviewed articles [16]. This dataset included 7,666 viscosity values of ionic liquids at various temperatures, and the experimentally determined melting point for 2,212 unique ionic liquids. The datasets consisted of a SMILES string encoding of an ion pair, the measurement temperature (if applicable), and the desired property. This unique encoding of the molecular structure allows for the 3-D connectivity of a molecule to be expressed as a canonical string. A visualization of one entry common to both datasets is shown in Table 1, along with the corresponding molecular structures.

The values in the viscosity dataset ranged several orders of magnitude, from 1.34 to 7,079,457.8 cP. As such, the logarithm of the viscosity was used as the labeled y-values so that both large and small datapoints would more equally contribute to the loss function during training. In the case of the melting point predictor, the y-values were scaled between -1 and 1, as this was found to improve performance of the network.

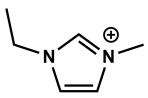
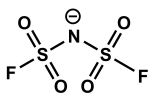
Cation	Anion	Temperature [K]	$\log_{10}(\eta)$ [cP]	Melting Point [K]
CCN1C=C[N+](=C1)C	C(F)(F)(F)S(=O)(=O)[N-]S(=O)(=O)C(F)(F)F	298	1.51	257
				

Table 1: Dataset visualization using the cation/anion pair of EMIM/TFSI. The first row shows SMILES string encoding and measured physical properties that are respectively used as the model inputs and outputs, while the second row contains the corresponding physical structure

4 Methods

The model implemented in this work borrows heavily from the neural fingerprinting architecture designed for other organic molecule systems [7], and is schematically depicted in Figure 1. As described above, the inputs are comprised of a pair of SMILES strings corresponding to the anion and cation respectively. Each string is then converted into a set of dictionaries corresponding to the atom, bond and connectivity information. Atom states are distinguished based on atomic symbol, charge state, bonding environment and aromaticity. Bond states likewise determined by type of bond (single, double, triple), component atoms, conjugation and presence in a ring. A pair of embeddings are then generated based on the size of the atom state and bond state dictionaries, with the depth of the embedding left as a tunable hyperparameter, and initialized using Xavier initialization.

As depicted in Figure 1, both the atom and bond embeddings are shared between the anion and cation molecules, such that an atom that exists in an anion references the same vector/matrix in the embeddings as an identical atom in a cation. Further, both the anion and cation are passed through the same layers of the message passing algorithm with shared weights. This was done to both limit the number of trainable parameters due to the limited size of available datasets, and because any relevant features for prediction should be present in both the anion and cation populations.

The message passing steps are carried out using the matrix multiplication method [7, 17]. At each time step, the message m is calculated for each atom ν, according to the equation below. For each w of its bonded neighboring atoms $N(\nu)$, a matrix multiplication is performed between the bond between the bond embedding matrix corresponding to that bond type $A_{e_{\nu w}}$ and the atom feature embedding h_w^t calculated at the previous time step. The resulting vectors are then summed to obtain the updated message m_ν^{t+1} which is then passed into a gated recurrent layer wherein the atom feature embedding is updated. Following the given number of message passing steps ($M = 4$ in this case), the resulting atom embedding vectors are passed through a dense regression layer and then summed on a per-molecule basis to generate a molecular fingerprint of tunable size.

$$m_\nu^{t+1} = \sum_{w \in N(\nu)} A_{e_{\nu w}} h_w^t \qquad\qquad h_\nu^{t+1} = GRU(h_\nu^t, m_\nu^{t+1})$$

Because an identical encoding is performed on both the anion and cation, each molecular fingerprint is passed through an independent dense layer before taking the sum of the two results. This dense pre-combination layer mathematically then allows for such cross-interactions between non-directly correlated properties, such as mf_i^{AN} and mf_j^{CAT}. The resulting element-wise summed vectors are then piped to the corresponding network's prediction head.

In the case of melting point prediction (shown schematically in turquoise in Figure 1) the final output is of dimension 1, and is identified directly as $\hat{y}$, with a mean squared error loss function against the known value y. Viscosity prediction uses a slightly more advanced head, as shown in gold in Figure 1, where the final output is replaced with a vector of dimension 3. Using an empirical relationship between viscosity and temperature, the vector is mapped using built-in Keras functions to include the temperature dependence, resulting in $\hat{\eta}$ against which a mean squared error loss function is used. In this way, the full size of the dataset may be leveraged to predict the parameters we are interested in without the need for learning the temperature dependence.

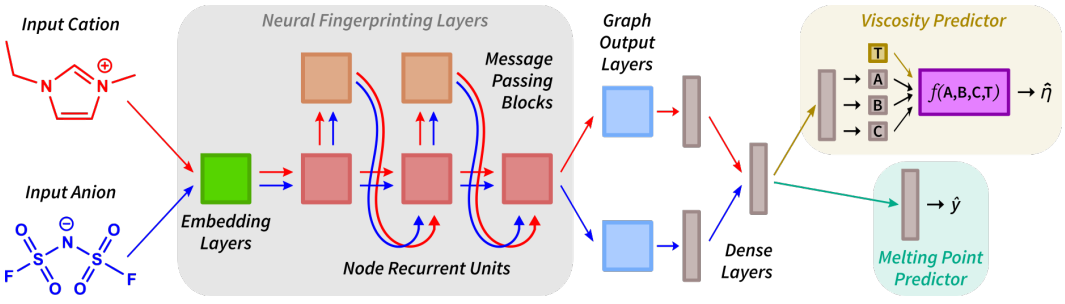


Figure 1: Model Architecture, inspired by [7]

5 Hyperparameter Tuning Experiments

When optimizing viscosity and melting point models, a priority was placed on tuning the base shared architecture and learning rate. The architectural hyperparameters addressed, as well as the range across which they were surveyed, were as follows: Atom Features: depth of the atom embedding matrix [8, 32]; Fingerprint Size: size of anion/cation fingerprint graph [8, 64]; Mixing Size: output length of anion/cation representations prior to "mixing" [8, 32]; and number of message passing steps: determining the degree of accounted bonding environment [1, 4].

The number of message passing steps was limited a maximum value of 4 to prevent atoms in 5-membered rings from retaining redundant information about themselves in their extended bonding environments. Experimentation was skewed towards higher valued hyperparameters under the assumption that larger networks would allow the base to learn more low/high-level features related to physiochemical properties.

Table 2 shows a sampling of melting point performances of the varied hyperparameters, ordered by highest R^2_{train} between true and predicted values (analogous to mean squared error). Those which performed best on test and train sets tended to have Atom Features = 32 and 3 to 4 message passing steps. Fingerprint size ≤ 64 and mixing size ≤ 32 tended to perform better, though the exact values were not as critical. The final model chosen for further optimization is highlighted Index 13 of in Table 2, given its high R^2_{dev} performance.

Index	R^2_{train}	R^2_{dev}	Atom Features Size	Fingerprint Size	Mixing Size	Message Passing Steps (M)
1	0.913162	0.708920	32	8	12	4
2	0.910837	0.691588	32	8	8	4
3	0.910206	0.683271	32	8	32	4
⋮	⋮	⋮	⋮	⋮	⋮	⋮
12	0.896435	0.674678	32	8	32	3
13	0.895648	0.713652	32	32	20	4
14	0.894692	0.664827	32	16	32	3
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 2: Sample of melting point prediction performance of explored hyperparameter space

Learning rate was tuned manually, using learning rate step decay. Adjustments to the learning rate decay and scheduling were made by monitoring the loss curve, with best results achieving using an initial learning rate of 0.01 and 0.55 factor drop every 100 epochs. In an effort to reduce variation on melting point prediction, regularization and dropout were introduced. However, neither had a strong effect on reducing variation. Both viscosity and melting point models were individually trained until training and development loss curves plateaued. Since development loss did not diverge and variation was not reduced with regularization or dropout, we conclude that our models did not overfit to training data.

6 Results & Discussion

With the hyperparameter choices determined above, the viscosity and melting point prediction models were run for 1000 epochs each. As can be seen in Figure 2a, the viscosity predictions lead to impressive R^2 scores, with a tight prediction distribution across the full 8 orders of magnitude spanned by the dataset. However, despite such remarkable distributions, mean absolute error across the train and dev sets are calculated at $0.124 \log(cP)$ and $0.155 \log(cP)$ respectively. When transformed back to original cP units, that translates to (-24.9%, +33.0%) error for train set and (-30.0%, +42.9%) error for dev set [18]. Given the huge range spanned by the dataset, although this work still has yet to outperform some of the tailor-made networks with hand-designed components [14], it nevertheless demonstrates a significant step toward full end-to-end learning of materials properties from structure.

On the other hand, the melting point predictor seen in Figure 2b demonstrates a significant improvement over any of the networks attempted elsewhere on the same dataset, for both the train and dev

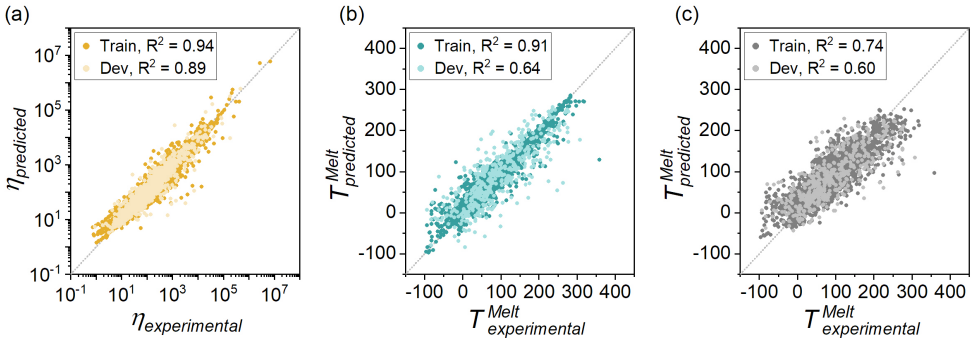


Figure 2: Prediction graph for (a) viscosity and (b) melting point prediction through independent networks, as well as (c) melting point prediction with weights transferred from viscosity model (a)

sets [10]. Although significant variance remains even in the best models, neither reshuffling nor modest regularization techniques are able to rectify the issue. Most likely, the dataset of melting point values is still simply too small to allow for proper learning and generalization.

In an attempt to counteract the weaknesses associated with having an insufficient dataset, we moved to pre-train the melting point predictor using the learned weights from the viscosity prediction model from Figure 2a. In this case, all weights from the basic model up until the addition layer were saved and set to non-trainable, while only the final dense regression layer associated with the melting point prediction head was set to be trainable. The results following 1000 epochs are shown in Figure 2c. Noticeably, the overall performance of the network is lower, although the apparent variance is decreased as well. While it is difficult to make any conclusive claims due to the greater bias, a decreased variance across the model would be consistent with the advantages associated with transfer learning.

The increased bias due to the transfer learning can be explained straightforwardly based upon further examination of the respective datasets. While the majority of atoms and bonds contained within the superset of the SMILES strings were shared commonly, approximately $\frac{1}{6}$ of the features contained in the melting point dataset were not present in the viscosity dataset. Consequently, the corresponding entries in the atom and bond embeddings were never trained during viscosity prediction, and remained at the initialized values. Because the embedding layers were set to be non-trainable for the transfer learning, a proper calculation of the respective molecular fingerprints could not be done, and the performance therefore suffered.

7 Conclusions and Future Work

From the hyperparameter tuning performed, it is apparent that the depth of the atom embedding matrix is a critical parameter that should be substantial enough in size. Excessive limitation on the number of atom features will result in a lower performing model. Additional message passing steps also help model performance, as expected due the larger resulting environment of consideration.

Both the viscosity and melting point prediction highlighted the challenges associated with small datasets. Although the viscosity dataset contained 7,666 distinct viscosity values, the number of distinct structures was considerably smaller due to the range of temperature values at which viscosity was measured. As the full dataset was shuffled into train and dev sets, several of the anion/cation pairs were likely in both sets, albeit at differing temperatures. Consequently, the R^2_{dev} values are probably not a *true* reflection of the predictive properties of the network, and in the future an entirely distinct dataset should be used for testing. In the melting point predictor as well, a larger dataset would have proven beneficial to counteract some of the variance measured.

To mitigate both effects, future work could focus on training the entire network on both viscosity and melting point simultaneously, i.e. $\hat{y} = \begin{pmatrix} \hat{\eta} \\ T_m \end{pmatrix}$ with a customized mean squared error loss function that would simply ignore absent values. In such a way, the detrimental effects of freezing training as seen in the transfer learning of Figure 2c would be negated, and the entire embedding would be able to learn in unison.

Author Contributions

Yaakov Tuchman adapted the source code from [7, 19] to allow for pairs of molecules and temperature to be passed through the network. Yaakov Tuchman and Tyler Quill wrote the code used for training the networks and performed training. Tyler Quill and Shayta Roy performed hyperparameter tuning of the network and hyperparameter selection. Yaakov Tuchman, Tyler Quill, and Shayta Roy contributed to initial literature and data set search, proposal writing, milestone writing, and final report writing.

The authors are also grateful to Dr. Peter St. John and Dr. Nolan Wilson for their valuable insights on architecture development in particular and end-to-end deep learning in general.

References

[1] Tsukasa Torimoto, Tetsuya Tsuda, Ken ichi Okazaki, and Susumu Kuwabata. New frontiers in materials science opened by ionic liquids. *Advanced Materials*, 22(11):1196–1221, March 2010.

[2] Natalia V. Plechkova and Kenneth R. Seddon. Applications of ionic liquids in the chemical industry. *Chem. Soc. Rev.*, 37(1):123–150, 2008.

[3] Vishwesh Venkatraman, Sigvart Evjen, Kallidanthiyil Chellappan Lethesh, Jaganathan Joshua Raj, Hanna K. Knuutila, and Anne Fiksdahl. Rapid, comprehensive screening of ionic liquids towards sustainable applications. *Sustainable Energy & Fuels*, 3(10):2798–2808, 2019.

[4] Andrzej Lewandowski and Agnieszka Świderska-Mocek. Ionic liquids as electrolytes for li-ion batteries—an overview of electrochemical studies. *Journal of Power Sources*, 194(2):601–609, December 2009.

[5] Maofeng Wang, Zhongqiang Shan, Jianhua Tian, Kai Yang, Xuesheng Liu, Haojie Liu, and Kunlei Zhu. Mixtures of unsaturated imidazolium based ionic liquid and organic carbonate as electrolyte for li-ion batteries. *Electrochimica Acta*, 95:301–307, April 2013.

[6] Wolfram Koch and Max C. Holthausen. *A Chemist's Guide to Density Functional Theory*. Wiley, July 2001.

[7] Peter C. St. John, Caleb Phillips, Travis W. Kemper, A. Nolan Wilson, Yanfei Guan, Michael F. Crowley, Mark R. Nimlos, and Ross E. Larsen. Message-passing neural networks for high-throughput polymer screening. *The Journal of Chemical Physics*, 150(23):234111, June 2019.

[8] Felix A. Faber, Luke Hutchison, Bing Huang, Justin Gilmer, Samuel S. Schoenholz, George E. Dahl, Oriol Vinyals, Steven Kearnes, Patrick F. Riley, and O. Anatole von Lilienfeld. Prediction errors of molecular machine learning models lower than hybrid DFT error. *Journal of Chemical Theory and Computation*, 13(11):5255–5264, October 2017.

[9] Kamil Paduszyński and Urszula Domańska. Viscosity of ionic liquids: An extensive database and a new group contribution model based on a feed-forward artificial neural network. *Journal of Chemical Information and Modeling*, 54(5):1311–1324, May 2014.

[10] Vishwesh Venkatraman, Sigvart Evjen, Hanna K. Knuutila, Anne Fiksdahl, and Bjørn Kåre Alsberg. Predicting ionic liquid melting points using machine learning. *Journal of Molecular Liquids*, 264:318–326, August 2018.

[11] I. Billard, G. Marcou, A. Ouadi, and A. Varnek. In silico design of new ionic liquids based on quantitative structure-property relationship models of ionic liquid viscosity. *The Journal of Physical Chemistry B*, 115(1):93–98, January 2011.

[12] Riccardo Bini, Cinzia Chiappe, Celia Duce, Alessio Micheli, Roberto Solaro, Antonina Starita, and Maria Rosaria Tiné. Ionic liquids: prediction of their melting points by a recursive neural network model. *Green Chemistry*, 10(3):306, 2008.

[13] Gonçalo V.S.M. Carrera, Luís C. Branco, Joao Aires de Sousa, and Carlos A.M. Afonso. Exploration of quantitative structure–property relationships (QSPR) for the design of new guanidinium ionic liquids. *Tetrahedron*, 64(9):2216–2224, February 2008.

[14] Wesley Beckner, Coco M. Mao, and Jim Pfaendtner. Statistical models are able to predict ionic liquid viscosity across a wide range of chemical functionalities and experimental conditions. *Molecular Systems Design & Engineering*, 3(1):253–263, 2018.

[15] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alan Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2224–2232. Curran Associates, Inc., 2015.

[16] Vishwesh Venkatraman, Sigvart Evjen, and Kallidanthiyil Chellappan Lethesh. The ionic liquid property explorer: An extensive library of task-specific solvents. *Data*, 4(2):88, June 2019.

[17] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. *CoRR*, abs/1704.01212, 2017.

[18] Robert Nau. The logarithm transformation. `https://people.duke.edu/~rnau/411log.htm`, 2019.

[19] Keras layers for end-to-end learning on molecular structure. `https://github.com/NREL/nfp`.