A System for Learning Atoms Based on Long Short-Term Memory Recurrent Neural Networks

Zhe Quan[†], Xuan Lin[†], Zhi-Jie Wang[‡,#], Yan Liu[†], Fan Wang[⊤,⊥], and Kenli Li[†]
[†] College of Information Science and Engineering, Hunan University, Changsha, China
[‡] School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China
[#] Guangdong Key Laboratory of Big Data Analysis and Processing, Guangzhou, China
[⊤] College of Chemistry, Central China Normal University, Wuhan, China
[⊥] Key Laboratory of Pesticide & Chemical Biology of Ministry of Education, Wuhan, China
quanzhe@hnu.edu.cn, jack_lin@hnu.edu.cn, wangzhij5@mail.sysu.edu.cn,
lyan@hnu.edu.cn, fwang@mail.ccnu.edu.cn, lkl@hnu.edu.cn

***Abstract*—In recent years, researchers in the fields of bioinformatics and cheminformatics have attempted to utilize machine learning methods for molecule modeling, bioactivity prediction, chemical property prediction, biology analysis, etc. In this paper, we present a system that merges the merits of various techniques such as long short-term memory (LSTM) recurrent neural networks, and is designed for learning atoms and solving the classic problems such as single task classification in the field of drug discovery. We have implemented our approach and conducted extensive experiments based on several widely used datasets such as SIDER and Tox21. The experimental results consistently demonstrate the feasibility and superiority of our proposed approach.**

***Index Terms*—machine learning, drug discovery, neural networks, molecule data.**

I. Introduction

Data-driven analysis plays a crucial part in many biological and chemical applications, including molecule modeling [1], chemical property prediction [2] and pharmacogenomics [3]. With the rapid development of machine learning [4]–[6], in recent years researchers in the fields of bioinformatics and cheminformatics have attempted to utilize machine learning methods for molecule modeling, bioactivity prediction, chemical property prediction, biology analysis, and so on [5], [7]–[9].

As we know, SMILES [10] (simplified molecular input line entry system) strings are usually used to represent and store molecule datasets, and they are in form of a single line text consisting of molecular notations. In realistic world, a molecule of arbitrary size and shape could be hard to be represented and used for machine learning tasks. Users usually need to transform them into other formats that are easy to be handled by machine learning algorithms. A widely adopted proposal is to use hand-crafted feature like ECFP [11], Coulomb Matrix [12], Graph-like structure [9], and so on. Such a process is usually called *featurization*. The transformed data (or featurization data) is usually as input and feed it into the interface of machine learning methods, such as classifier like random forest and multilayer perception, and so on [13]–[16].

Owing to the success of solving a wide range of machine learning problems by the Artificial Neural Networks(ANNs) [6], recently, *long short-term memory* (LSTM) *recurrent neural networks* [17] have emerged as powerful generative models in various domains including natural language understanding [18], images [19] , and video [20]. This lines of models regard the input data as sequential lists, and they are very suitable for solving time-dependent tasks like natural language understanding [21]. On the other hand, as shown in [22], the unsupervised machine, Atom2Vec [23], can learn the basic properties of atoms, and is used to discover the periodic table of the elements.

Inspired by the remarkable achievements mentioned above, in this paper we present a system that merges the merits of various techniques and is designed for solving the classic problems such as single-task classification and multi-task classification in the field of drug discovery. Generally, our system first transforms the molecule data in the SMILES format into a set of sample vectors via a component named *format transformer* (FT). The FT divides by spaces the molecule in SMILES format into atoms, which may consists of symbols and numbers. These atoms are then encoded by one-hot encoding, which allows us to transform atoms into a specific vectors with some certain dimensions. Then it uses a manner similar to word2vec [23] to extract the sample vectors by training the specific vectors previously obtained. These vectors are then as the input and fed into the embedding layer, which is used to extract high dimensional features. Meanwhile, in this layer a large matrix is constructed, which is convenient for model training at the upper layer (i.e., LSTM layer). The extracted features are then trained at the LSTM layer, and finally the trained samples are sent to a classifier (e.g., sigmoid) for single or multiple task classification.

In summary, the contributions of this paper are twofold: (i) we present an approach to learn atoms and solve the classic problems including single- and multi-task classification

in the field of drug discovery; and (ii) we conduct extensive experiments based on several widely used molecule datasets and demonstrate the feasibility and superiority of our proposed approach. (The codes of our implementation are to be shared at the open-source code repository, GitHub, after the paper is accepted.) The rest of the paper is organized as follows. Section II reviews prior works most related to ours. Section III presents our approach. Section IV analyzes and discusses the performance results of our approach. Section V concludes this paper.

II. Related Work

In the past decade, deep neural networks (DNN) has gained remarkable achievements in various areas of machine learning research. Deep neural networks can learn the potential regular pattern with access of training a large number of datasets to obtain better performance analysis and prediction. Different from other domains, DNN in drug discovery depends heavily on molecular featurization. The main molucular representation methods at present are ECFP [11] [24], Coulomb Matrix [12], and Graph-like structure [9], and so on. Recently, deep neural models have opened up new avenues for modeling SMILES strings as a language model. The unsupervised machines (Atom2Vec) shown by [22] can learn the basic properties of atoms. Our work is inspired by theirs, yet it is different from theirs. In that paper the methods and experiments are used to discover the periodic table of the elements. In addition, they mainly focused on the principle explanation of atom representation, while related model designs are not covered. Another unsupervised approach named *conditional diversity networks* [25] also transform SMILES string into vector, while the detailed steps are not covered, and they paid more attention on the generation of molecule and drug, instead of the tasks mentioned in this paper.

On the other hand, there are some researches concentrated on drug predictions including adverse drug events (ADEs), adverse drug reactions (ADRs), adverse drug-drug interactions (DDIs), activity prediction, and so on. For example, Page *et al.* [26] identified the adverse drug events by relational learning. In addition, several methods proposed by [27] are for extracting ADRs from forum posts and tweets. Xiao *et al.* [28] provided the efficient solutions for the real world ADR prediction, and cast the ADR-drug relation structure into a three-layer hierarchical Bayesian model. As for adverse drug-drug interactions (DDIs), most of methods focused on binary prediction (with or without DDI). Jin *et al.* [29] formulated such a problem as a multitask dynadic regression problem. Dietterich [30] introduced the dynamic reposing technique that iteratively learns a neural network, and then focused on maximizing the predicted output values. Warmuth *et al.* [31] used the active learning techniques for selecting the successive batches, and adopted three selection strategies in the drug discovery process. Ma *et al.* [32] made better trade-off between accuracy and interpretability. Our work shares a common feature with this line of works, since both discussed the issues related to drug discovery. Nevertheless, our work is different from their in two points at least: (i) they mainly focused on developing techniques for other tasks instead of single- and multi-task classification; and (ii) the LSTM recurrent neural networks (RNNs) are not covered in their works.

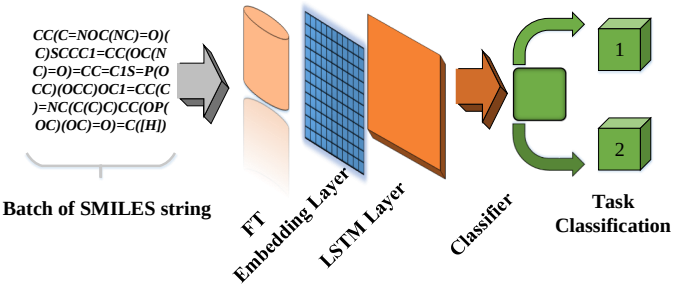

Fig. 1. The overall architecture of our approach

III. Approach

In this section, we first give an overview of our approach, and then present each part of our approach in detail.

A. Overview

Figure 1 shows the overall architecture of our approach. Briefly, the molecule data in the format of SMILES is first processed by a component named *format transformer* (FT), which transforms the molecule data into a set of sample vectors. These vectors are then as the input and fed into the embedding layer, which is used to extract high dimensional features. The extracted features are then trained at the LSTM layer using the long short-term memory (LSTM) recurrent neural networks. The output of the LSTM layer shall be processed by the classifier, which is used to generate the output label for task classification.

B. Representation

Choosing a proper molecular representation is at the heart of computer-based chemical analysis, and it is also very importance for drug discovery and prediction, since one may need to analyze and predict properties of drug with the same or similar molecule representation.

In the real world, most biological and chemical datasets are in the format of SMILES string. The SIMILES string of a unique molecule is a single line text representation. For example, a molecule is encoded as a linearly arranged string $s_i = s_1, s_2, ...s_i (i = 1, 2, ..., n)$. The encoding rules of SMILES follows the strict grammars, which consists of symbols indicating element types, bond values, and the start and terminal location for ring closures and branching components.

SMILES strings are powerful for representing and storing molecule data. To apply machine learning methods for learning advanced features, we need to transform them into the new format suitable for utilization. Take *Aldicarb* and its SMILES string $CC(/C = N/OC(NC) = O)(C)SC$ as an example, one can convert it by RDKit [33] software into a graph-structured representation, which can be later used for learning features via graph convolutions. Instead of using chemical

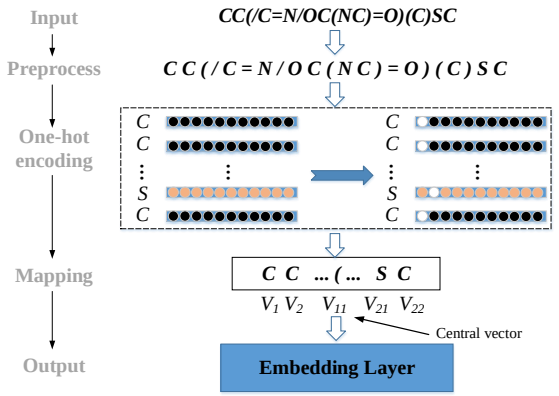

Fig. 2. The general process of FT

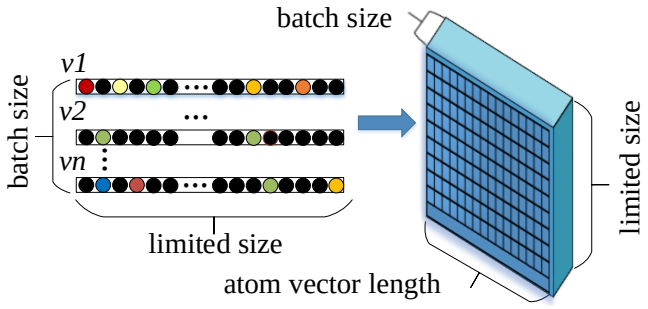

Fig. 3. The workflow of embedding implementation.

software such as RDKit to transform SMILES strings, we adopt another manner that directly transforms them into atom vectors. Briefly, molecule in SMILES format is first preprocessed as an independent atom symbol, and then they are expressed as a high dimensional vectors, which are sample vectors and also machine-readable characters or strings. Figure 2 illustrates the general process of this transformation. Next, we present the details of our transforming method, named *Format Transfer* (FT).

C. Format Transformer

As we known, in natural language processing (NLP), the sentences are processed using word vectors. We observe that SMILES is a linguistic grammar that employs an alphabet of characters to describe molecule, and each element or symbol has an associated definition in SMILES. Here we use the similar way in NLP to handle the SMILES strings. Specifically, for a series of molecules $s_i = s_1, s_2, ...s_i (i = 1, 2, ..., n)$ in the format of SMILES, we divide them into a series of atoms by space. Each single atom x_i may consists of symbols and numbers. Then, for all preprocessed atoms, we encode them by *one-hot encoding* [34], which allows us to transform atoms into a specific vectors $v_i (i = 1, 2...n)$ with some certain dimensions. Since these specific vectors may have less feature information. To obtain the sample vectors, we use a manner similar to *word2vec* [23] to extract the sample vectors, by training the specific vectors previously obtained. The sample vectors will be used as the input of the embedding layer.

Before the extraction process (i.e., training the specific vectors), we suggest a technique named *simplified feature learning* (SFL). This is based on the follow observation: molecules in SMILES format consist of numbers and characters, while some symbols and numbers may represent repetitive information, which may cause the complicated training process. For example, Toluene is denoted as $Cc1ccccc1$ in SMILES, a benzene ring is represented by number "1", while c and C denote aromatic and aliphatic carbon atoms, which essentially imply the existence of a benzene ring. Therefore, our SFL ignores some feature information that has been already expressed, and adds the occurrence frequency of each element (as the additional information) to the specific vector. These strategies ensure the simplicity and integrity of the featuring information.

D. Embedding Layer

When the sample vectors as the inputs are fed into the embedding layer, a huge matrix will be constructed, which is convenience for model training in the later steps. The size of the matrix depends on the product of the batch size and the limited size (i.e., the dimension of an atom vector). Note that, every vector v_i, which is encoded using an N-bit status register. Each state has its own register bit, and at any time only one of them is valid. A workflow of embedding implementation is shown in Figure 3. The construction of the embedding matrix is based on the following principle: the element v_i located in the center of window k will be output object, and the other elements are the input ones (for example, see Figure 2, if atoms $v_i^* (i = 1, 2, ..., 22)$ located in window k are as input, then the atom $v_i (i = 11)$ located in the central of window k will be output). In this way, it guarantees the output transmission channel unobstructed and ordered. In addition, each vector v_i encoded by one-hot encoding shall automatically find the corresponding vector in the pretrained matrix, until the mapping process completes.

E. LSTM Layer

Similar to the methods for dealing with semantics similarity in NLP, we use the *long short-term memory* (LSTM) recurrent neural network [17]. The LSTM is an alternative RNN, it uses the so-called *memory cell* (controlled by input, output and forget gates) to replace the *conventional neuron* in order to overcome the vanishing gradient problem of traditional RNNs. See Figure 4 for an illustration. In short, LSTM is a special class of RNN [35] that is capable of capturing long sentence relationships.

Owing to the existence of the gate of the adopted model, we can learn and recognize the information needed to be retained or forgotten. In our approach, each atom including special symbols (e.g., =, ≡), has a corresponding time step x_t (t=1,2...n). The intermediate state associated with each time step is referred to as a hidden state vector h_t. This hidden state vector is used to encapsulate and summarize all the information appeared in the previous time step. The hidden state is a function of the current atom vector and the hidden

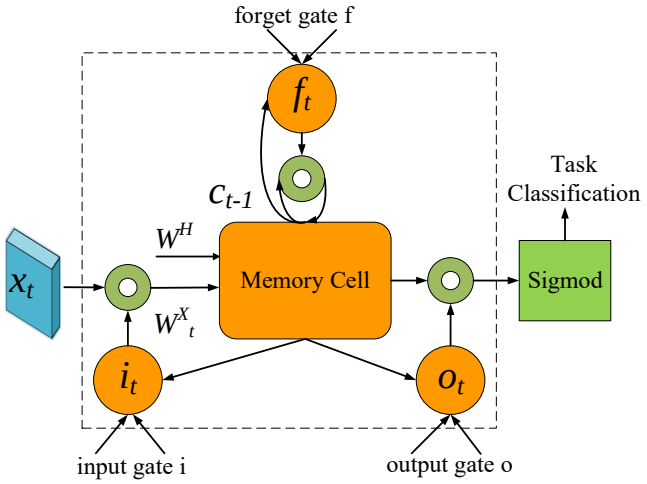

Fig. 4. An illustration of LSTM Memory Cell.

state vector of the previous step. The potential value of the hidden state vector would be

$$h_t = \sigma(W^H h_{t-1} + W^X x_t) \tag{1}$$

where W^H and W^X represent the weighted matrices. The value of W^H stays the same in all time steps, but the value of W^X changes in every input. The size of these values is not only affected by the current vector, but also by the previously hidden layer. It is easily observed that the value of h_t will vary when W^H and W^X change. For example, when W^H changes greater than W^X, h_t is more affected by h_{t-1} than x_t. In other words, the current hidden state vector is more concerned with a sum of the previous atom vectors than the current one.

Finally, the state vector of the hidden layer is sent to a classifier (e.g., *sigmoid*) for task classification or property prediction. Thus, the designed approach with learnable ability can predict whether the tested molecule is toxic or not.

IV. Experiments and Discussions

In this section, we cover and analyze the performance results of our system, based on several commonly used molecule datasets from the MoleculeNet Benchmark [36]. To evaluate the performance of system, we adopt the ROC-AUC score [36] (larger is better) throughout the paper. Everything necessary to reproduce our results can be found in the opensource code repository mentioned in Section I.

Note that, some datasets are single-task datasets while others are multi-task datasets (more detailed descriptions shall be discussed later). In our experiments, the output of the final layer is changed according to the number of tasks, while the other steps in our approach are applied to all tested datasets. Our system was trained based on Tensorflow [37], and Adam algorithm [38] is used to optimize all the parameters of the adopted neural networks.

A. Data Description

The datasets we used are BACE, BBBP, Tox21, and SIDER (see Table I). These datasets consist of a mix of physical and non-physical properties, single-task and multi-task classification problems. In these datasets, SMILES strings are used to encode input molecules. The details of these data are as follows.

- *BACE.* The BACE dataset provides quantitative and qualitative binding results, and is a collection of 1522 compounds with their 2D structures and binary labels. It is used as a classification task.
- *BBBP.* The Blood-brain barrier penetration (BBBP) dataset concentrates on the modeling and prediction of the barrier permeability. This dataset includes binary labels for over 2000 compounds on their permeability properties.
- *Tox21.* The Toxicology in the 21st Century (Tox21) dataset contains qualitative toxicity measurements for

TABLE I
Characteristics of the selected classification datasets used in model evaluation

Dataset	Category	Description	Data Type	Tasks	Classification Type	Compounds	Rec-Metric
BACE	Biophysics	quantitative inhibity	SMILES	1	Classification	1,522	ROC-AUC
BBBP	Physiology	barrier permeability	SMILES	1	Classification	2,053	ROC-AUC
Tox21	Physiology	toxicity	SMILES	12	Classification	8,014	ROC-AUC
SIDER	Physiology	side reactions	SMILES	27	Classification	1,427	ROC-AUC

TABLE II
ROC-AUC scores of various approaches in BACE and BBBP datasets.

Model \ Dataset	BACE				BBBP			
	Random		Stratified		Random		Stratified	
	test	validate	test	validate	test	validate	test	validate
LR	0.6888	0.6040	0.6412	0.6603	0.7374	0.6819	0.7277	0.8004
RF (n=10)	0.7662	0.7634	0.7488	**0.7521**	0.8106	0.7141	0.7358	0.8074
SVM	0.6045	0.5187	0.5279	0.5400	0.6700	0.5573	0.5874	0.6193
KNN	0.7598	0.7498	0.7603	0.7335	0.7456	0.8049	0.8142	0.6718
DT	0.6863	0.7272	0.7320	0.7416	0.7652	0.6973	0.7552	0.7692
Ours	**0.8144**	**0.8619**	**0.7628**	0.7330	**0.8320**	**0.8855**	**0.8759**	**0.9356**

TABLE III
ROC-AUC SCORES OF EACH TASK IN TOX21. THE RESULTS ARE BASED ON THE STRATIFIED SPLIT METHOD.

Model / Task	RF		SVM		**Ours**	
	test	validate	test	validate	test	validate
NR-AR	0.6732	0.6730	0.4951	0.5098	**0.6914**	**0.6909**
NR-AR-LBD	0.6384	0.5825	0.5216	0.5208	**0.7477**	**0.7228**
NR-AhR	0.5980	0.6076	0.6396	0.6160	**0.6780**	**0.6698**
NR-Aromatase	**0.5500**	**0.5798**	0.5458	0.5486	0.4964	0.4991
NR-ER	0.5507	0.5433	0.5000	0.4992	**0.6231**	**0.5546**
NR-ER-LBD	0.5170	**0.5931**	0.5216	0.5436	**0.5308**	0.5256
NR-PPAR-gamma	0.5263	0.4984	0.5074	0.4944	**0.5659**	**0.5000**
SR-ARE	0.5568	0.5562	0.6355	0.5804	**0.6414**	**0.5901**
SR-ATAD5	**0.5348**	**0.5356**	0.4931	0.4982	0.5000	0.5171
SR-HSE	0.5124	0.5107	0.5161	0.4986	**0.6120**	**0.6381**
SR-MMP	0.6862	0.6809	0.6489	0.6596	**0.7425**	**0.7438**
SR-p53	0.5138	0.5310	0.4931	0.4982	**0.5180**	**0.5149**

8014 compounds on 12 different targets, including nuclear receptors and stress response pathways.

- *SIDER.* The side effect resource (SIDER) is a dataset of marked drugs and adverse drug reactions (ADR), and it contains 1427 compounds on 27 system organ classes.

In general, conventional machine learning methods require datasets to be split into training/validating/testing subsets for benchmark. Usually, models are trained by training sets, hyperparameters are tuned through validating sets, and testing sets are used for evaluating models/approaches/systems. In our experiments we use two split methods mentioned in [36]. That is, random splitting and stratified random splitting. Usually, when there is some bias, random splitting is used for data splitting. In contrast, stratified random splitting ensures that each subset contains the full range of labels. To keep the same benchmark, in our experiments, random and stratified (random) split method with a 8/1/1 ratio are adopted for BBBP and BACE datasets, while only stratified (random) split method is adopted for Tox21 and SIDER datasets (since random split exhibits poor performance for all these methods).

B. Single Task Classification

For single-task classification (i.e., on BACE and BBBP datasets), we train several machine learning algorithms as baselines on an identical single-task dataset. These algorithms

TABLE IV
ROC-AUC SCORES OF EVERY TASK IN SIDER. THE RESULTS ARE BASED ON THE STRATIFIED SPLIT METHOD.

Task	RF		SVM		**Ours**	
	test	validate	test	validate	test	validate
Hepatobiliary disorders	0.5654	0.5696	0.5553	0.5733	**0.5843**	**0.6916**
Metabolism and nutrition disorders	**0.5490**	0.5337	0.5083	0.5091	0.5345	**0.5382**
Product issues	0.4906	0.5087	0.5032	0.5000	**0.5048**	**0.5120**
Eye disorders	0.5034	0.4984	0.5071	0.5179	**0.5087**	**0.5260**
Investigations	0.4877	**0.5185**	0.5014	0.5179	**0.5045**	0.4917
Musculoskeletal and connective tissue disorders	0.5180	0.5234	0.5116	0.5111	**0.5620**	**0.5355**
Gastrointestinal disorders	0.5551	0.4962	0.4926	0.5385	**0.5564**	**0.5686**
Social circumstances	0.4828	0.5185	0.4918	0.4958	**0.5170**	**0.5204**
Immune system disorders	**0.5558**	0.5271	0.5024	0.4950	0.5248	**0.5465**
Reproductive system and breast disorders	0.5436	0.5863	0.5601	**0.6101**	**0.5956**	0.5689
Neoplasms benign, malignant and unspecified (incl cysts and polyps)	0.5363	**0.5744**	0.5388	0.5676	**0.5396**	0.5073
General disorders and administration site conditions	0.4922	0.4962	0.4826	0.4955	**0.4929**	**0.5061**
Endocrine disorders	0.5245	0.5015	0.4920	0.4959	**0.5323**	**0.5301**
Surgical and medical procedures	**0.5228**	**0.5609**	0.4955	0.4858	0.4960	0.5000
Vascular disorders	0.4976	0.4981	0.5036	0.5129	**0.5011**	**0.5132**
Blood and lymphatic system disorders	0.5373	0.5248	0.5036	0.5833	**0.5408**	**0.5869**
Skin and subcutaneous tissue disorders	0.5417	0.5335	0.4959	0.4915	**0.5642**	**0.5407**
Congenital, familial and genetic disorders	0.4713	0.5015	0.4971	0.4950	**0.5000**	**0.5059**
Infections and infestations	0.5139	0.5005	0.4989	0.5135	**0.5148**	**0.5224**
Respiratory, thoracic and mediastinal disorders	0.4893	0.4952	**0.5122**	0.5238	0.4954	**0.5335**
Psychiatric disorders	0.5282	0.5126	0.5073	0.5070	**0.5378**	**0.5168**
Renal and urinary disorders	0.5632	0.5514	0.5137	0.5234	**0.5767**	**0.6394**
Pregnancy, puerperium and perinatal conditions	0.4769	0.4885	**0.4962**	**0.5288**	0.4961	0.4885
Ear and labyrinth disorders	**0.5617**	**0.5781**	0.5000	0.4938	0.5012	0.4851
Cardiac disorders	0.5530	**0.5871**	0.4899	0.5000	**0.5734**	0.5104
Nervous system disorders	0.4890	0.5346	0.5417	0.5147	**0.5147**	**0.5522**
Injury, poisoning and procedural complications	0.5262	0.5333	0.4947	0.5145	**0.5315**	**0.5546**

include LR (Logistic Regression) [39], RF (Random Forest) [13], SVM (Support Vector Machine) [14], DT (Decision Tree) [15] and KNN (K Nearest Neighbor) [16]. Also, for datasets used for these baselines, we use random and stratified splitting method with a 8/1/1 ratio. The comparison results are shown in Table II. It can be seen that our system generally outperforms these compared methods in both single-task datasets (i.e., BACE and BBBP). In particular, compared with the strongest baseline RF, in average, our system achieves an improvement of about 5% in these two datasets. These results demonstrate the competitiveness of our system. In addition, we observe that, for these two data splitting methods (i.e., random and stratified), there is no significant impact on our system. This indicates the stability of the system.

C. Multi-Task Classification

In order to comprehensively analyze the predictive performance of our system for multiple task classification, experiments at this stage are conducted on each task in Tox21 and SIDER datasets. In this set of experiments, we only show the results of RF and SVM methods, since they present better performance than other baselines for multiple task classification. Specifically, there are 12 and 27 tasks for Tox21 and SIDER respectively. The compared results on Tox21 are shown in Table III. It can be seen that, on the whole our system shows competitive performance on the Tox21 dataset. Specifically, on 10 out of 12 tasks, our system achieves the best performance compared against the baselines.

In addition, the results on the SIDER dataset are shown in Table IV. We can see that, on 20 out of 27 tasks, our system achieves the best performance. Overall, our system is still competitive, although it is inferior to some approaches on several tasks. But as we known, low ROC-AUC scores appeared commonly in most models found in the literature. Our system obtains much more leading scores for most tasks, and reaches the best results on more than 75% of the whole tasks, which essentially imply that our approach can effectively learn the structure and related information between atoms and the corresponding contexts.

V. Conclusion

In this paper, we presented an approach for learning atoms and solving the problem of single and multiple task classification in the field of drug discovery. Our approach transforms the molecule data in the SMILES format into a set of vectors via a component named the format transformer, and then feed them into the LSTM networks for training the sample vectors. We conducted extensive experiments based on several widely used molecule datasets. The results demonstrated the feasibility and superiority of our proposed approach. In the future, we would like to further optimize our approach and solve other related problems in the fields of drug discovery and bioinformatics.

References

[1] Junshui Ma, Robert P. Sheridan, Andy Liaw, George E. Dahl, and Vladimir Svetnik. Deep neural nets as a method for quantitative structure-activity relationships. *Journal of Chemical Information & Modeling*, 55(2):263–274, 2015.

[2] David L. Mobley, Karisa L. Wymer, Nathan M. Lim, and J. Peter Guthrie. Blind prediction of solvation free energies from the sampl4 challenge. *Journal of Computer-Aided Molecular Design*, 28(3):135–150, 2014.

[3] R. T. Mcgibbon, A. G. Taube, A. G. Donchev, K Siva, F Hernández, C Hargus, K. H. Law, J. L. Klepeis, and D. E. Shaw. Improving the accuracy of mϕller-plesset perturbation theory with neural networks. *Journal of Chemical Physics*, 147(16):161725, 2017.

[4] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–444, 2015.

[5] Hongming Chen, Ola Engkvist, Yinhai Wang, Marcus Olivecrona, and Thomas Blaschke. The rise of deep learning in drug discovery. *Drug discovery today*, 23(6):1241–1250, 2018.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *International Conference on Neural Information Processing Systems*, pages 1097–1105, 2012.

[7] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.

[8] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann Lecun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[9] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *NIPS 2015, Advances in neural information processing systems*, pages 2224–2232, 2015.

[10] David Weininger. Smiles, a chemical language and information system. *Journal of Chemical Information & Computer Sciences*, 28(1):31–36, 1988.

[11] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754, 2010.

[12] Grégoire Montavon, Katja Hansen, Siamac Fazli, Matthias Rupp, Franziska Biegler, Andreas Ziehe, Alexandre Tkatchenko, Anatole von Lilienfeld, and Klaus-Robert Müller. Learning invariant representations of molecules for atomization energy prediction. In *NIPS 2012, Advances in neural information processing systems*, pages 449–457, 2012.

[13] Fábio Fabris, Aoife Doherty, Daniel Palmer, João Pedro de Magalhães, and Alex Alves Freitas. A new approach for interpreting random forest models and its application to the biology of ageing. *Bioinformatics*, 34(14):2449–2456, 2018.

[14] Yu Bao, Morihiro Hayashida, and Tatsuya Akutsu. Lbsizecleav: improved support vector machine (svm)-based prediction of dicer cleavage sites using loop/bulge length. *BMC Bioinformatics*, 17:487:1–487:11, 2016.

[15] Kazunori D. Yamada, Satoshi Omori, Hafumi Nishi, and Masaru Miyagi. Identification of the sequence determinants of protein *N*-terminal acetylation through a decision tree approach. *BMC Bioinformatics*, 18(1):289, 2017.

[16] Zhenyun Deng, Xiaoshu Zhu, Debo Cheng, Ming Zong, and Shichao Zhang. Efficient knn classification algorithm for big data. *Neurocomputing*, 195:143–148, 2016.

[17] Pengfei Liu, Xipeng Qiu, Xinchi Chen, Shiyu Wu, and Xuanjing Huang. Multi-timescale long short-term memory neural network for modelling sentences and documents. In *Conference on Empirical Methods in Natural Language Processing*, pages 2326–2335, 2015.

[18] Kaisheng Yao, Geoffrey Zweig, Mei-Yuh Hwang, Yangyang Shi, and Dong Yu. Recurrent neural networks for language understanding. In *INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France, August 25-29, 2013*, pages 2524–2528, 2013.

[19] Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. Cnn-rnn: A unified framework for multi-label image classification. In *Computer Vision and Pattern Recognition*, pages 2285–2294, 2016.

[20] Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *Computer Vision and Pattern Recognition*, pages 4584–4593, 2016.

[21] Santiago Ndez, Alex Graves, J Schmidhuber, and rgen. An application of recurrent neural networks to discriminative keyword spotting. In *Artificial Neural Networks - ICANN 2007, International Conference, Porto, Portugal, September 9-13, 2007, Proceedings*, pages 220–229, 2007.

[22] Q. Zhou, P. Tang, S. Liu, J. Pan, Q. Yan, and S. C. Zhang. Learning atoms for materials discovery. *Proceedings of the National Academy of Sciences of the United States of America*, 115(28):E6411–E6417, 2018.

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

[24] Cao Xiao, Ping Zhang, W. Chaovalitwongse, Jianying Hu, and Fei Wang. Adverse drug reaction prediction with symbolic latent dirichlet allocation. In *AAAI 2017, AAAI Conference on Artificial Intelligence*, 2017.

[25] Shahar Harel and Kira Radinsky. Accelerating prototype-based drug discovery using conditional diversity networks. In *The ACM SIGKDD International Conference*, 2018.

[26] D Page, V. S. Costa, S Natarajan, A Barnard, P Peissig, and M Caldwell. Identifying adverse drug events by relational learning. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 790–793, 2012.

[27] Andrew Yates, Nazli Goharian, and Ophir Frieder. Extracting adverse drug reactions from social media. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2460–2467, 2015.

[28] Cao Xiao, Ping Zhang, W. Art Chaowalitwongse, Jianying Hu, and Fei Wang. Adverse drug reaction prediction with symbolic latent dirichlet allocation. In *Thirty-First AAAI Conference on Artificial Intelligence*, pages 1590–1596, 2017.

[29] Bo Jin, Haoyu Yang, Cao Xiao, Ping Zhang, Xiaopeng Wei, and Wang Fei. Multitask dyadic prediction and its application in prediction of adverse drug-drug interaction. In *Thirty-First AAAI Conference on Artificial Intelligence*, pages 1367–1373, 2017.

[30] Thomas G. Dietterich, Ajay N. Jain, Richard H. Lathrop, and Tomas Lozano-Perez. A comparison of dynamic reposing and tangent distance for drug activity prediction. In *NIPS 1994, Advances in Neural Information Processing Systems*, volume 6, pages 216–223, 1994.

[31] Manfred K. Warmuth, Gunnar Rätsch, Michael Mathieson, Jun Liao, and Christian Lemmen. Active learning in the drug discovery process. In *NIPS 2001, Advances in Neural Information Processing Systems*, volume 43, pages 1449–1456, 2001.

[32] Tengfei Ma, Cao Xiao, Jiayu Zhou, and Fei Wang. Drug similarity integration through attentive multi-view graph auto-encoders. In *IJCAI 2018, International Journal of Computational Intelligence and Applications*, 2018.

[33] Greg Landrum. Rdkit: Open-source cheminformatics. 2006. http://www.rdkit.org.

[34] Pau Rodríguez, Miguel Ángel Bautista, Jordi Gonzàlez, and Sergio Escalera. Beyond one-hot encoding: Lower dimensional target embedding. *Image Vision Comput.*, 75:21–31, 2018.

[35] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *INTERSPEECH 2010, Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September*, pages 1045–1048, 2010.

[36] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

[37] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[38] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[39] Heewon Park, Yuichi Shiraishi, Seiya Imoto, and Satoru Miyano. A novel adaptive penalized logistic regression for uncovering biomarker associated with anti-cancer drug sensitivity. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 14(4):771–782, 2017.