A Machine Learning Method for Material Property and Function Prediction: Example Polymer Compatibility

Methodology

There can be several perspectives to study compatibility prediction, such as parameters estimation mentioned above and direct compatibility classification. We find that compatibility related literature usually describe compatibility by compatible and incompatible. We use ML to implement compatibility prediction. For this purpose, we need to prepare a database to get a good predictive model. Following, we will introduce modules in our work flow: Data Collection and Information Extraction, Molecular Representation and Predictive Model. It's worth noting that these models are basically general and can be transferred to other problems just with minor adjustments.

1 Data Collection and Information Extraction Module

Although many researchers have conducted lots of compatibility studies and published their work, there is not a specific database constructed for polymer compatibility. To get our predictive model, we collect data by following means.

Database extraction Database Polyinfo is developed by National Institute for Materials Science (NIMS) [1]. It contains a number of polymer blend information and blend morphology information. Some entries have clear compatibility information which can be inferred from morphology description, such as miscible, compatible, incompatible and so on. We collect them and tag them with compatible and incompatible according to morphology description. We discard those cases where blend is partially compatible or description is ambiguous.

Text Data Mining We search and download papers related to keywords 'Compatibility' and 'Polymer' from Google Scholar [1] and Tsinghua University Library [2], which sum up to 47K articles. During these articles, some sentences contain clear compatibility information. For example, "Results of physical properties measurements reveal the blends of SR and FKM are technologically compatible" [3]. We design a filter to automatically export these sentences from articles. To achieve this goal, we design Information Extraction Model. Details are presented as following and also shown in Figure.S1

As mentioned above, although we have collected thousands of compatibility-related articles, it is still a great challenge to extract specific sentences containing clear compatibility description. Since these data are in the form of language text, we use ML for language to process them automatically. Natural Language Processing (NLP) is a such kind of AI technology and has shown good power. We use this technology on our literature texts, and extract all sentences containing information we need.

All sentences in literature can be divided into compatibility-related ones and compatibility-unrelated ones. We design an Information Extraction Module to achieve classification.

Information Extraction Module Firstly, we search for all sentences under a keyword root 'compati' as our potential corpus, randomly choose 2,000 sentences to establish database RawSen, and label them mutually according to their

[1] https://scholar.google.com

[2] https://lib.tsinghua.edu.cn/en

[3] https://doi.org/10.1177/0095244309345409

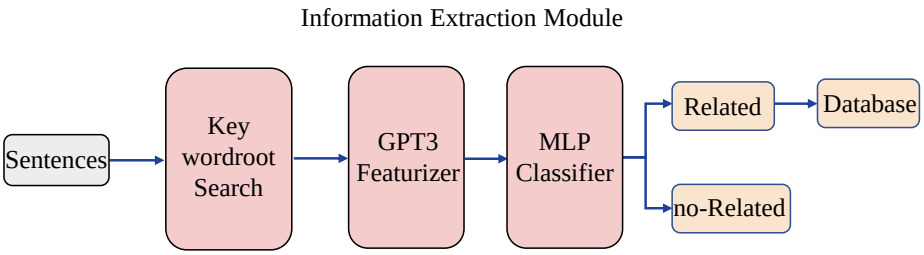

Figure S1: Information Extraction Module. All sentences are screened through this module and compatibility-related ones are collected to construct Database WholeSen.

meanings. Secondly, we transform all sentences to vectors through GPT3-featurizer. GPT3 (Generative Pre-Training) is an NLP generative model which has a great power in semantic representation [2]. It can characterize sentences by long vectors based on their meanings, and distance between vectors is related semantic difference. Thirdly, we pass these vectors to Multi-Layer Perception (MLP), to train a binary-classification screening network. MLP is comprised of linear function and nonlinear function and theoretically has the ability to fit a function. Whole computation process is described below:

$$h = W_0 x + b_0, \tag{1a}$$
$$x_1 = Relu(h) = max(0, h), \tag{1b}$$
$$y = W_1 x_{1+} b_1, \tag{1c}$$
$$\tag{1d}$$

where x presents input vector, y presents output label, and its value can be 0 (for no-related sample) or 1 (for related sample). The whole error of this network is evaluated by Cross-Entropy loss:

$$L_{IEM} = CrossEntropy(y, p) = \frac{1}{N} \sum_i -[y_i \log(p_i) + (1 - y_i) \log(1 - p_i)], \tag{2}$$

where p_i represents the probability of the sample i being compatibility-related.

Finally, we pass all sentences in our corpus through binary-classification screening network, and get the database WholeSen from literature.

2 Molecular Representation Model

To improve we should transform molecular structure into proper vector representation. We have transformed texts into vectors successfully, but molecular structure is far more complicated than simple text. We divide the representation process into two steps.

First, we represent polymer repeating unit structure with proper character string according to a specific rule, which has a strong relation with spatial information. For character string, we decide to use SMILES (Simplified Molecular Input Line Entry System) [3, 4, 5]. Although InChI is introduced as a standard for formula representation by IUPAC, SMILES is still generally considered to be more human-readable than InChI [6].

Second, we transform these strings to vectors according to chemical structures at different scales. There are some methods based on SMILES, such as RDkit Descriptors [7], MACCS Keys FingerPrint [8], PubChem FingerPrint [9] and Circular FingerPrint [10]. Among these methods, RDkit Descriptors present high-dimensional features and indicate molecular physical and chemical properties, while FingerPrint presents structure of molecules and indicate, for example, whether there is phenyl or not. We use Circular FingerPrint in our polymer compatibility network.The whole process can be shown as Figure.S2.

3 Predictive Model

Our model here mainly consists of three modules: Feature Extraction Module, Features Dense Module, Difference and Decision Module. The whole process can be shown as Figure.S3.

Feature Extraction Module Since molecular representations are always high-dimensional (for example, 2048-D for Circular FingerPrint) and chemistry theories focus much more on just a few factors (for example, 4-D for HSP),

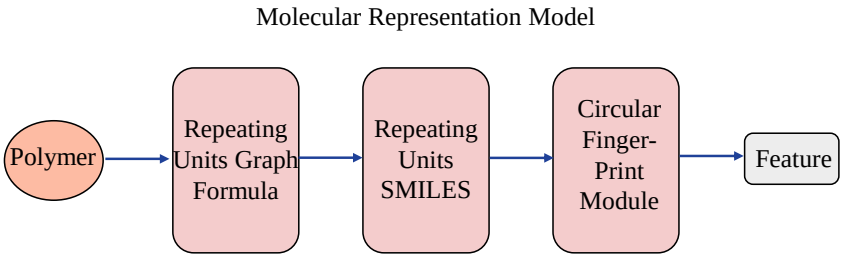

Figure S2: Molecular Representation Model. Polymers are processed through these stages and finally transformed to vector representations.

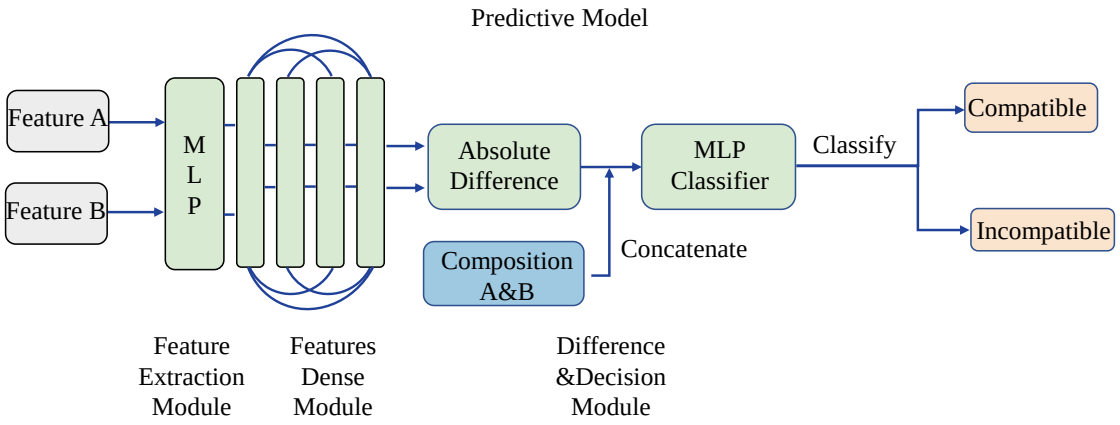

Figure S3: Predictive Model. Whole architecture includes three main modules, and composition are concatenated after Feature Dense Module. After Decision Module, model will finally output classification prediction.

we assume that many parameters in molecular representation are likely to be redundant. We use an MLP with linear connection layers and Sigmoid layers to reduce the features dimension. Each node in next hidden layer is a function of all nodes in the current layer, and represents a new feature made up of former factors. In this way, we can extract the proper features from initial input.

Features Dense Module We assume that not only some specific parameters such as HSP and components influence compatibility, but also basic structure features matter to compatibility, because the total interaction depends on atoms, chains and functional group of polymers. Therefore, our final polymer compatibility prediction depends on features at different depth level inside the network. Besides, existing researches also prove that connection between different hidden layers and shortcut of network will improve the representation ability and avoid over-fitting [11, 12]. Therefore, we construct dense shortcuts that connect different layers in order to make our model learn the right rules. Features Dense Module integrates different depth features and these features will participate in the final prediction together. The

algorithm is presented as follow:

$$h_1 = W_0 x + b_0, \tag{3a}$$
$$x_1 = Sigmoid(h_1) * \lambda, \tag{3b}$$
$$h_2 = W_1(x + x_1)_+ b_1, \tag{3c}$$
$$x_2 = Sigmoid(h_2) * \lambda, \tag{3d}$$
$$h_3 = W_2(x + x_1 + x_2)_+ b_2, \tag{3e}$$
$$x_3 = Sigmoid(h_3) * \lambda, \tag{3f}$$
$$Sigmoid(x) = \frac{1}{1 + e^{-x}}, \tag{3g}$$

where x is feature vector after Feature Exaction Module, W_i is weight matrix, b_i is bias vector, and λ is a parameter to control the training loss (λ is set as 10 in actual experiments).

Difference and Decision Module According to F-H theory and HSP theory, if polymer molecules have similar structures and HSP parameters, F-H interaction parameter χ will be small and ΔG_M will be negative, so the blends can be compatible. Therefore, we use the difference between two vectors obtained after Features Dense Module as the inputs of Decision Module. We don't care about the sign so we calculate the absolute value of the difference and pass the absolute difference vector to followed layers.

At the same time, we notice that in fact, the composition will obviously influence the polymer blend compatibility. It is common that blend are more compatible at 10%-90% composition than at 50%-50% composition. We concatenate the difference vector and composition information, and put the whole vector into a 3-layers MLP, which will give the final prediction. As to our labels, 0 corresponds to compatible and λ (set to 10 in experiments) corresponds to incompatible. Therefore, if the final output is closer to 0, it means our model predicts "compatible", while output closer to λ means our model predicts "incompatible". Although our problem is a classification problem, we use MSE (Mean Square Error) loss to finetune our network.

References

[1] Shingo Otsuka, Isao Kuwajima, Junko Hosoya, Yibin Xu, and Masayoshi Yamazaki. Polyinfo: Polymer database for polymeric materials design. In *2011 International Conference on Emerging Intelligent Data and Web Technologies*, pages 22–29, 2011.

[2] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[3] David Weininger. Smiles, a chemical language and information system. 1. *Introduction to*, 1970.

[4] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, 1988.

[5] David Weininger. Smiles. 3. depict. graphical depiction of chemical structures. *Journal of Chemical Information and Computer Sciences*, 30(3):237–243, 1990.

[6] Stephen R Heller, Alan McNaught, Igor Pletnev, Stephen Stein, and Dmitrii Tchekhovskoi. Inchi, the iupac international chemical identifier. *Journal of Cheminformatics*, 7(1):1–34, 2015.

[7] Greg Landrum. Rdkit documentation. *Release*, 1(1-79):4, 2013.

[8] Joseph L Durant, Burton A Leland, Douglas R Henry, and James G Nourse. Reoptimization of mdl keys for use in drug discovery. *Journal of Chemical Information and Computer Sciences*, 42(6):1273–1280, 2002.

[9] Xiang-Qun Sean Xie. Exploiting pubchem for virtual screening. *Expert Opinion on Drug Discovery*, 5(12):1205–1220, 2010.

[10] Robert C Glen, Andreas Bender, Catrin H Arnby, Lars Carlsson, Scott Boyer, and James Smith. Circular fingerprints: flexible molecular descriptors with applications from physical chemistry to adme. *IDrugs*, 9(3):199, 2006.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.