NLP Application on Identifying the Gene Phenotype

Bin Li, Chen Song, Zhenyu Zhao

Overview

Other than some traditional applications of NLP, such as analysis of the medical record to track a patient's condition, NLP is now more and more used in understanding the semantic meaning of gene information. Languages such as English and Japanese use sequences of words to encode complex meanings and have complicated rules. In the bioinformatics area, different language structures and content also imply biological substances differently. Our project is to build a system to predict the phenotype of a gene based on reading semantic information of it. To achieve this goal, we study ex-works in virus mutants, select training models, run the training, and improve the performance by revising the structure and codes. Finally, we reach our goal. The accuracy we reach is considered reliable in the detection area, similar to the former work's accuracy in the virus.
The project will be organized as follows. We will briefly go through the former work, then introduce our system, training data, training model, effort to improve our work's performance, and limitations.

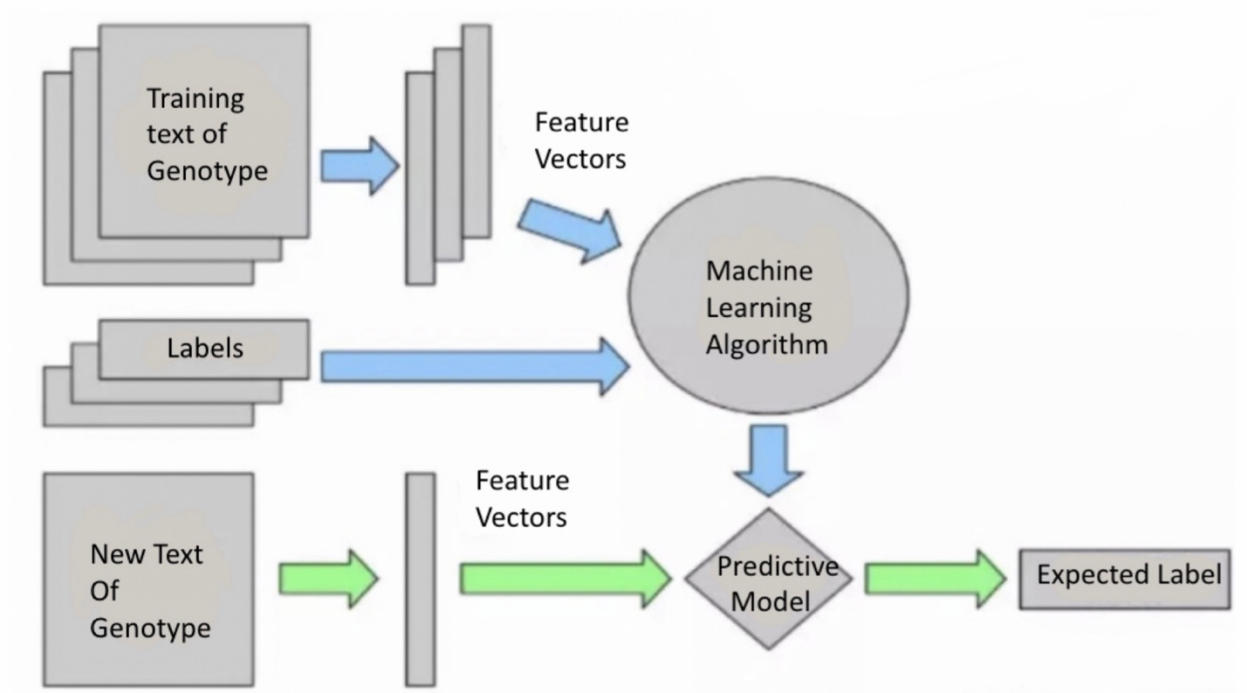
Former work

The global pandemic began over one year ago. The biggest challenger for the world to deal with is RNA virus mutates very fast. However, it's hard for our scientists to identify every mutants' infectivity by biological identification. In a recent study, the escape and evolution of the COVID-19 virus are represented as 'semantic' and 'grammaticality' changes in the NLP model. By learning the virus's 'semantic' structure, the researchers can define the relations between the content information and biology information of the virus. This method will help government more easily to be aware of dangerous variants.

System Model of Our Project

The goal of our project is to predict the phenotype based on semantic information of the genotype. We use labeled training text into different algorithms, then achieve the predictive model. Then the performance of the system is valued by

training data. The training algorithms and training data information will explain in the subsequent two sessions. The system model is below.



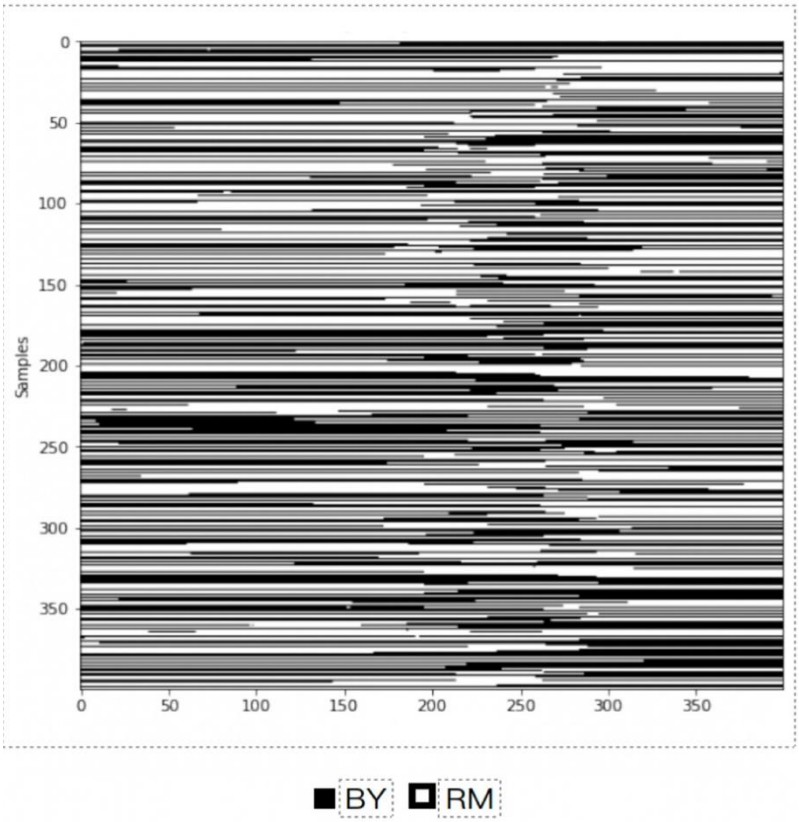
System model of Predict Phenotype of Gene

Training Data

The dataset used in our project is the popularly used yeast data, which represents a scenario that the genetic background is simple, and the genotypes are highly correlated. This yeast genotype dataset contains the genotype profile of 28,820 unique genetic variants obtained by sequencing 4390 observations from a cross between two strains of yeast: a widely used laboratory strain (BY) and an isolate from a vineyard (RM). The original data fields in the yeast genotype profile were encoded as -1 for BY and 1 for RM. We studied the single phenotype named '1-Diamide-1'. The original value for this single phenotype is a continuous numerical measurement. To complete the prediction task, we label those samples whose phenotype measurements are greater than the statistical mean value and the rest of the samples as 0.

Part of the raw data is attached below. It is noticed that CNN model, regression model can learn the correlation among samples, which is shown as the relationship

among rows. But the RNN can extract the internal pattern within a sample (the information in a line in the picture). We want to test whether or not the internal pattern can improve the classification performance.

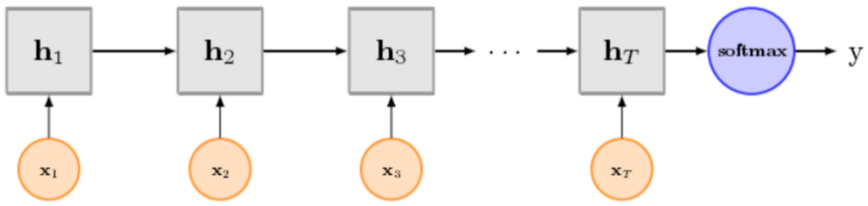


The visualization of genotype profile of the yeast data. BY represents genotypes from a laboratory strain and RM stands for genotypes from a vineyard strain.

Algorithm Testing

We train with the following machine learning algorithms: CNN, Deep CNN, RNN, and Federate Learning.

TextRNN

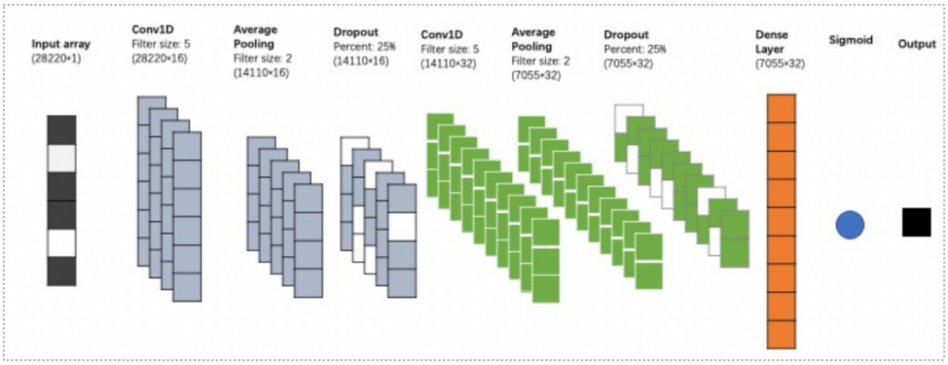


TextRNN Model

The TextRNN is known as suitable for sequence structure. In the first place, we thought it might be a good choice for genotype training. However, it takes 50 minutes for a single epoch when runs the training process, which is dramatically slow. We tried different computers and cloud servers, still the same problem. So, we decided not to go further with only TextRNN. We also don't have RNN training result.

TextCNN

We first tested CNN and Linear model. Results showed that CNN (accuracy of 78%) works better than linear model (accuracy of 73%). So, CNN is used as the first layer of our model.



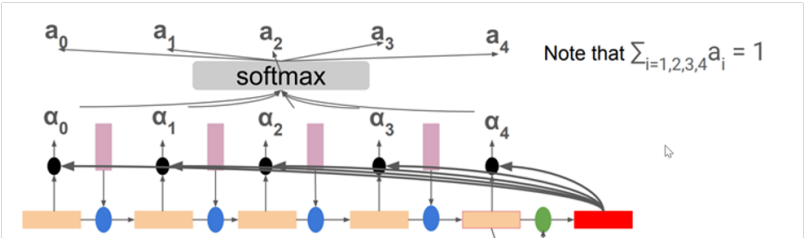
The structure of the deep learning model

For TextCNN, we begin with the CNN and the Deep CNN. We used 1-D convolutional neural network in our model. To avoid the overfitting, we added a dropout layer with 25% dropping rate behind the CNN-Pooling structure.

Attention to RNN

The normal RNN treats all of its hidden states equally, which may overweight the unimportant feature but ignore those critical features. So, we added an attention block after the RNN cell. Attention is to give different hidden state weights, which can represent their importance to the results. The ideal weights should focus on

those features that contribute more to the output and should not disturb the model.



Attention mechanism of a RNN cell

So, we employed a soft attention query to learn the importance of each hidden state, and then we summarized those weighted hidden states by using the mean value of them.

In detail, we used a tanh nonlinear function to learn the weights matrix, and we used a softmax layer to constrain the weights matrix so that the weight sum of each step can be limited in 1 and will not disturb the model.

For a hidden states vector

$$H = [h_1, h_2, \ldots, h_n]$$

where n is the total step of RNN cell, the attention mechanism works as

$$Q = tanh(H)$$

$$a = softmax(w^T Q)$$

$$o = Ha^T$$

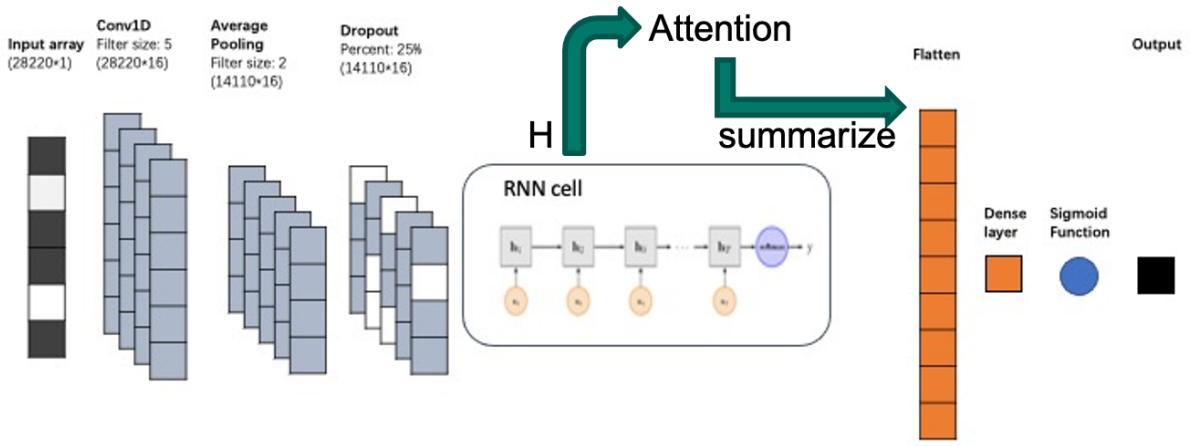
$$output = mean(o)$$

In this way, the RNN could learn the importance of each feature and then pay more attention to critical hidden states.

There are three RNN cells for us to use: simple RNN, Gated recurrent units (GRU), and LSTM (long short-term memory). The simple RNN is oversimple so that it will forget information that has been learned before. And the GRU is not stable. So, we used LSTM in the end. We used a Bi-LSTM to study both the forward sequence relationship and the backward relationship in a sample to learn more patterns. There is one parameter we need to decide, which is the number of the hidden unit. After 10-folds cross-validation, we used 80 as the number of the hidden unit, which performs well and saves computation resources.

Add RNN into the Structure

To reach better performance, we can put RNN into CNN structure, as known as RCNN. We tried adding it in the first layer of the model; however, the same as the single RNN comes up, the training becomes extremely slow and nearly needs an hour to finish training a single epoch. Then we add it at latent space in the CNN model and the system model attached below.



System model of RCNN

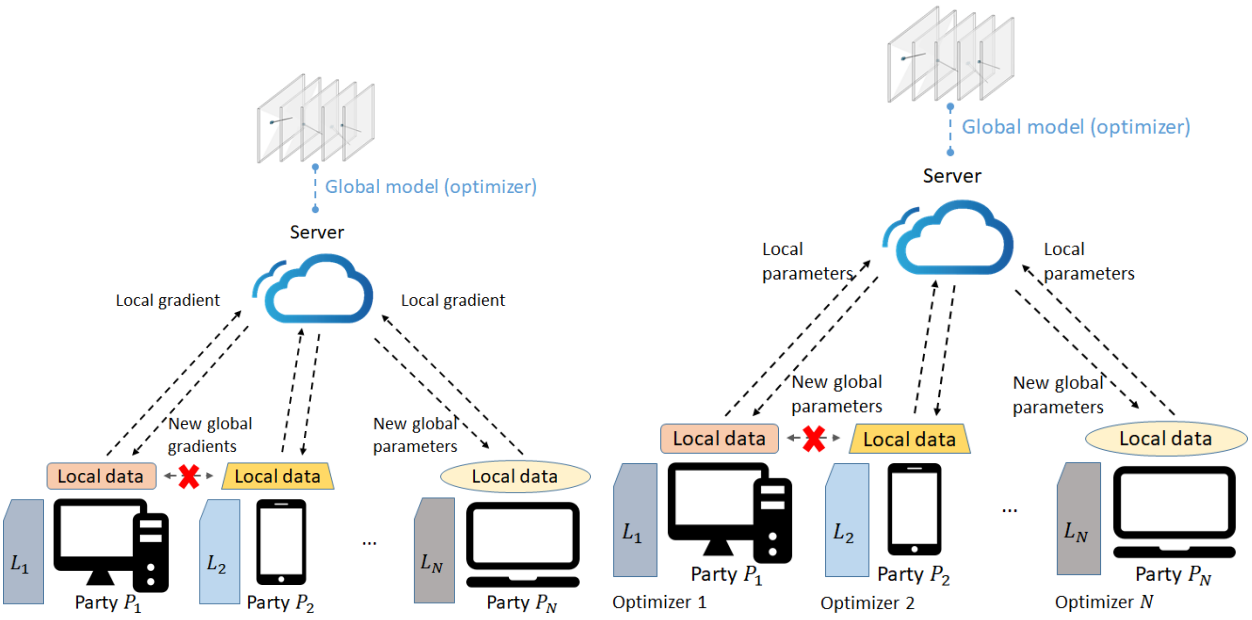
It is shown that a raw genotype data will be fed into a 1-D convolutional neural network at first. Then, we used a RNN cell to extract the internal pattern of them with an attention mechanism. The output will used for classification in the end. In the model, there are four parts needing to be training: the CNN model, the RNN model, the Attention weights and the final classifier.

Outcome of TextCNN

Algorithm	CNN	Deep CNN
Accuracy (%)	78.7	81.83

The accuracy reaches 81.83%, not good enough for predicting the phenotype.

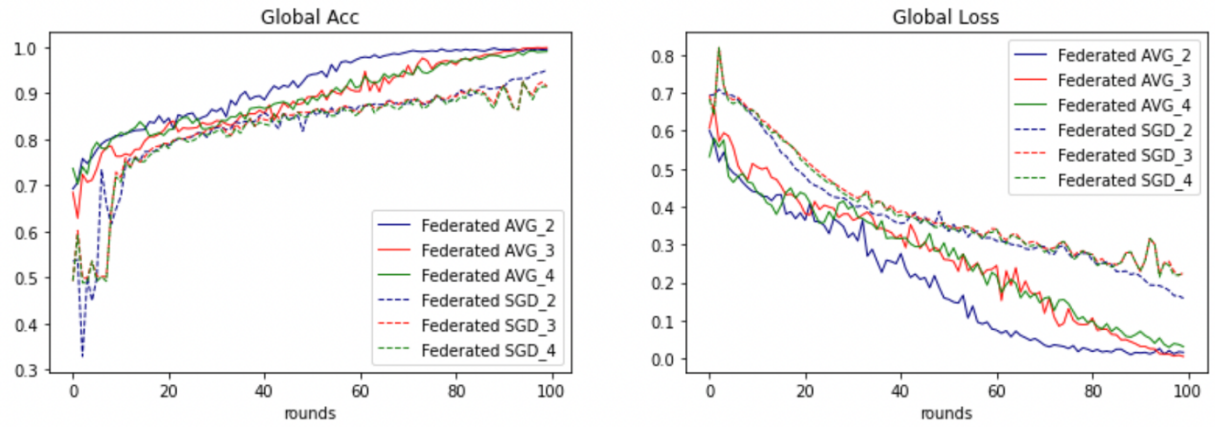
Federate Learning



FedSGD Model FedAVG Model

Federate learning is popularly used in biological area. Federate learning is a great tool for training privacy-sensitive data as it doesn't require users to concentrate data in the data center. The first one is Federate learning version of SGD. The second is Federate Averaging. Fed AVG is based on FedSGD, but multi-step cumulative gradients can be trained on the device, increasing the single-round calculation on each device.

Outcome of Federate Learning



Training Outcome comparison

Algorithm	FedSGD	FedAVG
Accuracy (%)	82	88

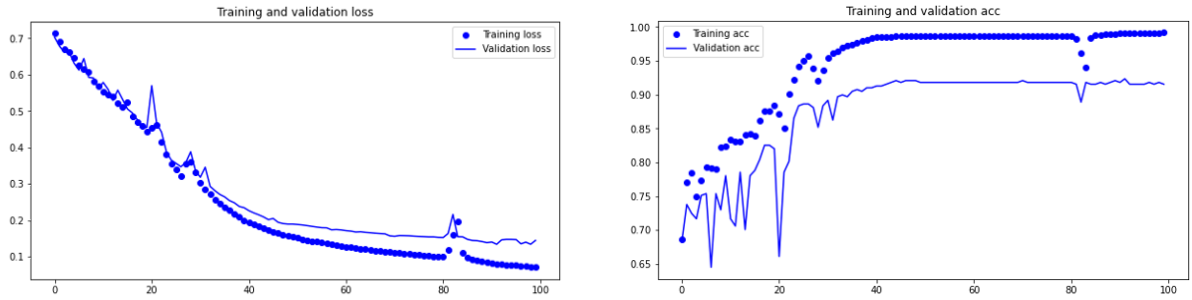
The Federate learning Averaging reaches higher accuracy than Federate SGD, as 88%, considered a high accuracy.

Work to improve Performance

After conduct training on different learning model, the highest accuracy we reach is 88%. Obviously, we still have room to improve our performance, and we will work on the TextCNN model.

Implementation of the model

We implemented the model on Keras platform. In detail, we used adam optimizer with learning rate of 1e-3 to update parameters. The batch size for training is 128. Then we ran one round of training for 100 epochs.



Training curve for 100 epochs (CNN+LSTM attention model)

It is noticed that the converging time of proposed model is very long, because there are four parts that need to train.

Final Result

Algorithm	CNN	Deep CNN	CNN-LSTM	CNN-LSTM+Attention	Federate learning (LSTM)
Accuracy (%)	78.76	81.83	84.45	91.82	88.31

The revised CNN structure, which is CNN-LSTM+Attention reaches the best performance.

Limitation

The first limitation we have is a common problem in NLP-based detection in the biological area. Compared with conventional biological detection, NLP-based methods are more like identification, and the accuracy level is way lower than testing. Compared with the accuracy rate of biological detection, which is close to 100%, our accuracy rate is only 91%. Such an accuracy rate can be considered reliable in the field of recognition but not in the field of testing.
The other limitation is when it comes to a specific phenotype, the system needs one more training for the system.
The attribute difference with biological detection makes the use of text-based recognition and biological detection different. The detection speed based on text is much faster than biological methods, so more applications based on NLP recognition provide scientists with an indicator when large-scale data analysis is needed. The final accurate detection still depends on biological methods. If the accuracy of NLP-based detection can be improved to nearly 99%, It will be a new page of biological history, and we do believe it is possible.

Conclusion

After going through the machine learning algorithm selection and solving the problems in the actual training, we finally achieved our expected goal and achieved the expected accuracy rate. Our project can predict the phenotype of the gene through textual information. This is a very novel application of NLP that we tried because there are no existing research results for genotype. Our project is a significant study, for example, in large-scale gene breeding and planting, the

phenotype of the gene can be known without testing one by one, reducing a lot of the burden in actual production.
After a semester-long study of AI, we get a better complete understanding of AI. During doing our project, we understand more about difficulties in AI research and application. We hope we can see more effective NLP application in the biological area to help more researchers and contribute more to society.

References

[1] Momcheva G . Sentiment detection with FedMD: Federated Learning via Model Distillation[C].
Information Systems and Grid Technologies. 2020.
[2] Zhu X , Wang J , Hong Z , et al. Empirical Studies of Institutional Federated Learning For Natural
Language Processing[C]. Findings of the Association for Computational Linguistics: EMNLP 2020.
2020.
[3] Hie B , Zhong E D , Berger B , et al. Learning the language of viral evolution and escape[J]. Science,
2021, 371(6526):284-288.
[4] Joshua S Bloom, Iulia Kotenko, Meru J Sadhu, Sebastian Treusch, Frank W Albert, and Leonid
Kruglyak. 2015. Genetic interactions contribute less than additive effects to quantitative trait variation in
yeast. Nature communications , Vol. 6 (2015), 8712.
[5] Sequence to Sequence Learning with Neural Networks. Sutskever et al., 2014.