Description of Solution

AILabs@tw

Data

The training dataset of Stage 2

The training dataset of Stage 1 (n=3,321) and the filtered testing dataset of Stage 1 (n=368) were combined as the training dataset of Stage 2 (n=3,689).

The testing dataset of Stage 2

Among the released testing dataset of the stage 2 (n=986), 809 data points already appeared in the testing dataset of stage 1. For the rest 177 data points, a classifier—built upon testing dataset of the stage 1 (n=5,668) to differentiate between real data points and machine-generated data points—was used to find the real testing data points in the stage 2. As a result, 150 data points were identified by our classifier as real testing data, and were used as our testing dataset of Stage 2 (n=150).

Besides, we discovered that the identified real testing dataset (n=150) consists of only the variation type of point substitutions according to the variation names (e.g., M224R, explained in the next session). Therefore, a special subgroup of the training data whose variation name is also identified as the type of point substitution was extracted as “point substitution training subset” (n=2,925) from the released training dataset of Stage 2, and was used when training one of our raw models (details described in Table 2).

Pipeline

1. **Preprocessing**

   A heuristic parser to identify the formats of variations, such as truncating mutations, ins, del, point substitution, was constructed. Also, we converted three-letter amino acid codes found in the texts of some data points into one-letter codes (e.g., Met224Arg to M224R).

2. **Feature extraction**

   Several features were extracted based on the three characteristics of data (i.e., gene, variation and text), as described in Table 1. Besides the features extracted by traditional natural language processing, **we designed several novel features (marked by bold-faced in Table 1) either from domain knowledge or statistical reasoning**.

3. **Classifier**

   XGBoost and hyperparameters are described in the section of Classifier Settings.

4. **Model switching/averaging**

We consolidated multiple models to avoid overfitting and solve the missing data problem by two different methods, as described in Model Switching/Averaging.

Table 1: Features of gene, variation, and text

	Feature	Dimension	Description	Feature index
GENE	Pretrained vector representation of gene	200	Vector representations of genes (word2vec) were induced from PubMed and PMC texts. (External data usage from biomedical natural language processing http://bio.nlplab.org)	1
VARIATION	One hot encoding for variation formats	28	Variation formats, such as truncating mutations, ins (e.g., Q58_Q59insL), del (e.g., S459del), point substitution (e.g., M224R), were encoded using a one-hot aka one-of-K scheme. There are 28 variation formats encoded in total.	2
	One hot encoding for point substitutions	41	For every point substitution (e.g., M224R), the beginning and ending amino acids (e.g., M and R) were encoded by one-hot (20 different possible amino acids for each), and the mutation site (e.g., 224) was encoded numerically. For other variation formats, this feature was treated as missing.	3
	Variation class distribution for point substitutions	9	For every point substitution (e.g., M224R), we searched all the data points in the available training set to find those with the same beginning amino acid and mutation site (in this case, M224). If some data points were found, the variation class distribution was calculated accordingly; otherwise, this feature was treated as missing. For other variation formats, this feature was treated as missing.	4
	Vector representation of variation	100	Vector representations of variations (word2vec) were induced from the texts of all data points (the training and testing datasets of Stage 2) by fastText.	5

TEXT	tf-idf → SVD	200	tf-idf from the texts (documents) of all data points (the training and testing datasets of Stage 2) was computed and then SVD was used to reduce the number of dimensions from 200k (terms) to 200 (principal components).	6
	Keyword frequency	120	Texts were grouped according to their labels into 9 meta-texts. Keywords were then identified by tf-idf of the 9 meta-texts. (Some keywords were added heuristically.) For every data point, frequencies of these keywords (and the negation of them) in the text were calculated. For example, “likely” was identified as a keyword from tf-idf, then we computed the frequencies of “likely” and “not likely” in the given text.	7
	Text class distribution	9	For every data point, we computed “class distribution of the text." Specifically, several segments (each with 200 characters) of the text were extracted. We then used the extracted segments to search all the data points in the available training set to find those with the same segments. If some data points were found, the class distribution was calculated accordingly; otherwise, this feature was treated as missing. (We handled this issue in the Model Switching/Averaging section.)	8
	Vector representation of text	100	Vector representations of texts (doc2vec) were induced from the paragraphs containing the variation name by gensim. (For every data point whose variation name cannot be found in its text, 0-vector was adopted.)	9
	Neighbor class distribution	9	For every data point, we found the paragraphs that contain the associated variation. All other variations in these paragraphs are called neighbors of the data point. If some neighbors were found, their class distribution was calculated according to the corresponding labels; otherwise, this feature was treated as missing.	10

Classifier Settings

XGBoost classifiers with various combinations of the aforementioned features were built with 5-fold cross validation (for each fold, validation set is about 15% of the training dataset of stage 2). Some critical parameters were set as follows: Max_depth = 6~8, Subsampling = 0.8, Min_child_weight = 2, and early stopping was adopted.

Results

Various combinations of features were considered to build classifiers. Generally speaking, the average of 5-fold cross validation errors (logloss) was around 0.73.

Surprisingly, one feature highly correlated with the class label was identified. If the classifier was built with the feature "text class distribution," the average of 5-fold cross validation errors could be further lowered to around 0.68. However, this feature has a missing data problem among ¼ of all the training data (and missing among ⅔ of the testing data). We will address this issue in the next session.

Model Switching/Averaging

Table 2 summaries the four major models we built. (For each Raw model, the average of 5-fold predictions was used as the predicted probabilities at the stage of inference.)

Table 2: Models

Model	Description	Features used (corresponding to the feature index in Table 1)
Raw model 1	With text class distribution feature, trained on all stage 2 training dataset	1, 2, 3, 4, 6, 7, 8
Raw model 2	Without text class distribution feature, trained on all stage 2 training dataset	1, 2, 3, 4, 6, 7
Raw model 3	Without text class distribution feature, trained on all stage 2 training dataset	1, 2, 3, 4, 5, 6, 7, 9, 10
Raw model 4	Without text class distribution feature, trained on "Point substitution training subset"	1, 3, 4, 5, 6, 7, 9, 10

To handle the problem of missing data for the “text class distribution” feature, model switching and model averaging were used to generate predictions for the testing dataset of Stage 2.

Model switching

For a given testing data point, if its “text class distribution” was available, we use the prediction from Raw model 1; otherwise, we use that from Raw model 2.

Model averaging

For a given testing data point, the arithmetic mean of predictions from four Raw models is used.