CS5780 Final Project

Kaggle team name: Meow_Meow
Score on the public leaderboard: 0.81666

Kaggle screen name (Name, NetID):
Sarah(Chengyan Zhan, cz336),
flying_meow(Hanjun Jiang, hj447),
yyyyyy(Yan Yang, yy642)

December 2018

1 The preprocessing techniques used

For each tweets, three pre-processing steps are performed:

- convert all upper case to lower case
- word stem via Porter stemming algorithm using Natural Language Toolkit in Python [1]
- remove non-English characters or symbols

2 What you learned while exploring the data

One thing we noticed is that the best validation error we got was from SVM with rbf kernal with $C = 1.74$, which is a very small value. It means the model is very loose, and the solution is simpler, thus reduce the overfitting issue.

3 How you selected your model

We selected ERM, Naive Bayes, and Kernel SVM as well as ensemble algorithms by using bagging and boosting technique as our models. We averaged the models with the lowest error rate, including random forest, boost trees, RBF kernel SVM, and ERM with hinge loss and ridge loss. We choose these models because they are suitable for the size of given data set to avoid overfitting.

4 What features you extracted

Three kinds of feature extraction methods are implemented:

- One word bag
- Two words bag
- One and Two word bag

In order to select the most word and/or two word combinations, three dictionaries are built for the training set with label $y \pm 1$, training set with label $y = +1$, and training set with label $y = -1$, respectively. The key of the dictionary is the word or word combination, while the value is how many times it appears in the corresponding sets. After this, for each key in the dictionary for $y = \pm 1$, we compute the importance of the key, I(key) defined as:

$$\mathrm{I(key)} = \frac{\mathrm{dict}_{\mathrm{y}=+1}(\mathrm{key})/|\mathrm{dict}_{\mathrm{y}=+1}| - \mathrm{dict}_{\mathrm{y}=-1}(\mathrm{key})/|\mathrm{dict}_{\mathrm{y}=-1}|}{\mathrm{dict}_{\mathrm{y}=\pm 1}(\mathrm{key})/|\mathrm{dict}_{\mathrm{y}=\pm 1}|} \tag{1}$$

where $|\mathrm{dict}_{\mathrm{y}=\pm 1}|$ is the total number of keys, and $|\mathrm{dict}_{\mathrm{y}=\pm 1}(\mathrm{key})|$ means the number of times the key appeared. With a threshold, we select a list of important key with I(key) < threshold.
The tweet time for each tweets is also extracted to be one of the features, and is converted to an list with length of 2, the first element represents the day of the week, and the second element represents the time in a time. The final feature dimension is the length of the selected keys, plus two for tweets time.

5 How you searched for hyper-parameters

We used random search to find a good combination of hyper-parameters. Initially several combinations of hyper-parameters are generated randomly and evaluated, and the three of them with the lowest cross-validation errors are selected as seeds. Starting from each seed, run a trajectory, which means searching in the vicinity of the current combination of hyper-parameters (initially the seed) until either a smaller error is found or the number of points evaluated reaches a threshold. If a smaller error is found, the vicinity region is slightly shrunk and re-centered around the point with this smaller error, and the shrunk region is searched. If the number of points evaluated reaches a threshold, the vicinity region is shrunk by almost a factor of 2 in each dimension and the shrunk region (with the same center) is searched. This process is repeated iteratively, so we call it a trajectory. The three trajectories all end after a pre-defined number of evaluations is reached. Then the best combination of hyper-parameters so far is set as the seed, and a final trajectory is run, starting with a small vicinity region. After another pre-defined number of evaluations is reached, the best among all evaluated points is returned.

References

[1] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit.* O'Reilly Media, Inc., 2009.