My name: Yuankun Zhu

Display name: Jenny

The best result is created by using TfidfVectorizer and SVM.

The best performance: accuracy = 0.788870.

The following are features/tools I have used during my exploration:

1. Split train and validation data.

   I have tried spliting the train.tsv into training data and validation data parts to have some understanding about my model performance.

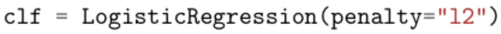
   ```
   train_x, valid_x, train_y, valid_y =

   model_selection.train_test_split(trainDF['text'], trainDF['label'])
   ```

   This part will not influence the final result so much, so I didn't use it when I generate my best performance in order to have more training data.

2. Classifiers.

   After generating the feature matrix for my training data, I have tried different classifier such as SVM and LogisticRegression to fit on the training data and to predict labels of the testing data. For the logistic regression part, I used the code `clf = LogisticRegression(penalty="l2")` . For this part, I got the accuracy 0.74812. This is a large improvement compared to my previous trials, but it is still lower than that of LinearSVC, which will result the accuracy 0.788870.

3. Tune parameters.

   I mainly tune the parameters for TfidfVectorizer and svm.LinearSVC. For TfidfVectorizer, I have changed the ngram_range to (1,2), (1,3) and so on. They have accuracy 0.78824, 0.78870 respectively.

   For svm.LinearSVC, I have changed the "penalty" to "l1" or "l2" (default is l2). Also, I have changed the penalty parameter C of the error term to 0.1, 1.0, 1.5, 2.0, 10.0 and so on. Finally, I have found out that I should choose penalty to be the default l2 and C = 2.0 to reach a relative high performance 0.78870.

APPENDIX:

```
import pandas as pd
from sklearn import svm
from sklearn.feature_extraction.text import TfidfVectorizer



TRAINING_DATA = "../train.tsv"
TESTING_DATA = "../test.tsv"
OUT_FILE = "../result.csv"

train_data = pd.read_csv(TRAINING_DATA, sep='\t')
labels = list(train_data['label'])
texts = list(train_data['text'])

trainDF = pd.DataFrame()
trainDF['text'] = texts
trainDF['label'] = labels

# ngram level tf-idf
vec = TfidfVectorizer(ngram_range=(1, 3))
train_matrix = vec.fit_transform(trainDF['text'])
# train_x, valid_x, train_y, valid_y = model_selection.train_test_split(trainDF['text'], trainDF['l

print("Done")

test_data = pd.read_csv(TESTING_DATA, sep='\t')
test_texts = test_data['text']

test_matrix = vec.transform(test_texts)
```

```
clf = svm.LinearSVC(C=2.0)
clf.fit(train_matrix, labels)
predicted_labels = clf.predict(test_matrix)
# print(predicted_labels)
# clf = LogisticRegression(penalty="l2")
# clf.fit(train_matrix, labels)
# predicted_labels = clf.predict(test_matrix)

with open(OUT_FILE, 'a', newline='') as f:
    writer = csv.writer(f)
    writer.writerow(["Id", "Category"])
    for idx, tag in enumerate(predicted_labels):
        csv_list = []
        csv_list.append(idx)
        csv_list.append(tag)
        writer.writerow(csv_list)
```