THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

COMP5423 NATURAL LANGUAGE PROCESSING

Lab1 Homework: Emotion Classification

Written Report

Fung Che Hei 19013111G

Content

1. Introduction

The aim of this individual project is to implement an classification system to identify emotion with the given English sentence. For simplification, only six emotions are used in this project which are {joy, love, surprise, fear, anger, sadness}.

2. Programming language and Environment

Python 3.8 is the main programming language used for implementation. Also, HTML is used for implementing templates of web application. In the whole project, Pycharm is used as the IDE (Integrated Development Environment) for implementation.

3. Data set information

The data set used in this project is downloaded from the Kaggle database. In the data set, there are 16000 training data w/ labels, 2000 validation data w/ labels, and 2000 test data w/o labels. There are six classes for the labels, which are **joy, love, surprise, fear, anger, and sadness**. The numbers of each class in training data set are as follows:

Class	Joy	love	surprise	sadness	fear	anger
Number	5362	1304	572	4666	1937	2159
Index	1	2	3	-1	-2	-3

4. Data cleaning and processing

All the training data, validation data and test are first processed with a function called text_processing( ) in python file text_processor.py. In this function, any digits, punctuations are first removed from the input sentence. Then all the words in the sentence are transformed into lower case. The sentence is then be striped such that any space notation (i.e. \n in Mac) after a sentence is removed. Next, the sentence is tokenized and all tokens are labeled with a POS tagging. After that, all the stop words and noun phrases of the sentence are removed, because stop words and noun phrases are not considered to provide much information in classifying emotion. The final step is to lemmatize all the tokens into their original form. Lemmatization is used instead of stemming in emotion classification task because the morphological meaning of a word is considered to be important. For example, a stemmer will consider ‘good’ and ‘well’ as two different words while a lemmatizer will consider they are the same. After lemmatization, the input sentence is said to be processed.

5. Features extraction

After data cleaning step, the processed texts are passed to a function called extract_features( ) in python file feature_extraction.py. There are totally **6006** features with seven criteria designed for extraction information from the processed input text.

(1) The first feature criteria is TFIDF and it returns **6000** features with the most frequently occurred 6000 unigrams and bigrams that appeared in training data set. A **TfidfVectorizer** object is used from the module called **feature_extraction.text** inside **Sklearn** library. The Tfidfvectorizer is first fitted and trained with training data in python file vectorizer.py, and saved as trained_vectorizer.sav for the usage of transforming the future raw data into TFIDF vectors.

(2) The second feature criteria is word count and it eventually returns the word count in a given sentence. This feature bases on a assumption that human tends to write more or less when he or she is under a certain emotion.

(3) The third feature criteria is to decide whether a sentence is positive (1) or negative (-1). It is make sense to believe that a positive sentence always written with a emotion of joy or love; while a negative sentence always written with a emotion of sadness, fear or anger. *****a NLTK built-in pertained sentiment analyzer, VADER is used for the determination

(4) The forth feature criteria is polarity score and it eventually returns the polarity score calculated by a NLTK built-in pertained sentiment analyzer, VADER. Similarly to (3), a sentence with emotion of joy or love tends to give higher polarity score; while a sentence with emotion of sadness, fear or anger tends to give lower polarity score.

(5) The fifth feature criteria is the presence of the prefix ‘un’ and it returns the count of word with the prefix ‘un’, but in negative sign. This is because words with prefix ‘un’ usually have a negative meaning and tend to give a emotion of sadness, fear or anger.

(6) The sixth feature criteria is the presence of the prefix ‘dis’ and it returns the count of word with the prefix ‘dis’, but in negative sign. Same as (5), words with prefix ‘dis’ usually have a negative meaning and tend to give a emotion of sadness, fear or anger.

(7) The seventh feature criteria is the presence of the synonyms and it returns the index value of that emotion class. It is believed that in order to describe our feeling or emotion clearly, we have to use appropriate words. For example, if I am feeling fear, I would use ‘scare’ or ‘horrible’ to express my feeling but won’t use ‘happy’ or ‘lovely’. So in this feature, it tries to find words or the synonyms of a particular emotion. For example, if a sentence contains lots of words related to the emotion ‘joy’, this feature should return ‘1’, which is the index of emotion ‘joy’.

6. Machine Learning Model

Random Forest is used as a classier for this emotion classification task.

7. Evaluation and Discussion

Validation data set with 2000 inputs and labels are passed to the trained machine learning model for evaluation. The results are as follow:

```
(venv) (base) TOMOKInoMacBook-Pro:lab1_homework tomoki$ python emotion_classification.py
classification report:
              precision    recall  f1-score   support

          -3       0.83      0.64      0.72       275
          -2       0.79      0.66      0.72       212
          -1       0.71      0.83      0.77       550
           1       0.77      0.83      0.80       704
           2       0.76      0.59      0.66       178
           3       0.84      0.75      0.79        81

    accuracy                           0.76      2000
   macro avg       0.78      0.72      0.74      2000
weighted avg       0.77      0.76      0.76      2000

(venv) (base) TOMOKInoMacBook-Pro:lab1_homework tomoki$
```

Remarks: {'anger': -3, 'fear': -2, 'joy': 1, 'love': 2, 'sadness': -1, 'surprise': 3}

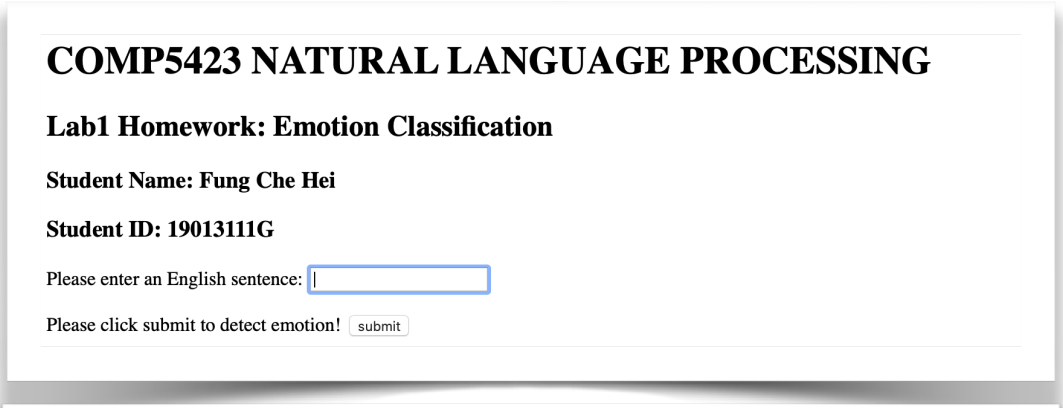
The model can achieve 76% of the overall accuracy. When looking at precision, recall and f1-score, it is found that although we have only 572 training data which are labeled as surprise (index 3) among the total 16000 training data, our classifier still classify well in sentence with emotion surprise, with 84% precision, 75% recall and 79% f1-score. Moreover, our classifier also perform fairly well in predicting emotion of anger (index -3), fear (index -2), sadness (index -1) and joy (index 1), with 72% - 80% f1-score. However, only 59% of the validation data with emotion of love (index 2) can be correctly recalled by our classifier. From the evaluation results, our classifier perform the best in classifying emotion of joy, while perform the worst in classifying emotion of love.

8. Prediction on test result

Test data set with 2000 inputs are passed to the trained machine learning model for prediction. Appendix A shows the predicted result on test data set.

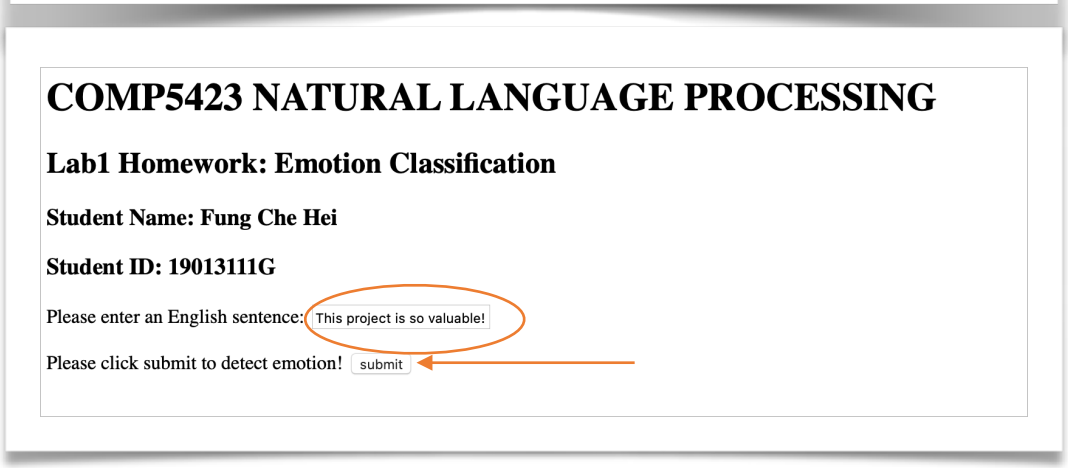
9. Web Application and user interface

A simple web application is developed for detecting emotion with a given English sentence. To use the web application, first please launch a python file application.py. After then, please enter 'python application.py' in terminal to run the web application and follow the following steps.



Step 1:

Visit localhost:5000/

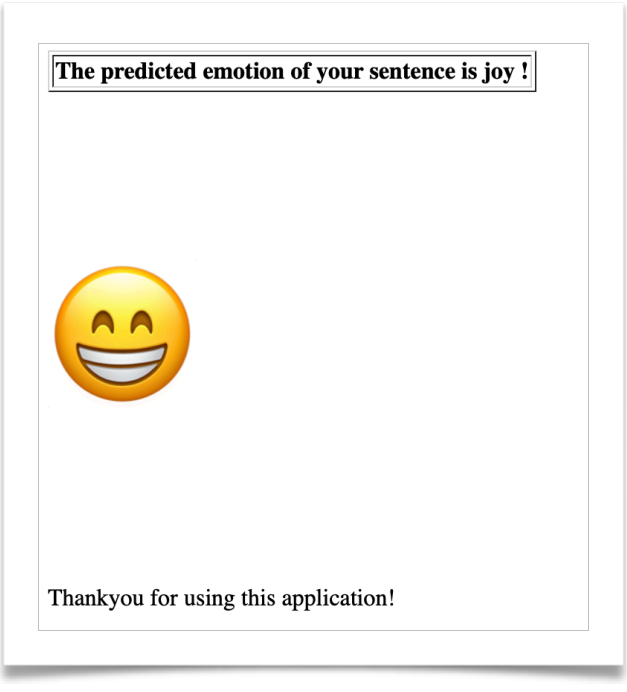


Step 2:

Input an English sentence in the box

Step 3:

Click 'submit' button



Step 4:

View the result

10. Limitation and future work

In the data set, due to the imbalance classes among the training data set, our classifier may not have sufficient example to learn well for certain emotion like ‘love’. More balanced data set can be obtained in future. Also, due to the lack of knowledge about how words or sentence styles are related to emotions, non-professional assumptions are made for generating different features. To build a better classifier, we can cooperate with phycological experts to understand more on the relationships between emotions and languages.

11. Conclusion

In this project, a classifier was built to identify emotion with a given English sentence. The classifier perform quite well with an overall accuracy of 76%. Generally the classifier can identify well with the emotions of joy, sadness, surprise, fear, and anger with 72% - 80% f1-score. However, due to imbalance classes among the training data set, the classifier cannot identify the emotion of love very well with just 66% f1-score. Besides the implementation of emotion classification system, a web application is also developed with our trained classifier. Such that emotion of a sentence inputted from the user can be identified in real time.

12. Appendix A — Predicted results on test data

test_prediction.txt

Please kindly find all my source code and data in GitHub via the following link:

https://github.com/Mickey1018/POLYU-COMP5423-NLP/tree/main/individual_project