LING 573

Natural Language Processing Systems And Applications

Prof. Gina-Anne Levow

TA: Haotian Zhu

PlaceboAffect - D2

1. Mohamed Elkamhawy (mohame@uw.edu)
2. Karl Haraldsson (kharalds@uw.edu)
3. Alex Maris (alexmar@uw.edu)
4. Nora Miao (norah98@uw.edu)

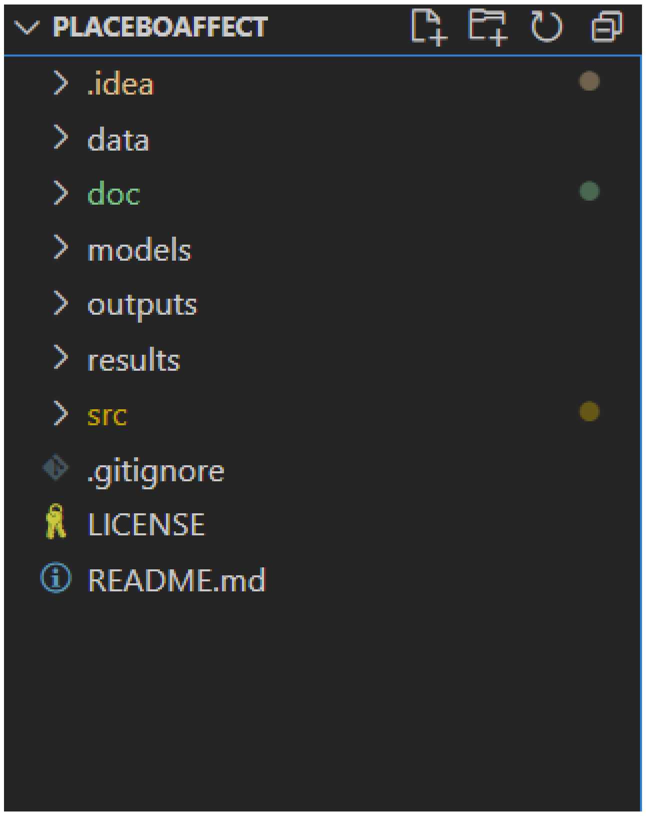
Introduction

This document will contain the details about D2 delivery for PlaceboAffect team. It will contain a walk-through for running the program and the expected output for each run.

Repo Link: https://github.com/MElkamhawy/PlaceboAffect

Folder Structure

This is the current folder structure for PlaceAffect Recognition system



data: This is the folder containing train/dev/test data for both English and Spanish for this task. For D2 we work only with English while Spanish will be used for adaptation later.
doc: This is the folder containing documentation files such as reports in addition to the current document as well.
models: This folder contains the saved versions of each model variations which will be explained later.
outputs: This contains the prediction files for each model.
results: This folder contains scores for each model.
src: This folder contains all source code for the system in addition to shell scripts that will be used for training/testing.

Setup

Given that each team is running our own task different from other teams. We created an environment that should be installed before running different shell scripts that will be consuming this environment.

Please, run the following script ***create_env.sh*** which contains the needed libraries that will make the project run smoothly.

```
./create_env.sh
```

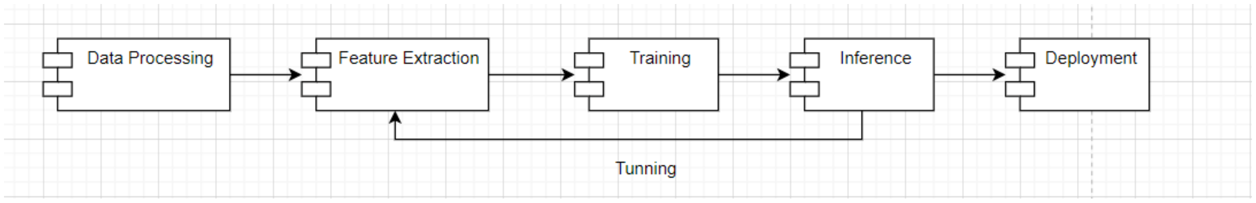
make sure to chmod for script to make it runnable beforehand if needed.

```
chmod +x create_env.sh
```

There is an extra (optional) step which is to activate the environment. (It'll be added to all shell scripts anyway)

```
conda activate PlaceboAffect
```

Components

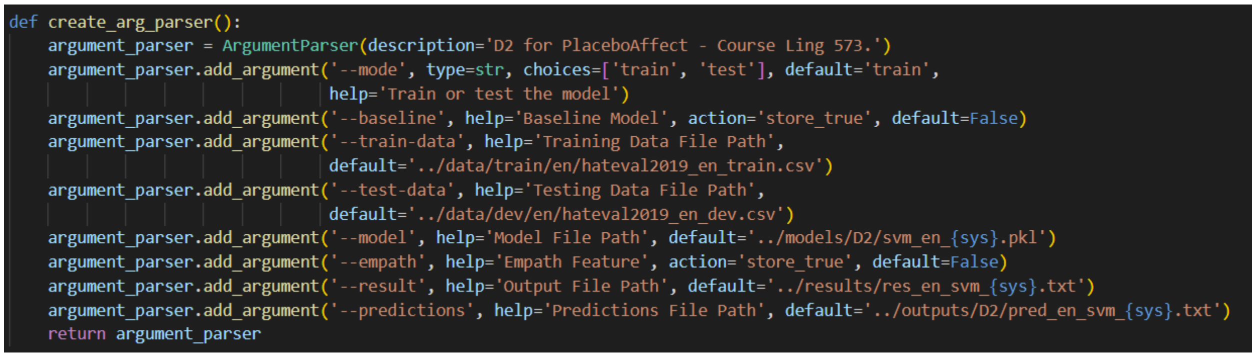


Data Processing: This is where we do ingestion/preprocessing/cleaning of the incoming data (tweets).
Feature Extraction: This is where we extract text features that includes: Bag of words, embeddings, empath…etc.
Training: This is for model choosing and doing training.
Inference: This is for doing the inference of the model output.
Deployment: This is for deploying the model to be running on patas/dryas clusters.

There is another one that's not part of the graph → **Evaluation**
We utilized the task script with slight modifications to make it work for our case.

Parameters

The parameters below are used to control the running of the system.



```
def create_arg_parser():
    argument_parser = ArgumentParser(description='D2 for PlaceboAffect - Course Ling 573.')
    argument_parser.add_argument('--mode', type=str, choices=['train', 'test'], default='train',
                                 help='Train or test the model')
    argument_parser.add_argument('--baseline', help='Baseline Model', action='store_true', default=False)
    argument_parser.add_argument('--train-data', help='Training Data File Path',
                                 default='../data/train/en/hateval2019_en_train.csv')
    argument_parser.add_argument('--test-data', help='Testing Data File Path',
                                 default='../data/dev/en/hateval2019_en_dev.csv')
    argument_parser.add_argument('--model', help='Model File Path', default='../models/D2/svm_en_{sys}.pkl')
    argument_parser.add_argument('--empath', help='Empath Feature', action='store_true', default=False)
    argument_parser.add_argument('--result', help='Output File Path', default='../results/res_en_svm_{sys}.txt')
    argument_parser.add_argument('--predictions', help='Predictions File Path', default='../outputs/D2/pred_en_svm_{sys}.txt')
    return argument_parser
```

```
--mode
```
Options are train or test.

```
--baseline
```
This is used to run the baseline system (BOW).

```
--train-data
```
Training data file path.

```
--test-data
```
Test data file path.

```
--model
```
Model file path to load/store from.

```
--empath
```
To activate empath feature for non-baseline model.

```
--result
```
Results file path.

```
--predictions
```
Predictions file path.

Scripts

We have 3 model variations:

- BOW (Bag of Words) – Baseline System
- Embedding
- Embedding with Empath

After looking at the results, BOW model was the winner although it was our baseline.
There are 3 train/test scripts for each model.

```
train.sh & test.sh
```
This is to run train/test for the baseline model.

```
train_embd.sh & test_embd.sh
```
This is to run train/test for Embedding model.

```
train_embd_empath.sh & test_embd_empath.sh
```
This is to run train/test for Embedding with empath model.

Condor

There is a condor file named **D2.cmd** which runs the inference for our best model (BOW).
Note: For some of our team members Condor was not running fine, so this might need to be run against the head node.

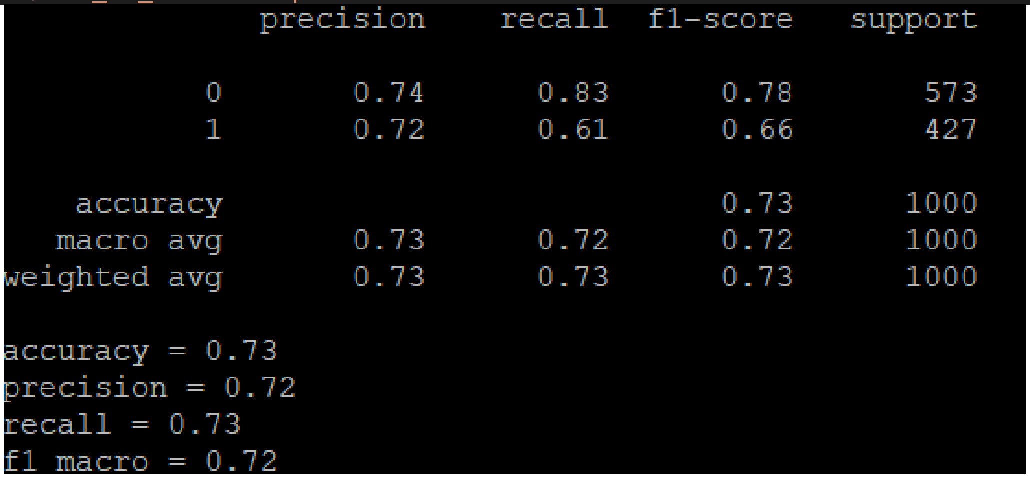
Output

There are 3 files as an output for each model run:

- Scores file: That contains model scores.
- Prediction file: That contains model predictions.
- Model file: That is the saved model file.

Bag of Words (BOW) – Main model

```
results/D2_scores.out
outputs/D2/pred_en_svm_baseline.txt
models/D2/svm_en_baseline.pkl
```



```
              precision    recall  f1-score   support

           0       0.74      0.83      0.78       573
           1       0.72      0.61      0.66       427

    accuracy                           0.73      1000
   macro avg       0.73      0.72      0.72      1000
weighted avg       0.73      0.73      0.73      1000

accuracy = 0.73
precision = 0.72
recall = 0.73
f1 macro = 0.72
```

Figure 1 - Numbers in D2_scores.out

Embeddings Model

```
results/D2_scores_embd.out
outputs/D2/pred_en_svm_embd.txt
models/D2/svm_en_embd.pkl
```

Embeddings with Empath Model

```
results/D2_scores_embd_empath.out
outputs/D2/pred_en_svm_embd_empath.txt
models/D2/svm_en_embd_empath.pkl
```