CSC 791- 603 Fall'21
Natural Language Processing
Assignment P2 Report

Krishna Prasanna (kbhamid2)

Assignment Task:

The dataset is a collection of sub-reddit posts, where there is a context post provided by OP (opinion practitioner) asking people 'AITA?'. Then there is a back-and-forth dialogue between challengers of the context and OP's own response.
The task is to classify the posts on whether OP's responses are 'excuses for his/her behavior', 'complementary information supporting his/her opinion' or neither.

Approach:

The dataset consists of 411 story descriptions and 1496 back-and-forth arguments between the challengers and practitioner. The classification labels are unequally distributed, however, not imbalanced.
The following steps were implemented in baseline model:

- **Data preprocessing:**
  Sentence cleaning: The sentences have an informal language with several abbreviated words like 'AITA', 'ESH', 'OP' etc. The embedding algorithm may not be able to handle these abbreviations.
  Dataset manipulation: All story descriptions have id1 value starting with **'t3_'.** This has been used to separate out the story descriptions from the arguments. Once separated, they were added as additional input (a new column) to the arguments that they support.
  The following tasks were performed in pre-processing on each sentence of the dataset:
  1. The common abbreviations were replaced with their expanded word phrases.
  2. Extra punctuations (. and :) have been removed.
  3. Dataset was modified by adding the story description as new column. This column will have the story description for the arguments made in its context setting. The final dataset will consist of only 1496 rows, but 3 text columns – story, text1, text2.
     Note that the OP's story description is used as an additional feature in both base-line as well as proposed models.
  4. NAs in the labels for rows corresponding to arguments were replaced with 0
- **Sentence Embeddings and their combinations:**
  For generating sentence embeddings there are two ways:
  (A) Perform tokenization with word-piece, generate word-level embeddings using BERT's pre-trained model and average them out to get sentence level embeddings.

(B) Generate sentence embeddings using SBERT. This paper discusses further in detail how SBERT generates better sentence embeddings using Siamese-network structure, in comparison to averaging the BERT-based word-embeddings of a sentence.

The following steps were implemented to generate sentence embeddings:

1. Sentence embeddings were generated using SBERT by leveraging the Hugging-Face's transformers library function SentenceTransformer.
2. Three SBERT embeddings were generated for text1, text2 and story sentences. Then a single embedding vector was generated by taking a weighted average of these embeddings.
3. The following weight combinations were explored for the three available features - text1, text2 and story:
    - Combination 1: Using only OP's defense argument with weights (0,1,0)
    - Combination 2: Using all three text inputs with weights (0.2,0.4,0.4)
    - Combination 3: Using OP's defense argument and story (0,0.5,0.5)
    - Combination 4: Using all three text inputs with weights (-0.2,0.6,0.6)

Note: "The most commonly used approach is to average the BERT output layer (known as BERT embeddings) or by using the output of the first token (the [CLS] token). This common practice yields rather bad sentence embeddings, often worse than averaging GloVe embeddings" (as taken from the paper)

- **Dataset preparation for model:**

  The final dataset comprises of a single weighted combination of 3 sentence embeddings. Labels have values 0,1 and 2. 75% (1122 rows) of the argument rows were used for training and 25% (374 rows) were used for testing.

Baseline Model:

- **Model selection:**

  Multi-class support vector classifier – SVM classifier is a supervised learning technique which constructs a separating hyper-plane that maximizes the margin between classes. The following hyper-parameters were set:
    - gamma = 0.7 – Indicates desired curvature of separating hyperplane
    - C = 0.5 – Allowed error margin traded off with misclassified points.
    - random_state = 124 – Friend's birthdate!
    - Kernel – The baseline model was implemented for two kernels – **linear kernel** and **polynomial kernel**.

  These models have been selected because they support multi-class classification which is desired for our tasks. All hyper-parameters were selected with trial-and-error, no optimization has been performed here.

- **Model Evaluation:**

  The model performances were evaluation on following metrics:

  1. **Accuracy score -** Since we have multi-class classification problem, accuracy is a good evaluation metric to gauge how our models are performing.
  2. **Confusion matrix -** Confusion matrix helps us understand our classification model better and access which classes are being most misclassified.
  3. **Classification report** – This report comprises of some extra evaluation metrics such as F1-score, precision, recall.

- **Model training and implementation:**
  1. There were 8 models run in total – 4 models each for Linear and Polynomial Kernels
  2. Each of these 4 models corresponds to a particular combination of weights for weighted-average sentence embedding.
  3. All models were evaluated on the above-described evaluation metrics.

- **Baseline model results:**

Weight combinations	Linear SVM accuracy	Polynomial SVM accuracy
(0,1,0)	**62%**	60%
(0.2,0.4,0.4)	59%	59%
(0,0.5,0.5)	61%	61%
(-0.2,0.6,0.6)	61%	**62%**

From above tables we can infer that challenger's text is not contributing to model's improvement. In fact, the model performance is degrading when we consider text1 as well. Also, Linear-SVM suffices for this classification task.

- **Classification reports:**

  Results of the top-2 models have been displayed in screenshots below:

```
RESULTS FOR COMBINATION 1
Classification report for linear SVM:
              precision    recall  f1-score   support

         0.0       0.66      0.61      0.63        96
         1.0       0.60      0.74      0.66       161
         2.0       0.61      0.45      0.52       117

    accuracy                           0.62       374
   macro avg       0.62      0.60      0.61       374
weighted avg       0.62      0.62      0.61       374

************************************************
Classification report for polynomial SVM:
              precision    recall  f1-score   support

         0.0       0.58      0.64      0.60        96
         1.0       0.61      0.71      0.66       161
         2.0       0.62      0.43      0.51       117

    accuracy                           0.60       374
   macro avg       0.60      0.59      0.59       374
weighted avg       0.60      0.60      0.60       374

************************************************
```

```
************************************************
RESULTS FOR COMBINATION 4
Classification report for linear SVM:
              precision    recall  f1-score   support

         0.0       0.64      0.58      0.61        96
         1.0       0.61      0.76      0.68       161
         2.0       0.59      0.44      0.50       117

    accuracy                           0.61       374
   macro avg       0.61      0.59      0.60       374
weighted avg       0.61      0.61      0.60       374

************************************************
Classification report for polynomial SVM:
              precision    recall  f1-score   support

         0.0       0.61      0.66      0.63        96
         1.0       0.62      0.71      0.67       161
         2.0       0.62      0.45      0.52       117

    accuracy                           0.62       374
   macro avg       0.62      0.61      0.61       374
weighted avg       0.62      0.62      0.61       374

************************************************
```

Proposed Model:

- **Additional features explored:**

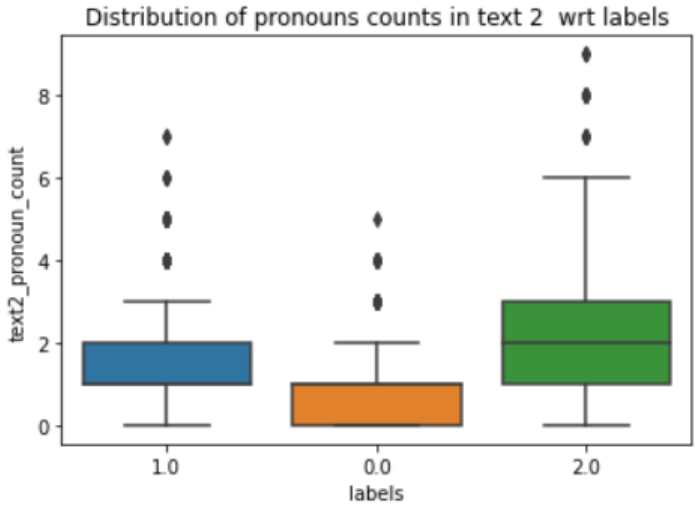
  The following features have been explored as additional features to improve SVM model performance:

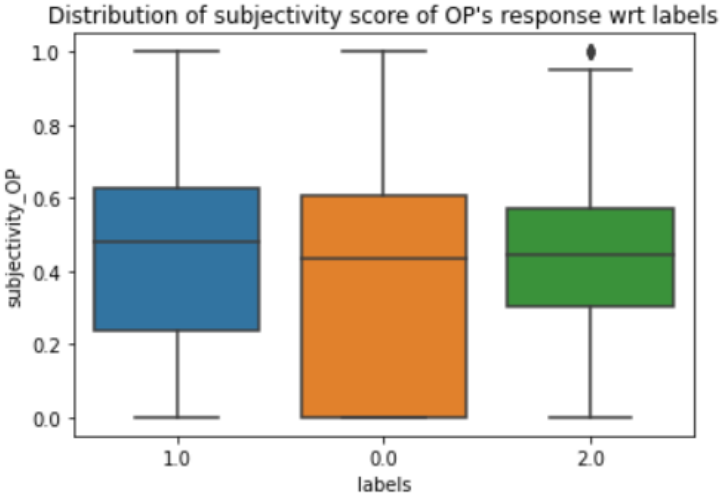
  1. Subjectivity score - Subjectivity score for OP's response texts was computed. Subjectivity score in Textblob library gives the score of how subjective a particular piece of text is, by accounting for emotions/personal opinions vs facts. It gives a score between 0 and 1, where 0 indicates very objective and 1 indicates very subjective.
  2. Count of pronouns: The count of distinct pronouns in OP's response and challenger's response were other features for consideration.
  3. Polarity score - Polarity (sentiment) score of challenger's response was considered. Polarity score in Textblob library gives the score of sentence polarity (positive or negative).

- **Additional feature selection (top 2):**

  Box-plots were used to evaluate how well these additional features explained the labels (0,1,2). Features having a varying range w.r.t these labels are capable of improving the model.

  Based on box-plots, the finalized additional features are – **Count of pronouns in OP's response, subjectivity score in OP's response.**





- **Model selection:**

  To maintain a fair comparison of performance by inclusion of additional features, SVM classifier has been used to perform the desired multi-class classification – the linear and polynomial kernels were explored with hyper-parameters mentioned in base-line model.

- **Model training and implementation:**

  1. There were 8 models run in total as described in base-line model.
  2. All models were evaluated on the above-described evaluation metrics.

- **Proposed model results:**

Weight combinations	Linear SVM accuracy	Polynomial SVM accuracy
(0,1,0)	61%	61%
(0.2,0.4,0.4)	**63%**	59%
(0,0.5,0.5)	63%	61%
(-0.2,0.6,0.6)	61%	63%

From above tables we can infer that by addition of new features, challenger's text seems to be useful for the model's improvement, contributing a slight increase in the accuracy (59% to 63%).

Hence proposed model's accuracy seems to be 63% against best-baseline model's 62% (which is only a very slight increase)

- **Classification reports:**

  Results of the one top model have been displayed in screenshot below:

```
**********************************************
RESULTS FOR COMBINATION 2
(1496, 770)
Classification report for linear SVM:
              precision    recall  f1-score   support

         0.0       0.66      0.62      0.64        96
         1.0       0.61      0.80      0.69       161
         2.0       0.68      0.41      0.51       117

    accuracy                           0.63       374
   macro avg       0.65      0.61      0.61       374
weighted avg       0.64      0.63      0.62       374

**********************************************
Classification report for polynomial SVM:
              precision    recall  f1-score   support

         0.0       0.57      0.65      0.61        96
         1.0       0.59      0.70      0.64       161
         2.0       0.62      0.41      0.49       117

    accuracy                           0.59       374
   macro avg       0.60      0.58      0.58       374
weighted avg       0.60      0.59      0.59       374

**********************************************
```

Conclusion:

Hence the classification task has been successfully performed. Some take-away points here are:

1. Other classifiers need to be explored (which support multiple epochs of training to improve weights of features). Some suggestions are CNN based soft-max classifiers, which are seen in text-classification models of transformers.
2. Although challenger opinion influences OP's response (whether challenger says YTA or ESH), challenger's statement itself doesn't necessarily add to model's improvement. We see that we can make do without challenger's opinion without affecting accuracy too much.
3. The additional features only slightly improve accuracy of the model, and linear SVM suffices for this task.