SciKit Black Belt

Oliver Alonzo, Josh Bicking, Jennifer Liu

Preprocessing

Overall approach: We broke the task into 3 parts. We tried different approaches for solving the same problem, and compared the results from each of the method. For each task, we chose the method that produced the highest accuracy on the splitted training data.

Evaluation Methods: To determine how well our classifiers were doing, we took the training data and split it into training and test set. 20% for testing, and 80% for training. We ran the classifiers on the testing data, and observed the precision, recall and f1-measure, as well as their macro, micro and weighted averages. We also observed the confusion matrix.

Preprocessing Data: We noticed that there were some entries where the sentence was “None”. Thus, we created functions to disregard those data points. We removed stop words and punctuations from the sentences, and lemmatized all the words. We also noticed that there were words like “soooo”, which is the same as “so”. So, we wrote functions that would convert these types of words into its original form.

Method: Polarity Prediction

Our approach to solving the polarity prediction was by using Support Vector Machine.

After the preprocessing phase, we converted the comment into a TF-IDF vector. These were the features that were fed into the SVM classifier. We used Sci-Kit Learn SVM library to find the best decision boundary that separates the positive,neutral, and negative comments.

I attempted to tune the kernel parameter. I tried all 4 possible kernel parameters, and linear kernel seemed to perform the best.

Method: Genre Prediction

To predict the Genres, we used an approach similar to what we learned from our textbook, using cosine similarity.

To do this, the text was represented as tf-idf vectors of the pre-processed text. Then, a K-Nearest Neighbor classifier was trained on those vectors, using cosine distance as its metric.

In order to determine the number of K neighbors for the classifier, we trained models using 1 through 40 as possible values, and selected the one that returns the lowest mean squared error using 10-fold cross validation for each model.

Method: Topic Prediction

Complement Naive Bayes classifier used for topic prediction.

Sample sentence rendered as a count vector (a matrix of token counts that SciKit models can understand).

While CNB is suited for imbalanced data sets, it performed well on balanced topic data, as CNB is essentially a “usually better performing” Multinomial NB.

Results: Genre Prediction

Baseline

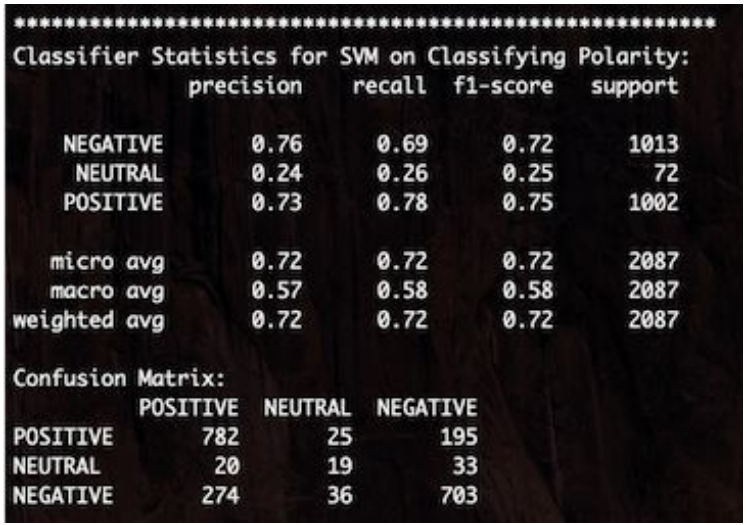
```
**************************************************
Classifier Statistics for Baseline*:
              precision    recall  f1-score   support

     GENRE_A       0.00      0.00      0.00       371
     GENRE_B       0.82      1.00      0.90      1716

   micro avg       0.82      0.82      0.82      2087
   macro avg       0.41      0.50      0.45      2087
weighted avg       0.68      0.82      0.74      2087

Confusion Matrix:
           GENRE_A  GENRE_B
GENRE_A        0      371
GENRE_B        0     1716
```

KNN

```
**************************************************
Classifier Statistics for KNN:
              precision    recall  f1-score   support

     GENRE_A       0.45      0.87      0.59       371
     GENRE_B       0.96      0.77      0.86      1716

   micro avg       0.79      0.79      0.79      2087
   macro avg       0.71      0.82      0.72      2087
weighted avg       0.87      0.79      0.81      2087

Confusion Matrix:
           GENRE_A  GENRE_B
GENRE_A      322       49
GENRE_B      397     1319
```

Method for determining Baseline: Assigning the most frequent class to all (GENRE_B)

Results: Polarity Prediction



```
**********************************************************
Classifier Statistics for SVM on Classifying Polarity:
              precision    recall  f1-score   support

    NEGATIVE       0.76      0.69      0.72      1013
     NEUTRAL       0.24      0.26      0.25        72
    POSITIVE       0.73      0.78      0.75      1002

   micro avg       0.72      0.72      0.72      2087
   macro avg       0.57      0.58      0.58      2087
weighted avg       0.72      0.72      0.72      2087

Confusion Matrix:
          POSITIVE  NEUTRAL  NEGATIVE
POSITIVE       782       25       195
NEUTRAL         20       19        33
NEGATIVE       274       36       703
```

Baseline: 48% (1013/2087)

Method for determining Baseline: Negative comments were the majority class in the training set. Hence, if we assigned all the testing data with negative, we would have a 48% accuracy.

Result: Topic Prediction

****
Classifier Statistics for CNB on Classifying Topic:

	precision	recall	f1-score	support
(FEAR_OF)_PHYSICAL_PAIN	0.74	0.58	0.65	45
ATTENDING_EVENT	0.69	0.53	0.60	45
COMMUNICATION_ISSUE	0.63	0.39	0.48	44
GOING_TO_PLACES	0.85	0.85	0.85	48
LEGAL_ISSUE	0.72	0.93	0.82	45
MONEY_ISSUE	0.83	0.84	0.83	51
OUTDOOR_ACTIVITY	0.67	0.83	0.74	46
PERSONAL_CARE	0.59	0.74	0.66	47
micro avg	0.72	0.72	0.72	371
macro avg	0.72	0.71	0.70	371
weighted avg	0.72	0.72	0.71	371

Baseline:

~13.7% (51/371)

Method for determining Baseline:

Assigning the most common class (MONEY_ISSUE) to all data.

Confusion Matrix:

	NONE	GOING_TO_PLACES	MONEY_ISSUE	PERSONAL_CARE	ATTENDING_EVENT	(FEAR_OF)_PHYSICAL_PAIN	COMMUNICATION_ISSUE	OUTDOOR_ACTIVITY	LEGAL_ISSUE
NONE	0	0	0	0	0	0	0	0	0
GOING_TO_PLACES	0	41	0	0	3	1	0	1	2
MONEY_ISSUE	0	2	43	2	0	0	1	0	3
PERSONAL_CARE	0	2	5	35	0	0	3	2	0
ATTENDING_EVENT	0	0	2	1	24	3	1	10	4
(FEAR_OF)_PHYSICAL_PAIN	0	0	2	10	1	26	2	4	0
COMMUNICATION_ISSUE	0	2	0	8	6	4	17	1	6
OUTDOOR_ACTIVITY	0	1	0	2	0	1	3	38	1
LEGAL_ISSUE	0	0	0	1	1	0	0	1	42

Observation and Discussion: Polarity Prediction

- **Results remained consistent**
  - When the data was tested by spitting the training data, we received a weighted average of 72% for precision on the test set. Similarly, when testing it on the test data provided by professor, we also received a weighted average of 72% for precision.
- **Neutral received the lowest accuracy**
  - Due to the minimal data we have for comments that are considered neutral, the classifier did significantly poorer on detecting comments that are neutral
- **Results was higher than baseline**
  - The overall weighted averaged for precision, recall, and f1 results showed that the classifier did significantly better than the baseline . Therefore, it is reasonable to say that our classifier is doing alright, even though the accuracy rate seems rather low.

Observation and Discussion: Genre Prediction

- **Importance of Domain Knowledge**
  - When first applying KNN, we were getting f1-scores no higher than around 0.50. However, this improved by 75% (to f1-scores averaging 0.85) by adding the cosine similarity as the metric for KNN.
- **Cross Validation**
  - The results above where further enhanced by the implementation of cross validation to determine the optimal value of K.
- **Results was higher than baseline**
  - The results showed that the classifier did better than the baseline for precision and f1-scores. We speculate that the reason recall was higher in the baseline is simply because of the unbalanced test dataset, which we will discuss in the reflection about the data.

Observation and Discussion: Topic Prediction

- **Results varied by topic**
  - *COMMUNICATION_ISSUE* produced an F1 score of 0.48, while a handful of others produced more than 0.80
- **Difficult to compare results to baseline**
  - Since topic data was mostly balanced, neither Random Prediction or Zero Rule baselines were a competitive standard
  - Baseline accuracy of 13.7% was surpassed for each topic

Reflection on Data: Polarity Prediction

- **Unbalanced Dataset causing low accuracy for polarity**
  - An approach solving the Unbalanced Dataset was to give different weights for SVM class separation. For the class that appear the least, which is neutral, the weight would be higher versus the class that appears the most, which is negative comments. However, adjusting the weight mechanism didn't seem to boost performance much when tested on the testing data during the split.
- **Lack of rich contextual understanding from classifier**
  - There were some comments that require one to have a rich knowledge of the world in order to make polarity judgements. However our classifier may not be able to capture such context as well as human can. For instance, this comment, "I haven't been to a dentist for a year or two.", extracted from the training dataset was marked as positive. As humans, we mark this as positive because we know that "going to a dentist" generally means that you have bad teeth and it requires fixing. Thus, we can conclude that not having to go is a positive thing. While the TF-IDF can consider contextual relationship in terms of the co-occurring phrases, it cannot consider all the possible context that a human acquire throughout its lifetime. From the above example, TF-IDF may not be able to learn that "going to a dentist" carries a negative connotation.

Reflection on Data: Genre Prediction

- **Unbalanced test dataset's impact on metrics**
  - The unbalanced nature of the test dataset had an impact on the metrics used. For instance, precision was lower than recall because the number of false negatives for GENRE_B was over even the total true positives for GENRE_A, even though in percentages they were almost the same. So, even if the system achieved a 100% recall, precision would have been lower than 50%.
- **Benefits of balanced training dataset**
  - In spite of the issues mentioned above, the genre prediction may have benefitted the most from a balanced training dataset, given that both classes had nearly the same number of instances (with the exception of a few sentences that were not considered because of typos or formatting errors). In addition, it had the largest amount of instances for each class when compared to the other two problems. This may be one of the reasons why genre prediction did better than the other two tasks.

Challenges

Initially, when comparing how well each classifier did for each of the 3 tasks, we were extracting 80% of the sample as training data and the remaining as testing data. Each classifier was shuffling the data randomly before splitting it. Due to the random shuffling process, there was no common basis for comparing the accuracy result. It is possible that Naive Bayes classifier was extracting a test sample that was by chance classifying the data more accurately. Thus, to combat this issue, we made sure to split the data the exact same way when running the prediction method for each of the classifier.

Task Distribution

Jennifer Liu 30% - Built the SVM Classifier

Oliver Alonzo 30% - Built the KNN Classifier

Josh Bicking - 30% - Built the Complement Naive Bayes Classifier

Conclusion

How can we improve the classifier?

- I think we can do more when cleaning the data. In future work, we can consider using Part of speech tags as an additional feature input.
- We can consider combining multiple classification techniques. In other words, we can use the ensemble learning technique.

Sources

1. Jurafsky, Dan, and James H. Martin. Speech and Language Processing. 3rd ed., Draft.
2. Kumar, Abhijeet. “Conventional Approach to Text Classification & Clustering Using K-Nearest Neighbor & K-Means: Python Implementation.” Machine Learning in Action, 3 Sept. 2018, appliedmachinelearning.blog/2018/01/18/conventional-approach-to-text-classification-clustering-using-k-nearest-neighbor-k-means-python-implementation/.
3. Robinson, Scott. “K-Nearest Neighbors Algorithm in Python and Scikit-Learn.” Stack Abuse, Stack Abuse, 15 Feb. 2018, stackabuse.com/k-nearest-neighbors-algorithm-in-python-and-scikit-learn/.
4. Zakka, Kevin. “A Complete Guide to K-Nearest-Neighbors with Applications in Python and R.” Kevin's Blog, 13 July 2016, kevinzakka.github.io/2016/07/13/k-nearest-neighbor/#parameter-tuning-with-cross-validation.