Aspect Based Sentiment Analysis

A Project Report

Presented to Dr. Bing Liu

Department of Computer Science

University of Illinois at Chicago

In Partial Fulfillment of the Requirements for the Class CS 583

By KruneetKumar Patel and Vishwas Sreevalli Ramamohan

Fall 2018

Abstract

To determine the opinion of any person experiencing any services or buying any product, the usage of Sentiment Analysis, a continuous research in the field of text mining, is a common practice. It is a process of using computation to identify and classify opinions expressed in a piece of text. Individuals post their opinion via reviews, tweets, comments or discussions which is our unstructured information. Sentiment analysis gives a general conclusion of audits which benefit clients, individuals or organizations for decision making.

Introduction

Internet is amongst the most rapidly developing technologies and has become an essential part in today's world. Data on internet varies from areas like academics, criticism or conclusion about items, remarks on social issues and so forth. Individuals regularly communicate, examine and share data via web. It helps individuals to compare and settle on choice in numerous things. Large number of individuals dependably tune into other's assessment before making any choice of the service.The fundamental goal is to anticipate the sentiment of a review by performing and analyzing multiple classifying techniques and learning models.

Data pre-processing

Special character removal and stop word expulsion strategy has been performed. Tokenization separates given text into tokens. Using NLTK, the following different strategies forms pieces of the preprocessing step, as a part of sentiment analysis: • Conversion from uppercase to lower case letter, expel undesirable punctuations and additional white spaces, and evacuate newline and special characters. • Stemming, a process to reduce inflected or morphological word forms to their base form or root form i.e. word stem is used to reduce the total number of unique words in the dataset. Porter Stemming algorithm is an algorithm to expel suffixes from English words. Bigram and Unigram features are utilized for getting the features.

TF-IDF Vectorizer and CountVectorizer are considered as well-known techniques for feature selection.

Out[36]:

	example_id	text	aspect_term	term_location	class	clean_text	clean_aspect_term	element_weights
0	3121_0	But the staff was so horrible to us.	staff	8--13	-1	[But, staff, horrible, u]	[staff]	{'But': 1.0, 'staff': 2, 'horrible': 1.0, 'u':...
1	2777_0	To be completely fair[comma] the only redeemin...	food	57--61	1	[To, completely, fair, redeeming, factor, food...	[food]	{'To': 0.2, 'completely': 0.25, 'fair': 0.3333...
2	1634_0	The food is uniformly exceptional[comma] with ...	food	4--8	1	[The, food, uniformly, exceptional, capable, k...	[food]	{'The': 1.0, 'food': 2, 'uniformly': 1.0, 'exc...
3	1634_1	The food is uniformly exceptional[comma] with ...	kitchen	55--62	1	[The, food, uniformly, exceptional, capable, k...	[kitchen]	{'The': 0.2, 'food': 0.25, 'uniformly': 0.3333...
4	1634_2	The food is uniformly exceptional[comma] with ...	menu	141--145	0	[The, food, uniformly, exceptional, capable, k...	[menu]	{'The': 0.0769230769230769, 'food': 0.0833333...

Classification approaches and Models used

Machine learning and lexicon-based approaches are two the types of classification approaches. The corpus-based approach classifies words by considering the bunch of words as word list. Naïve Bayes and Support Vector Machine (SVM), Random forest algorithms are utilized for classification, and it has been shown that Support Vector Machine performs better to Naïve Bayes.

Experiment Results

SVM:

In [30]:
```
print(precision_recall_fscore_support(df1['class'], pred_SVM, labels=[-1,0,1]))

print("\n Classification Report \n ", classification_report(pred_SVM,df1['class']))
```

```
(array([0.68120805, 0.55555556, 0.78251599]), array([0.73639661, 0.47018349, 0.78251599]), array([0.70772807, 0.50931677, 0.782
51599]), array([827, 436, 938], dtype=int64))

 Classification Report
              precision    recall  f1-score   support

          -1       0.74      0.68      0.71       894
           0       0.47      0.56      0.51       369
           1       0.78      0.78      0.78       938

avg / total       0.71      0.70      0.71      2201
```

Decision Tree:

```
In [38]: DT = DecisionTreeClassifier(max_depth=10).fit(X,Y)
         pred = cross_val_predict(DT,X,Y,cv=10)
         print(precision_recall_fscore_support(Y, pred, labels=[-1,0,1]))

         print("\n Classification Report \n ", classification_report(pred,Y))

         (array([0.45732052, 0.65957447, 0.90649351]), array([0.97823458, 0.07110092, 0.37206823]), array([0.62326656, 0.12836439, 0.527
         58881]), array([827, 436, 938], dtype=int64))

          Classification Report
                       precision    recall  f1-score   support

                 -1       0.98      0.46      0.62      1769
                  0       0.07      0.66      0.13        47
                  1       0.37      0.91      0.53       385

         avg / total       0.85      0.54      0.60      2201
```

Naive Bayes Multinomial Classifier:

```
n [43]: print(precision_recall_fscore_support(Y, pred_NB, labels=[-1,0,1]))

        print("\n Classification Report \n ", classification_report(pred_NB,Y))

        (array([0.68047983, 0.68571429, 0.70541958]), array([0.75453446, 0.22018349, 0.86034115]), array([0.71559633, 0.33333333, 0.775
        21614]), array([827, 436, 938], dtype=int64))

         Classification Report
                      precision    recall  f1-score   support

                -1       0.75      0.68      0.72       917
                 0       0.22      0.69      0.33       140
                 1       0.86      0.71      0.78      1144

        avg / total       0.78      0.69      0.72      2201
```

Random Forest:

```
In [46]: print(precision_recall_fscore_support(df1['class'], pred_RF, labels=[-1,0,1]))

         print("\n Classification Report \n ", classification_report(pred_RF,df1['class']))

         (array([0.66633858, 0.6490566 , 0.79130435]), array([0.81862152, 0.39449541, 0.7761194 ]), array([0.73467173, 0.49072753, 0.783
         63832]), array([827, 436, 938], dtype=int64))

          Classification Report
                       precision    recall  f1-score   support

                 -1       0.82      0.67      0.73      1016
                  0       0.39      0.65      0.49       265
                  1       0.78      0.79      0.78       920

         avg / total       0.75      0.72      0.73      2201

In [ ]:
```

Conclusion and Future work:

We were able to successfully analyze the sentiments for the given data using various aspects. Looking at the accuracy we've got, further preprocessing could increase the performance. We plan to build models based on neural networks to learn how the results would differ.