Subtheme Sentiment Analysis Task

Task:

The task is to develop an approach that given a sample will identify the subthemes along with their respective sentiments.

Approach:

We develop a rule-based approach combined with natural language processing (NLP) techniques to identify subtheme sentiments in customer reviews. Each review is tokenized and analyzed for key aspects (subthemes) and their associated sentiments using TextBlob. Sentiments are classified as positive, negative, or neutral based on polarity scores. The subthemes are identified by matching predefined keywords with review segments.

Problem to be solved: Given a dataset containing some text related to a movie, the problem is to predict the sentiment behind the statement in the form of -1 and 1 label (-1 for negative and 1 for positive).

Let's first begin with importing all the necessary libraries.Quickly go through the purpose of the libraries.

Pandas- for storing and analysing the data.

CountVectorizer-to get the frequency of words.

re-for locating and matching patterns in text.

PorterStemmer-for stemming words.

nltk-for accessing other packages like PorterStemmer and stopwords.

stopwords-for removal os stopwords.

WordCloud-for generating word cloud showing words according to their frequency in the text.

matplotlib-for plotting some necessary graphs.

TfidVectorizer-to generate word vectors based on the document weightage.

Code:

The code includes loading data, renaming columns, performing sentiment analysis, replacing NaN values, and calculating sentiment scores and percentages. It leverages pandas for data manipulation and TextBlob for sentiment analysis.

Explanation:

1. **Load the Data**: Load the CSV file into a DataFrame.
2. **Rename Columns**: Rename the columns appropriately.
3. **Replace NaN Values**: Replace NaN values in the specified review columns with 0.
4. **Sentiment Analysis**: For each review column, apply a function that determines the sentiment of the review and replaces it with 1 for positive, -1 for negative, and 0 for NaN or 0 values.
5. **Aggregate Sentiment Scores**: Calculate the total sentiment score for each row.
6. **Calculate Percentages**: Compute the percentage of positive and negative reviews in each row based on the total number of review columns.
7. **Plot the Calculate Percentage**: Now we plot a pie chart of the positive and negative data to understand the ratio of the data. Matplotlib library is used to plot the pie chart using various parameters.
8. **Customer Preprocessing Function:** Tokenize the data present in the customer review column with the help of wordcloud to determine the positive and negative words and show it on the positive and negative chart.
9. **Train and Test Data:** Train and Test the data help of logistic regression.A model is created by passing various parameters. Number of folds in cross validation is given as 6, scoring is estimated based on accuracy and maximum iterations are set to 500 (you may try out with some other values). The model is then fitted on the training data and further predicted on the test data(X_train) to get y_pred which is the predicted labels.
10. **Accuracy Check:** Its time to check how well it has learnt everything. For that we use accuracy_score from metrics package of the sklearn library. The predicted labels and actual test labels are passed as parameters to get the accuracy score.

Conclusion :

The approach successfully extracts subtheme sentiments, categorizing them accurately based on context. For example, in the sample review, it identifies "incorrect tyres sent" as negative, "garage service" as positive, and "wait time" as negative. This method effectively highlights the different aspects of the service and their associated sentiments, providing valuable insights.

Result: That's cool, 93% is the accuracy of our model . We got on Subtheme Sentiment Analysis Task.