NLP Assignment

Overview: Used SemEval dataset to predict the sentiment/polarity given a text and corresponding entity in the Aspect column

Steps

1. Preprocessing
2. Embedding Methods
    a. Frequency based method: Bag of Words
    b. Embedding based method: GloVe
    c. Sentence Vectors: Universal Sentence Encoder
3. Trained using 3 classifiers (Random Forest, Decision Tree, SVM)for every embedding method and performed comparative analysis

1. Preprocessing

- ⇨ For preprocessing, first I checked weather the dataset consists of any null values.
- ⇨ Checked the unique value counts in the output feature, the polarity is categorized into 4 different polarity types: positive, negative, neutral and conflict. As conflict represents only small portion of the dataset and also it is not very important to classify into a clear cut sentiment trend.

polarity	count
positive	987
negative	866
neutral	460
conflict	45

- ⇨ After analysing the dataset we come to know that the two important features for determining the polarity of the given text are sentence and aspect hence I performed further preprocessing only these two features. In the next step I imported necessary libraries from Nltk and performed basic text preprocessing task such as:
- ⇨ **Lowercasing:** Lowercasing all the text to ensure uniformity which also eliminates the issue where same word might be considered different being in different cases.
- ⇨ **Removing stop words:** Removed stop words from the text as they do have any significant meaning in determining the polarity of the text.
- ⇨ **Removing punctuations:** Removed punctuations and special characters.
- ⇨ **Removing extra whitespaces:** Extra whitespaces can cause distortion in tokenization.

4. **Frequency Based Method:**

   a. **Bag of Words**
      - BoW represents the text data as a collection of words based on the frequency of the words. Each word in the corpus is considered as a unique feature. The first step in bag of words is to create its vocabulary from the entire dataset, each sentence is then converted into a vector where each position within the vector corresponds to the word in the vocabulary.
      - In this technique Bag of Words doesn't consider the context of each word with respect to the sentence, rather it just focusses on the frequency of occurrence of every word in the sentence.
      - Post creating BoW for aspect and sentence feature, I concatenated both because of following reasons
        - To provide both contextual and aspect specific information
        - Better for model to understand how sentence relates to specific aspect being analysed
      - **Feature Selection:** Post Concatenating, I use chi square to pick the top 100 features, I did so for the following reasons:
        - To focus on more important words
        - To reduce the dimensionality by reducing the number of input features.
        - To Improve model Accuracy
      - **Training the model:** Classifiers used: Random Forest, SVM, Decision tree
      - **Results:**

```
Random Forest 10-fold CV Accuracy: [0.5862069  0.61206897 0.65517241 0.63203463 0.58008658 0.62770563
 0.58441558 0.61471861 0.65800866 0.58441558]
Mean Random Forest Accuracy: 0.6134833557247349
SVM 10-fold CV Accuracy: [0.5862069  0.62931034 0.68965517 0.61038961 0.5974026  0.62337662
 0.58874459 0.61471861 0.62337662 0.61904762]
Mean SVM Accuracy: 0.618222869084938
Decision Tree 10-fold CV Accuracy: [0.5862069  0.56896552 0.61637931 0.60606061 0.54978355 0.62337662
 0.57575758 0.5974026  0.59307359 0.57142857]
Mean Decision Tree Accuracy: 0.5888434841021047
```

      - **Analysis:**
        - From analysis, SVM outperforms the other two algorithms with an accuracy of 61.82%. Basically SVM is suitable for bag of words because of its ability to handle many features. SVM finds the best separation between the classes which is suitable for classification. As in BoW we generate a sparse matrix is manages to focus on key features
        - Random forest accuracy of 61.34% is close to SVM. Random forest combines multiple decision tress and prevents the overfitting

drawback of decision trees. It performs well because random forest is good at handling complex relationships or interactions between the words which is important during classification.

- Decision tree has the least accuracy of 58.88%. As discussed in the above point decision tress tend to overfit on BoW because it splits the data based on individual words and may learn patterns that don't generalize well.
- Basically bag of words fail to understand the context of the words hence the performance of the models is low as compared to the other type of embeddings.

b. Embedding Based Method: GloVe

- GloVe is basically a popular word embedding model which is used to transform words into continuous vector spaces. In this assignment, I have used a pretrained word embedding model that provides a vector representation of words. I am using wiki-gigaword-300 which is trained on Wikipedia and the Gigaword corpus. Unlike bag of words, this technique focusses on understanding the semantic relationship between words hence the model can understand the context better.
- For this I first performed word tokenization(splitting the text into smaller units), post which I passed these token of sentence and aspects to be mapped to their corresponding vectors in the GloVe model.
- Post this, I concatenated the aspect and sentence embeddings and passed it as an input to train the classifiers
- **Training the model:** Classifiers used: Random Forest, SVM, Decision tree

```
Random Forest 10-fold CV Accuracy: [0.61206897 0.59913793 0.68534483 0.64069264 0.55844156 0.62770563
 0.65800866 0.64935065 0.67099567 0.65367965]
Mean Random Forest Accuracy: 0.6355426183012389
SVM 10-fold CV Accuracy: [0.63793103 0.61637931 0.61637931 0.62337662 0.5974026  0.61904762
 0.61471861 0.63203463 0.62770563 0.60606061]
Mean SVM Accuracy: 0.6191035975518735
Decision Tree 10-fold CV Accuracy: [0.56896552 0.42672414 0.5        0.47619048 0.44588745 0.51515152
 0.5021645  0.47619048 0.51515152 0.48051948]
Mean Decision Tree Accuracy: 0.49069450664278247
```

- **Analysis:**
  - Random forest gives the highest accuracy of 63.55%, as discussed above random forest combines multiple decision tress and prevents the overfitting drawback of decision trees. As GloVe embeddings capture the context of the sentences they have an accuracy higher than bag of words which fail to capture the contextual understanding.
  - SVM gives an accuracy of 61.91% which is slightly lower than Random Forest. SVM is capable of handling high dimensional data well and GloVe's ability to represent into a dense vector enhances its ability to find boundaries between classes.
  - Decision Trees have the lowest accuracy of 49.07%. Decision trees are not able to handle complex embeddings. Hence have the poorest performance.
  - GloVe embeddings are capable of understanding the semantic relationship between the words. On the other hand bag of words is incapable of understanding the semantic relations hence has a poorer performance as compared to GloVe.

c. **Sentence Vectors: Universal Sentence Encoder:**
   - For Sentence Vectors I have used Universal Sentence Encoder, unlike GloVe, USE generates embeddings for entire sentence. USE is used to capture the semantic meaning of the entire sentence using advanced deep learning techniques. The embeddings are basically derived from a model which is trained on sentences which allows it to understand the meaning based on context.
   - Post forming embeddings of aspect and sentences I combined and provided it as an input to train the classifiers.
   - **Training the model:** Classifiers used: Random Forest, SVM, Decision tree

```
Random Forest 10-fold CV Accuracy: [0.65086207 0.60344828 0.72413793 0.68398268 0.65800866 0.64502165
 0.66666667 0.69264069 0.64935065 0.65800866]
Mean Random Forest Accuracy: 0.6632127929541722
SVM 10-fold CV Accuracy: [0.63793103 0.65086207 0.7112069  0.64502165 0.6969697  0.64935065
 0.68398268 0.61904762 0.70995671 0.65367965]
Mean SVM Accuracy: 0.6658008658008658
Decision Tree 10-fold CV Accuracy: [0.5        0.51293103 0.55603448 0.50649351 0.54545455 0.5974026
 0.52380952 0.54112554 0.57142857 0.54978355]
Mean Decision Tree Accuracy: 0.5404463352739215
```

   - **Analysis:**
     - SVM gives an accuracy of 66.58% which is highest among all the classifiers. USE basically focusses on understanding the context of the word with respect to other words in the sentence hence it captures meaning more effectively as compared to other two techniques (bag of words and GloVe). SVM is able to capture complex patterns effectively.
     - Random Forest gives an accuracy of 66.32 which is close to SVM.
     - Decision Tree gives the least accuracy of 54.04%.
     - USE is pretrained on wide variety of texts and tasks which enhances it ability to generalize.

Conclusion:

- ⇨ In conclusion, the results show clear performance differences based on the embeddings and classifiers used. Bag of Words, which lacks contextual understanding, resulted in the lowest accuracies, with SVM achieving 61.82%, the best among the classifiers for this method. This reflects how SVM can handle high-dimensional sparse data, but BoW's limitations in capturing semantics restricted its performance.
- ⇨ GloVe embeddings, which account for word meaning and context, outperformed BoW, with Random Forest achieving the highest accuracy at 63.55%. This shows that GloVe's dense vector representations better support models like Random Forest, which excels at handling complex interactions between features.
- ⇨ Universal Sentence Encoder (USE) yielded the highest accuracies overall, with SVM achieving 66.58%. This demonstrates USE's superiority in capturing sentence-level context, which SVM leveraged to effectively classify sentiment. Therefore, USE combined with SVM is the best-performing approach, providing the most accurate sentiment predictions in this analysis.