Hotel Review and Sentiment Analysis

CNN

Word2vec:

Pseudocode:

1. Connected the dataset to Colab using Google Drive.
2. Loaded the dataset, named reviews_df, from text_reviews.csv.
3. Performed text cleaning with the function def clean(text){}.
4. Imported TensorFlow and Keras libraries.
5. Created a dictionary with keys for text (reviews) and sentiments (labels).
6. Initialized a tokenizer.
7. Split the data into training and test sets with a 70:30 ratio.
8. Created a tokenizer using vocabsize for Word2Vec embedding.
9. Developed an embedding matrix using the tokenizer.
10. Trained a CNN model with a single layer.
11. Assessed the accuracy of the trained model on the testing dataset, stored as accuracy through model.evaluate().

Fasttext:

Pseudocode:

1. Established the connection between the dataset and Colab using Google Drive.
2. Loaded the dataset, reviews_df, from the file text_reviews.csv.
3. Applied text cleaning using the function def clean(text){}.
4. Imported the TensorFlow and Keras libraries.
5. Created a dictionary with keys 'text' for reviews and 'sentiments' for labels.
6. Initialized a tokenizer.
7. Divided the data into training and test sets with a 70:30 ratio.
8. Imported the wiki-news dataset for fasttext embedding.
9. Generated an embedding matrix using fasttext.
10. Trained a CNN model with a single layer.
11. Evaluated the accuracy of the trained model on the testing dataset, stored as accuracy using model.evaluate().

MCNN

Word2vec:

Pseudocode:

```
1. Connected the dataset to Colab via Google Drive.
2. Loaded the dataset, designated as reviews_df, from text_reviews.csv.
3. Implemented text cleaning through the function clean(text){}.
4. Imported TensorFlow and Keras libraries.
5. Established a dictionary with keys for text (reviews) and sentiments
   (labels).
6. Initialized a tokenizer.
7. Divided the data into training and test sets with a 70:30 ratio.
8. Created vocabulary size using the tokenizer.
9. Generated an embedding matrix using Word2Vec with the tokenizer.
10.Trained an MCNN model with multiple layers.
11.Evaluated the accuracy of the trained model on the testing dataset, stored
   as accuracy using model.evaluate().
```

Fasttext:

Pseudocode:

```
1. Connected the dataset to Colab through Google Drive.
2. Loaded the dataset, named reviews_df, from text_reviews.csv.
3. Implemented text cleaning through the function clean(text).
4. Brought in TensorFlow and Keras libraries.
5. Established a dictionary with keys for text (reviews) and sentiments
   (labels).
6. Initialized a tokenizer.
7. Divided the data into training and test sets with a 70:30 ratio.
8. Incorporated the wiki-news dataset for fasttext embedding.
9. Developed an embedding matrix utilizing fasttext.
10.Conducted training on an MCNN model with multiple layers.
11.Assessed the accuracy of the trained model on the testing dataset, stored
   as accuracy through model.evaluate().
```

BERT:

Pseudocode:

```
1. Established the connection between the dataset and Colab using Google Drive.
2. Installed BERT.
3. Loaded the dataset, reviews_df, from the file text_reviews.csv.
4. Implemented text cleaning using the function def clean(text){}.
5. Imported TensorFlow and Keras libraries.
6. Created a dictionary with keys 'text' for reviews and 'sentiments' for labels.
7. Initialized a BERT tokenizer.
8. Shuffled the dataset.
9. Created a batched Dataset for model training.
10. Divided the data into training and test sets with a 70:30 ratio.
11. Trained the MCNN model with multiple layers.
12. Evaluated the accuracy of the trained model on the testing dataset, stored as accuracy using model.evaluate().
```

RNN

Word2vec:

Pseudocode:

```
1. Connected the dataset to Colab via Google Drive.
2. Loaded the dataset, designated as reviews_df, from text_reviews.csv.
3. Implemented text cleaning through the function clean(text){}.
4. Imported TensorFlow and Keras libraries.
5. Established a dictionary with keys for text (reviews) and sentiments (labels).
6. Initialized a tokenizer.
7. Divided the data into training and test sets with a 70:30 ratio.
8. Created vocabulary size using the tokenizer.
9. Generated an embedding matrix using Word2Vec with the tokenizer.
10. Trained an RNN model with multiple layers.
11. Evaluated the accuracy of the trained model on the testing dataset, stored as accuracy using model.evaluate().
```

Fasttext:

Pseudocode:

```
1. Connected the dataset to Colab through Google Drive.
2. Loaded the dataset, named reviews_df, from text_reviews.csv.
3. Implemented text cleaning through the function clean(text).
4. Brought in TensorFlow and Keras libraries.
5. Established a dictionary with keys for text (reviews) and sentiments (labels).
6. Initialized a tokenizer.
7. Divided the data into training and test sets with a 70:30 ratio.
8. Incorporated the wiki-news dataset for fasttext embedding.
9. Developed an embedding matrix utilizing fasttext.
10. Conducted training on an RNN model with multiple layers.
11. Assessed the accuracy of the trained model on the testing dataset, stored as accuracy through model.evaluate().
```

HAN

Word2vec:

Pseudocode:

```
1. Connected the dataset to Colab via Google Drive.
2. Loaded the dataset, naming it reviews_df, from text_reviews.csv.
3. Implemented text cleaning with the function def clean(text){}.
4. Imported Theano and Keras libraries.
5. Created a dictionary with keys for text ("reviews") and sentiments ("labels").
6. Initialized a tokenizer using Word2Vec.
7. Split the data into training and test sets with a 70:30 ratio.
8. Created both a sentence tokenizer and a word tokenizer.
9. Generated an embedding matrix using both tokenizers.
10. Trained the Hierarchical Attention Network (HAN) model.
11. Evaluated the accuracy of the trained model on the testing dataset, stored as accuracy through model.evaluate().
```

Fasttext:

Pseudocode:

1. Connected the dataset to Colab using Google Drive.
2. Loaded the dataset, assigning it to reviews_df, from text_reviews.csv.
3. Implemented text cleaning using the function def clean(text){}.
4. Imported Theano and Keras libraries.
5. Created a dictionary with keys for text ("reviews") and sentiments ("labels").
6. Initialized a tokenizer.
7. Imported the wiki-news dataset for word embedding.
8. Split the data into training and test sets with a 70:30 ratio.
9. Created both a sentence tokenizer and a word tokenizer.
10. Generated an embedding matrix using both tokenizers.
11. Trained the Hierarchical Attention Network (HAN) model.
12. Evaluated the accuracy of the trained model on the testing dataset, stored as accuracy through model.evaluate().

Glove:

Pseudocode:

1. Mounted the dataset to Colab using Google Drive.
2. Read the dataset, assigning it to reviews_df, from the file text_reviews.csv.
3. Implemented text cleaning through the function def clean(text){}.
4. Imported Theano and Keras libraries.
5. Created a dictionary with keys 'text' for reviews and 'sentiments' for labels.
6. Established a tokenizer using Word2Vec.
7. Split the data into training and test sets with a 70:30 ratio.
8. Created both a sentence tokenizer and a word tokenizer.
9. Developed an embedding matrix using both tokenizers.
10. Trained the Hierarchical Attention Network (HAN) model.
11. Evaluated the accuracy of the trained model on the testing dataset, stored as accuracy through model.evaluate().

RMDL

BERT:

Pseudocode:

1. Established the connection between the dataset and Colab using Google Drive.
2. Installed BERT.
3. Loaded the dataset, reviews_df, from the file text_reviews.csv.
4. Implemented text cleaning using the function def clean(text){}.
5. Imported TensorFlow and Keras libraries.
6. Created a dictionary with keys 'text' for reviews and 'sentiments' for labels.
7. Initialized a BERT tokenizer.
8. Shuffled the dataset.
9. Created a batched Dataset for model training.
10. Divided the data into training and test sets with a 70:30 ratio.
11. Trained the RMDL model with multiple layers.
12. Evaluated the accuracy of the trained model on the testing dataset, stored as accuracy using model.evaluate().

Fasttext:

Pseudocode:

13. Connected the dataset to Colab using Google Drive.
14. Loaded the dataset, assigning it to reviews_df, from text_reviews.csv.
15. Implemented text cleaning using the function def clean(text){}.
16. Imported Theano and Keras libraries.
17. Created a dictionary with keys for text ("reviews") and sentiments ("labels").
18. Initialized a tokenizer.
19. Imported the wiki-news dataset for word embedding.
20. Split the data into training and test sets with a 70:30 ratio.
21. Created both a sentence tokenizer and a word tokenizer.
22. Generated an embedding matrix using both tokenizers.
23. Trained the RMDL model.
24. Evaluated the accuracy of the trained model on the testing dataset, stored as accuracy through model.evaluate().

Glove:

Pseudocode:

```
1. Established the dataset connection to Colab using Google Drive.
2. Installed RMDL.
3. Loaded the dataset, naming it reviews_df, from text_reviews.csv.
4. Implemented text cleaning with the function def clean(text){}.
5. Imported TensorFlow and Keras libraries.
6. Created a dictionary with keys for text ("reviews") and sentiments ("labels").
7. Imported the GloVe 6B dataset for word vectors.
8. Shuffled the dataset.
9. Created word vectors.
10. Generated an embedding matrix.
11. Split the data into training and test sets with a 70:30 ratio.
12. Trained the RMDL model with DNN, CNN, and RNN as three internal models.
13. Evaluated the accuracy of the trained model on the testing dataset, stored as accuracy using model.evaluate().
```