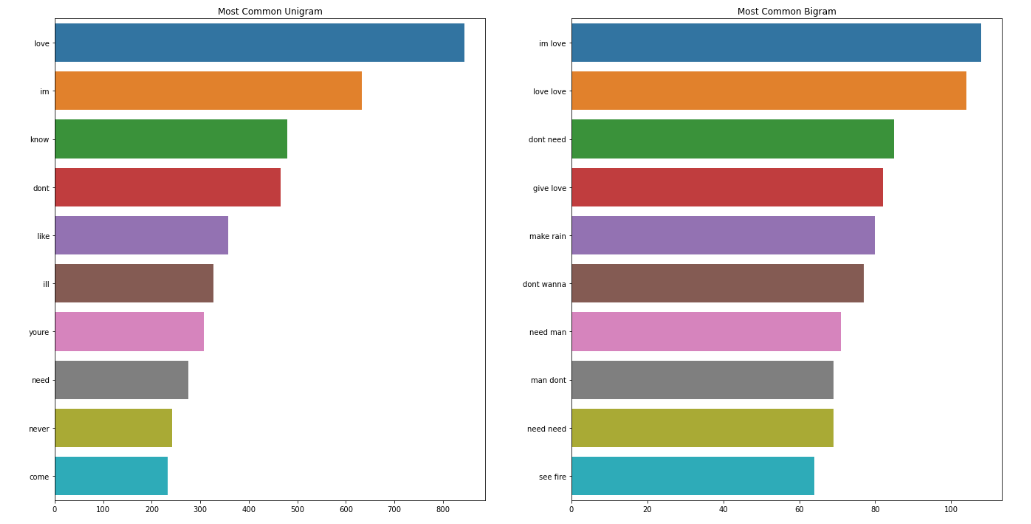
**Ed Sheeran Songs generation**

* **EDA –**

**1) Most Common Unigram and Bigram**



Love is the most common word in the Ed Sheeran’s song corpus. He mainly writes about love and compassion.

**2) Word Cloud**



To find out more details about the corpus we created word cloud as well. Words live “love”,” loving”,” Darling” had been used multiple times.

**Conclusion –**

The EDA tells that the ED Sheeran songs is more about love and compassion

* **Preprocessing steps followed -**
  1. Lowercase the Lyrics column
  2. Split the column word-wise and created a corpus of words
  3. Organised the corpus into sequences of tokens of length 51 (51-grams)
  4. Created a text file containing all these sequences
  5. Number of sequences :
     + Total – 62380
     + Train – 50628
     + Test – 11701
* **Tokenize & Embeddings -** 
  1. Createdavocabulary consisting of unique words in the corpus

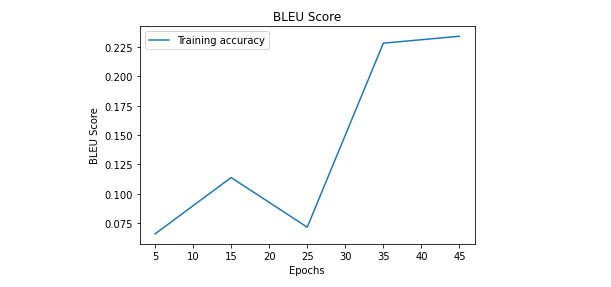
- total unique words : 3043

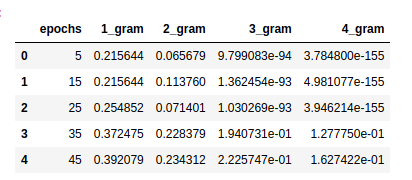
* 1. Assigned each unique word to a unique index using Keras Tokenizer
  2. Converted the sequences list to respective word-index list
  3. GloVe embedding :
     + Used glove.6B.100d.txt file of 100 dimensions emedding
     + Created an embedding matrix of the above vocabulary [3043 x 100]
  4. word2vec embedding : Used word2vec model to embed our vocabulary with the following settings
     + min\_count = 1
     + size = 100
     + window = 5
  5. Splitted the word-index array for training and converted y to categorical:
     + X\_shape – (50628, 50)
     + y\_shape – (50628, 3043)
* **Models -** 
  1. Embedding layer :
     + For word-index – input\_dim (the vocab size that we choose) = 3043; output\_dim (the number of dimensions we wish to embed) = 50; input\_length (length of sequence) = 50
     + For GloVe & word2vec embedding - input\_dim (the vocab size that we choose) = 3043; output\_dim (the number of dimensions we wish to embed) = 100; input\_length (length of sequence) = 50; weights = [embedding\_matrix]
     + One-hot encoding – input\_length (length of sequence) = 40, Vocab Size – 76, Indexing - Character type
  2. RNN : 2 LSTM layers with 100 nodes each
  3. Bidirectional RNN : 2 Bidirectional LSTM layers with 100 nodes each
  4. Multi-layer RNN: 3 LSTM layers with 100 nodes each
  5. Dense layer :
     + One dense layer with 100 units & activation = ‘relu’
     + Other dense layer with units of our vocabulary size & activation = ‘softmax’
  6. Compiled using :
     + loss = Categorical Crossentropy
     + optimizer = Adam
     + metrics = Accuracy
  7. Saved the models & tokenizer pickle file



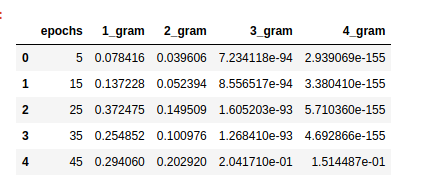
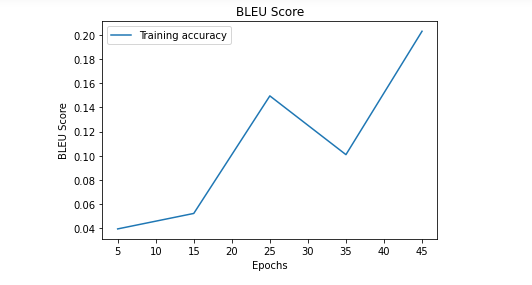
* **BLEU Score -**
  1. Loaded the sequences present in the test file
  2. Randomly selected a sequence from the loaded sequences
  3. Generated a new sequence using the above language models & tokenizer
  4. Compared the new sequence with the actual sequence

BLEU plot & df for Bidirectional RNN using glove embeddings:





BLEU plot & df for Bidirectional RNN using word2vec embeddings:



* **Lyrics generated using inference code:**

