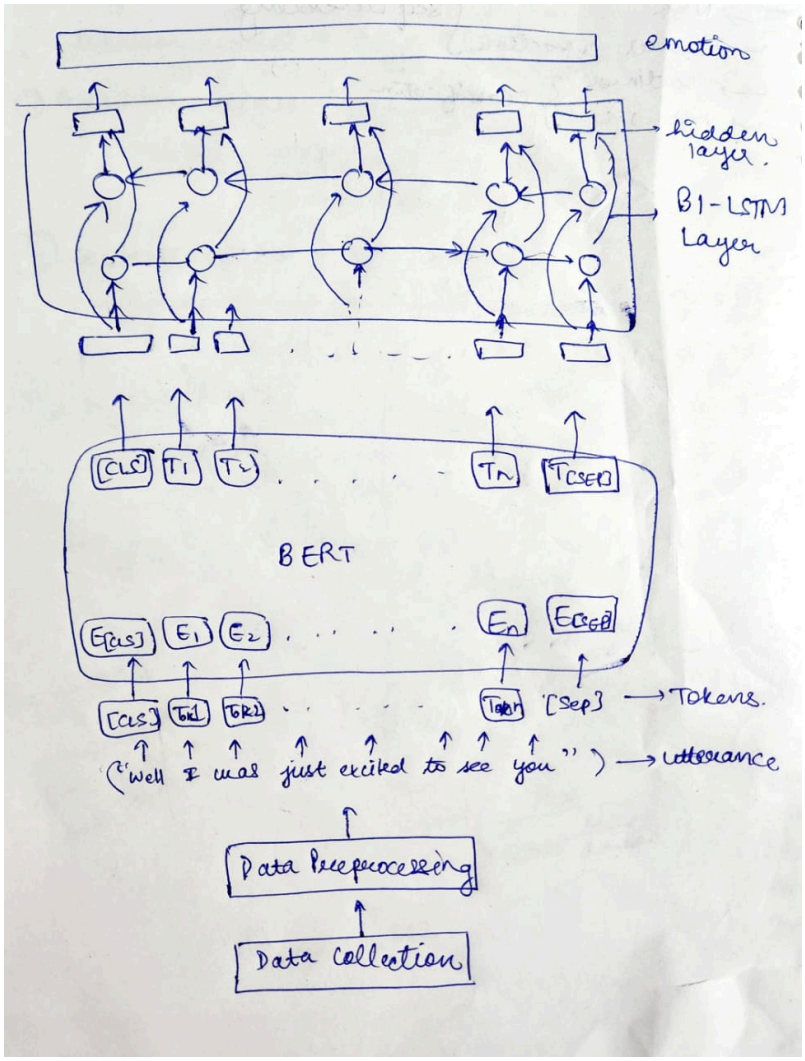
NLP Assignment 4 Report
Group 30

Model 1: **LSTM with BERT Embeddings**

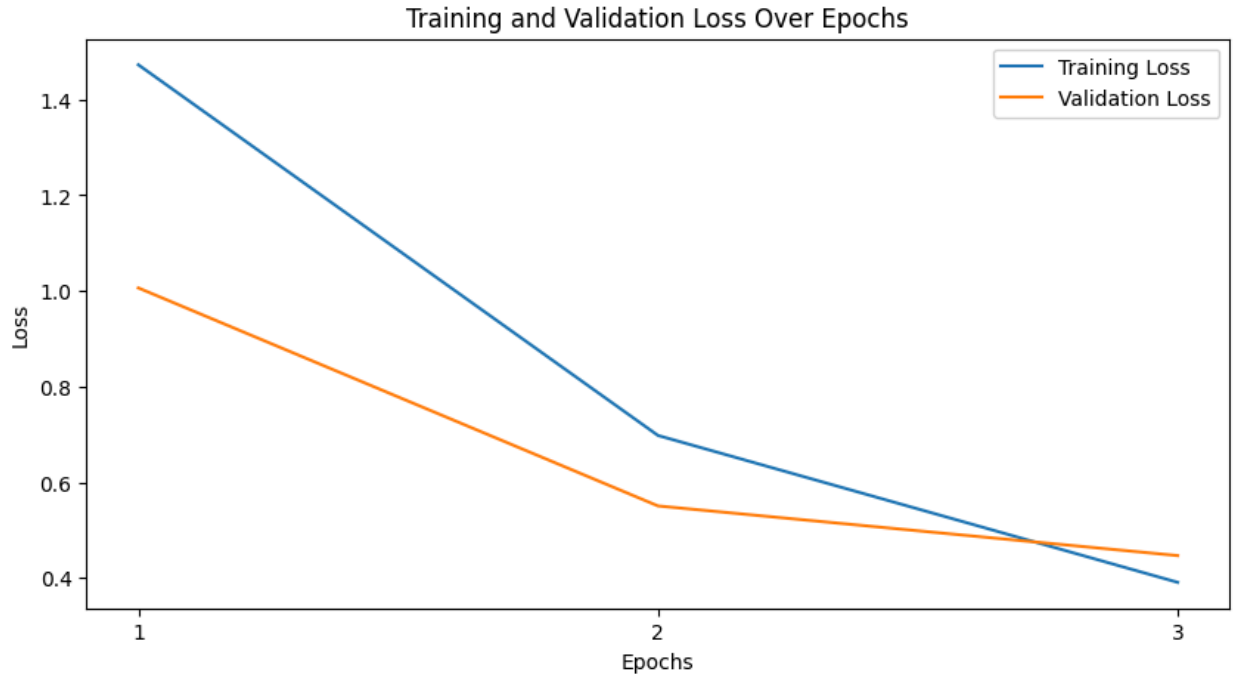
Model Architecture Diagram:



Intuition behind model:
The model uses BERT embeddings for providing contextual embeddings for each token in the utterances. These embeddings are then sent to the bidirectional LSTM layer which captures both forward and backward dependencies in the sequence. There are 2 distinct output layers. The first one is for emotion classification which predicts the emotion for each utterance. The second one is for flip detection, which has a binary output (flip or no flip) that checks if there is a change in a speaker's emotion from their last utterance. The data preparation pre-processing is done in the "EmotionDataset" class. We used a representative subset of the train data for training purposes because of computational constraints to get a manageable training time.

Cross-entropy loss was used for the emotion classification task to handle multi-class outputs, and binary cross-entropy loss was used for flip detection to handle binary output. For evaluation metrics, we have reported the accuracy, precision, recall and f1 scores for both emotion recognition and flip detection.

Plot:



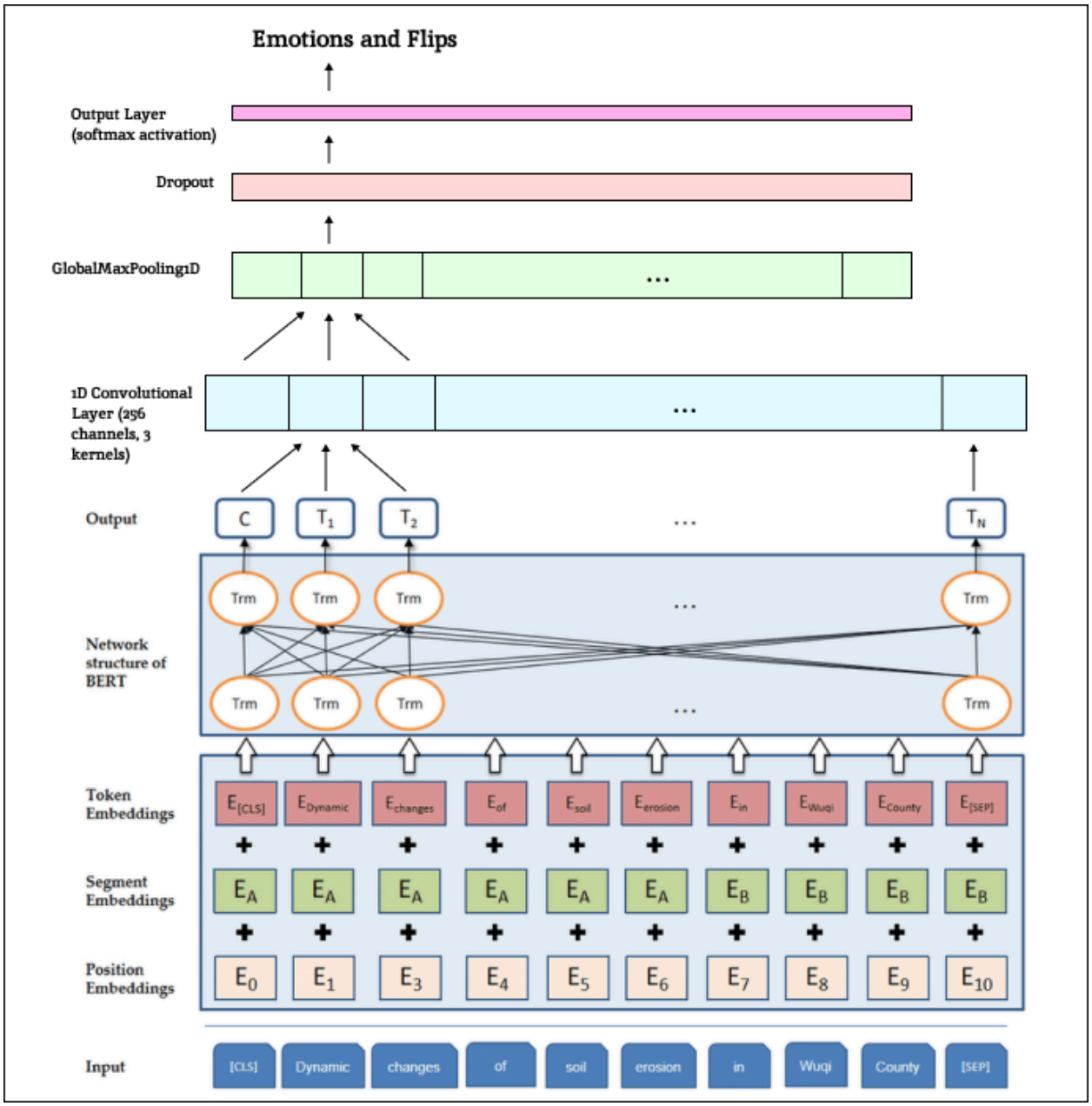
Evaluation Metrics:
{'Emotion Accuracy': 0.9315782256958728, 'Emotion Precision': 0.913664130022544, 'Emotion Recall': 0.9095073461976168, 'Emotion F1': 0.9115089494898415, 'Flip Accuracy': 0.9184149184149184, 'Flip Precision': 0.9079903147699758, 'Flip Recall': 0.9115770282588879, 'Flip F1': 0.9097801364670205}

Model 2: BERT Embeddings + 1D CNN + Pooling

The second model was trained using BERT's rich contextual embeddings and layering them with a 1D Convolutional Layer along with pooling that beat the metrics of Bi-LSTM model. Given that GRU was used in the paper provided and other RNNs could not beat the LSTM model, this architecture was finalised. The 1D Convolutional layer with GlobalMaxPooling1D can efficiently extract local patterns and features from the BERT embeddings. This is particularly beneficial for tasks where capturing short-range dependencies or local context (extracting of emotion using words) is crucial. The use of GlobalMaxPooling1D was followed by a dropout layer reduced overfitting by focusing on the most important features while introducing regularization through

dropout. This regularization can prevent the model from memorizing noise in the training data, leading to better generalization.

Architecture:



Plot:

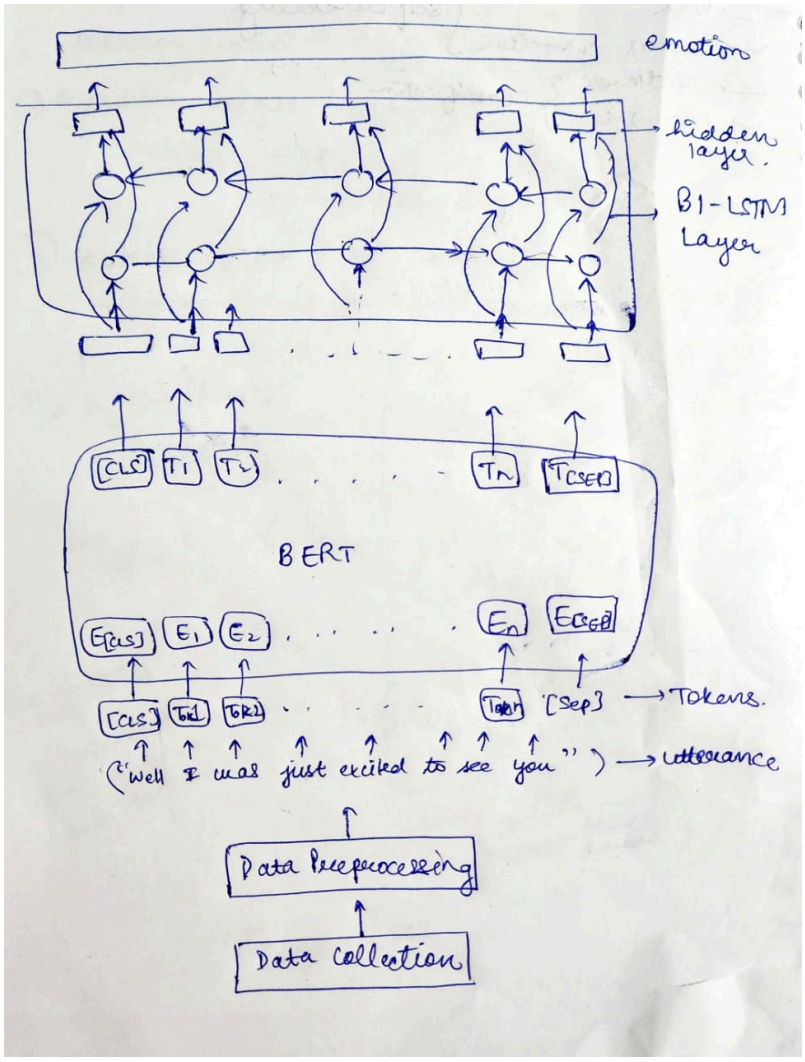
Evaluation Metrics:
{'Emotion Accuracy': 0.9411764705882353, 'Emotion Precision': 0.93326849555636, 'Emotion Recall': 0.9268612605975199, 'Emotion F1': 0.9297811852903068, 'Flip Accuracy': 0.9262306321129851, 'Flip Precision': 0.8984081041968162, 'Flip Recall': 0.9431783652385293, 'Flip F1': 0.9202490364660539}

Model 3 & 4
The data loader for model 3 and 4 is the same.
It takes a whole dialogue and tokenizes each utterance in it using pre-trained BertTokenizer (bert-base-uncased). Then a new tensor(utterances) is formed whose sub elements are the emotion of the utterance appended with the tokenized utterance.
Eg. [sad My day has been miserable]
The attention mask is also modified to handle the appending of emotion by ensuring that the mask of the emotion should be 1.

Bert tokenizer adds one position embedding and another positional embedding is added to each utterance in the new tensor to encode the position of each utterance wrt each other. Basically every token in a sentence has one embedding value which is added here. This dataset enables the model to get all the data necessary for training accurately for emotion flip data.

Model 3

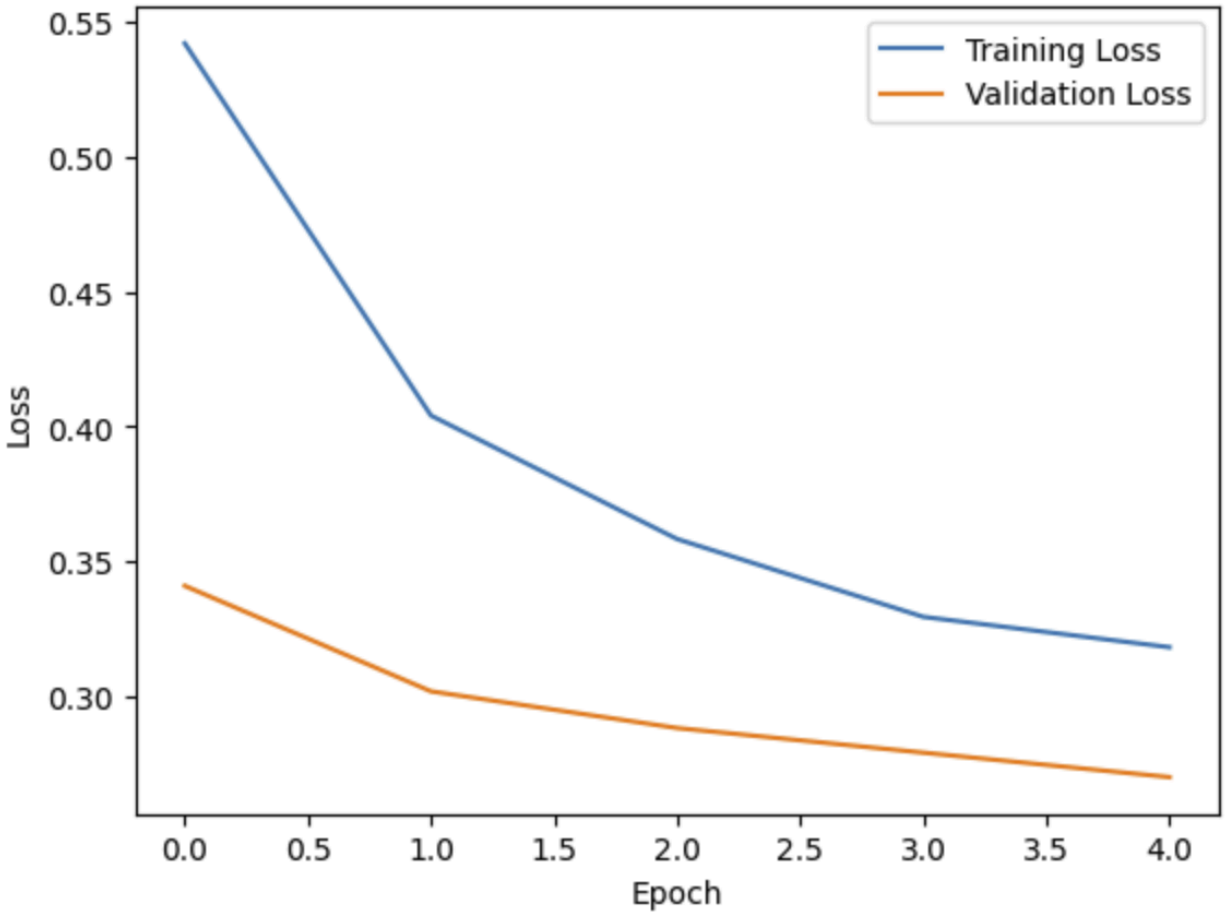
The architecture of model 3 is same as model 1:



The data which we get from the dataloader is first flattened to send to BERT, i.e., the tensor(utterances) which contains the list of all the tokenised utterances is flattened.
The flattened data is sent to pre-trained BertModel (bert-base-uncased) whose output is then sent to LSTM which is then sent to a Fully connected layer which makes the prediction of which utterance is a trigger or not. We did this to compare with the results of part 1. BERT is used to generate contextual embeddings, it is pretrained on a large corpus so it is useful for out tasks.

Loss Function: BCEWithLogitsLoss for binary classification
Optimiser: Adam

Loss Plot:



F1 Score: 0.3893033328266201

Model 4

The Architecture of the model:

This model takes the input which is a 2D matrix as explained above which is then sent into a 2D convolution layer, the output of which is sent to the ReLU layer and then max-pooling is done on it.

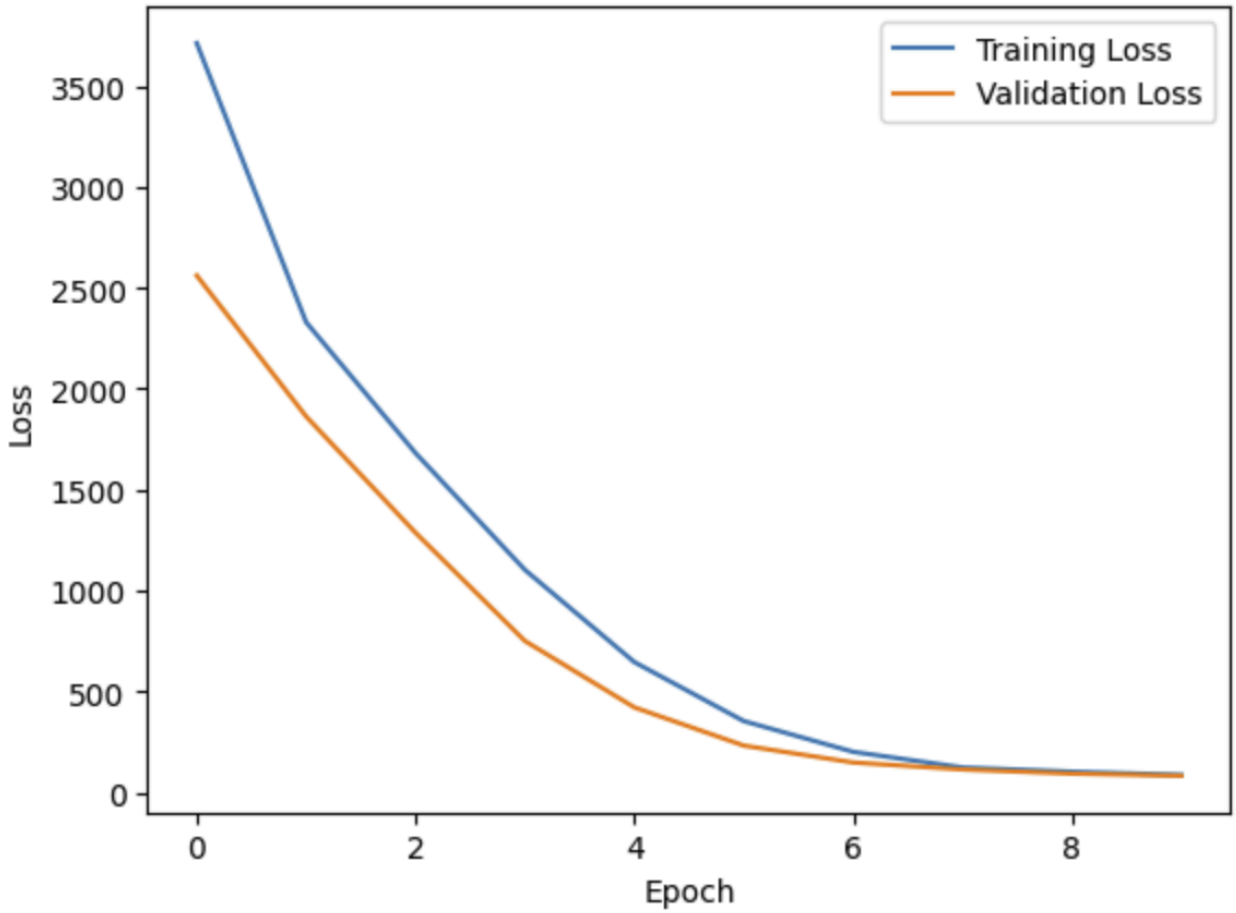
The 2D convolution layer was the first intuition from a 2D matrix input and ReLU is a common activation function used. Max Pooling to reduce output dimension. The 2 convolution layers are used for feature extraction.

First fully connected layer
64*7*7->128
Second fully connected layer
128->1 per utterance

Which is then finally flattened and sent to a fully connected layer with ReLU activation function, whose output is sent to another fully connected layer which makes the prediction if the utterance is a trigger or not. 2 fully connected layers are used.

Loss Function: CrossEntropyLoss
Optimiser: Adam

Loss Plot:



F1 Score: 0.7623529196166777
Accuracy: 0.7526722090261283