Programming Assignment 3

Yogesh Vijay Deokar

EE17B006

1 Part A

1.1 LSTM model

- Consider a sentence **'x1 x2 x3 x4 x5'**.
- For generating data, consider the word **x5**. The sequence of context words **x1, x2 , x3, x4** was input sequence for LSTM model and the expected output was **x5**.
- Similarly, an input sequence of previous four context words were generated for all the words in data.
- The first layer of the model was **embedding layer**. This was followed by two **LSTM** layers and and the end **Fully connected linear layer**.
- For the embedding layer , input length is vocabulary size and output length is embedding dimension that we want. This is then passed through two LSTM layers. The outputs are then stacked together and passed to a fully connected linear. The output size of this layer is vocabulary size. This output is then compared with the next word in the sequence to find the loss.
- Below is the **t-SNE** plot of most frequent 125 words in the vocabulary.
- As we can see in the figure, words **zero, one , two, three, four, five, six, seven , eight and nine** are placed very close together.
- Also words such as **english, british, american, french** etc are very close to each other.
- Words such as **government, military** etc are close to each other.

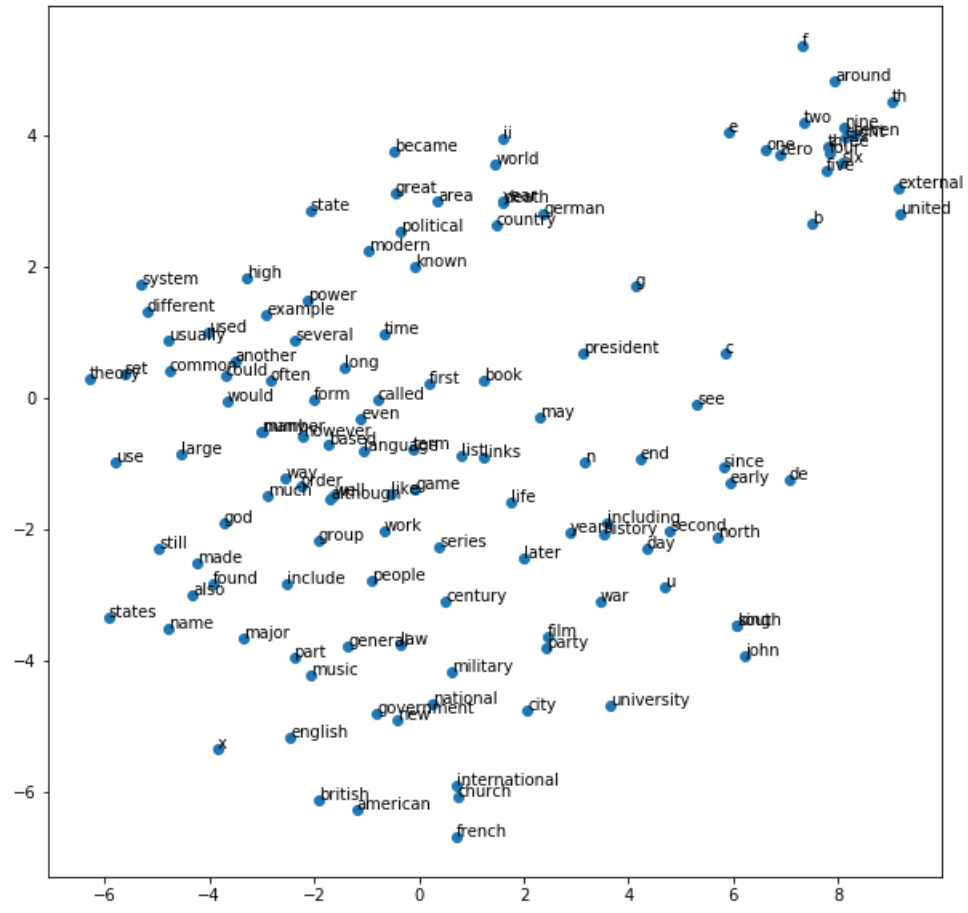

Figure 1: LSTM model t-SNE plot for 125 most frequent words

1.2 Skipgram model

- In skipgram model, **negative sampling** was used in data generation.
- Consider sentence **'x1 x2 x3 x4 x5 x6 x7'**.
- For the above sentence consider the centre word **x4**. If we choose a context

size of 2, the words in context window of **x4** are **x2, x3, x5, x6**. So **x2, x3, x5, x6** were labelled as **1**.

- For negative sampling, some random words were choosen from the entire dataset. These words were labelled as **0**.
- In skipgram model, the context word and the main word are given as input into seperate embedding layers.
- The embedding layer outputs are then used to calculate the similarity between these two using a dot product.
- This dot product output is passed to sigmoid function. This sigmoid output is compared against the assigned labels to compute the loss.
- Below is the plot of most frequent 100 words in the vocabulary.
- As we can see in the figure, words **six, nine, one, zero, two, four seven** etc are placed together.
- We can also see words such as **would, may** and **french, english** etc are close to each other.

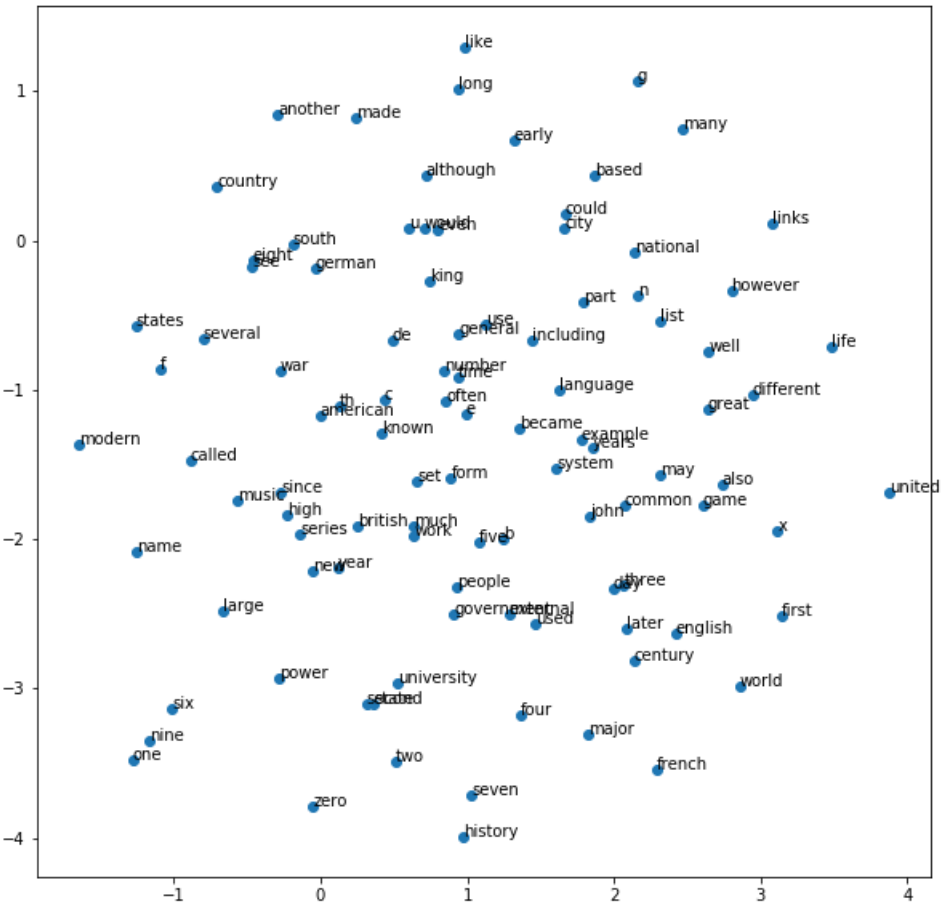

Figure 2: Skipgram model TSNE plot for 100 most frequent words

1.3 Bag-of-words model

- Consider the sentence **'x1 x2 x3 x4 x5 x6 x7'**.
- For the above sentence consider the centre word **x4**. If we choose a context size of 2, the words in context window of **x4** are **x2, x3, x5, x6**.

- The First layer of the model is an embedding layer. The context words are passed to this layer. The output of the embedding layer is averaged out and passed to a fully connected liner layer which has an output size same as vocabulary size. This output is then compared against the centre word to calculate the loss.
- Below is the plot of most frequent 100 words in the vocabulary.
- As we can see in the plot, **the numbers - one, two, three, four, five, six, seven etc** are close to each other.
- Also the words **french, british, english etc** are close to each other.

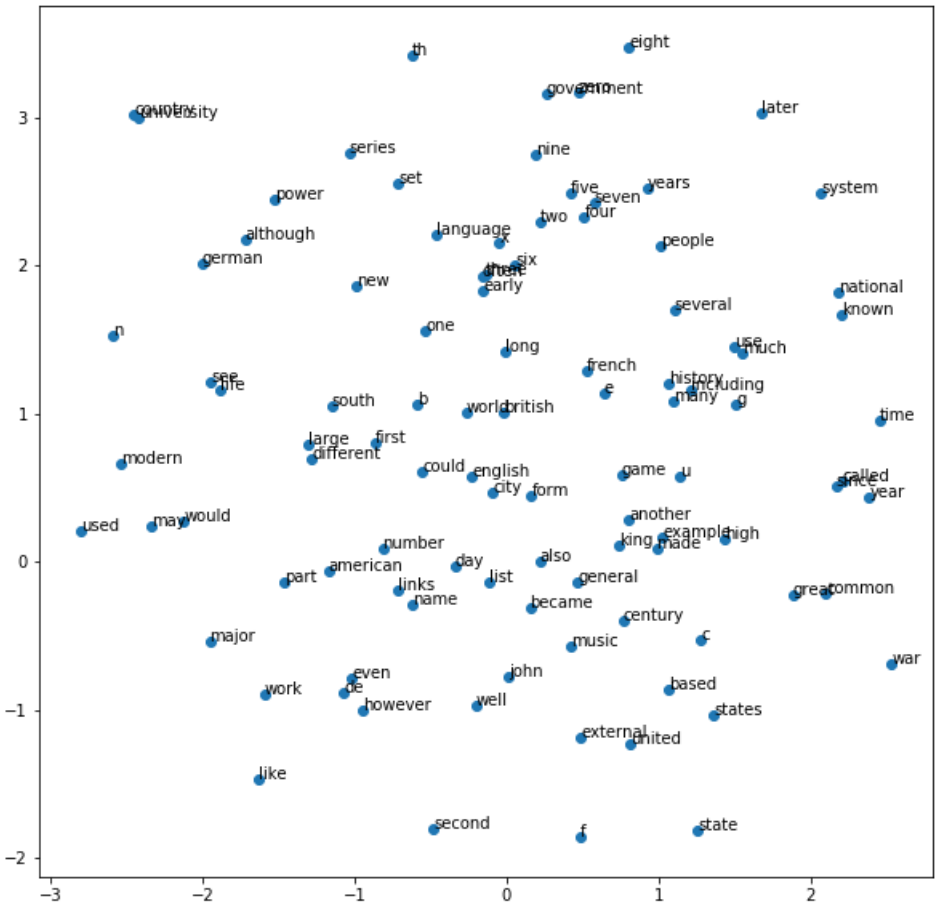

Figure 3: CBOW model TSNE plot for 100 most frequent words

2 Part B : Sentiment Analysis

- For sentiment analysis **LSTM** network was used. There were two LSTM layers in the network.
- All the sentences were made of same length by either padding with a zero

vector or stripping the extra words.

- Consider the sentence **'x1 x2 x3 x4 x5'**. The word2vec embeddings for these sequence of words were fed to the LSTM network.
- The output of the model was of dimension 5 representing the probabilities of belonging to one of the five given classes.
- The model was trained on the three networks above and also on pre-trained glove word embeddings and Gensim word2vec models.

2.1 Model trained on my Skipgram, LSTM and CBOW embeddings :-

- In the table below is the validation accuracy for different epochs for skip-gram, CBOW and LSTM models.
- All the three moels have embedding dimension of 100.

Epoch	**Skipgram**	**CBOW**	**LSTM**
10	0.566339	0.571457	0.59009
20	0.573729	0.592563	0.601623
30	0.57877	0.607475	0.604485
40	0.588723	0.611064	0.605681
50	0.613973	0.614994	0.619953
Test Accuracy	0.602776	0.606364	0.614656

Table 1: Validation Accuracy For different epochs

2.2 Model trained on pretrained Glove embeddings :-

- Glove embeddings of vector size 100, 200 and 300 were used for training the sentiment analysis model.
- The following table summarizes the obtained results

Epoch	**100**	**200**	**300**
10	0.5894	0.5909	0.6148
20	0.6031	0.6126	0.6192
30	0.6098	0.6152	0.6267
Test Accuracy	0.5952	0.6101	0.6211

Table 2: Validation Accuracy For Glove embeddings

2.3 Model trained on Gensim Word2Vec

- These models were trained using gensim's word2vec on text8 dataset for different values of embedding dimensions.
- As we can see in the table, as the embedding dimension increases, the Validation accuracy also increases after each epoch.

Epoch	**Embedding Dim 25**	**Embedding Dim 100**	**Embedding Dim 200**	**Embedding Dim 300**
5	0.5333	0.5631	0.5716	0.5774
10	0.5502	0.5787	0.5929	0.5944
15	0.5706	0.592	0.5941	0.6051
20	0.5766	0.5965	0.5994	0.6101
25	0.5804	0.6012	0.6019	0.6183

Table 3: Validation Accuracy for models trained on Gensim Word2vec