Opinion Analysis and Topic Modeling of Amazon Reviews

Natural Language Processing

CS F429

BITS Pilani

Hyderabad Campus

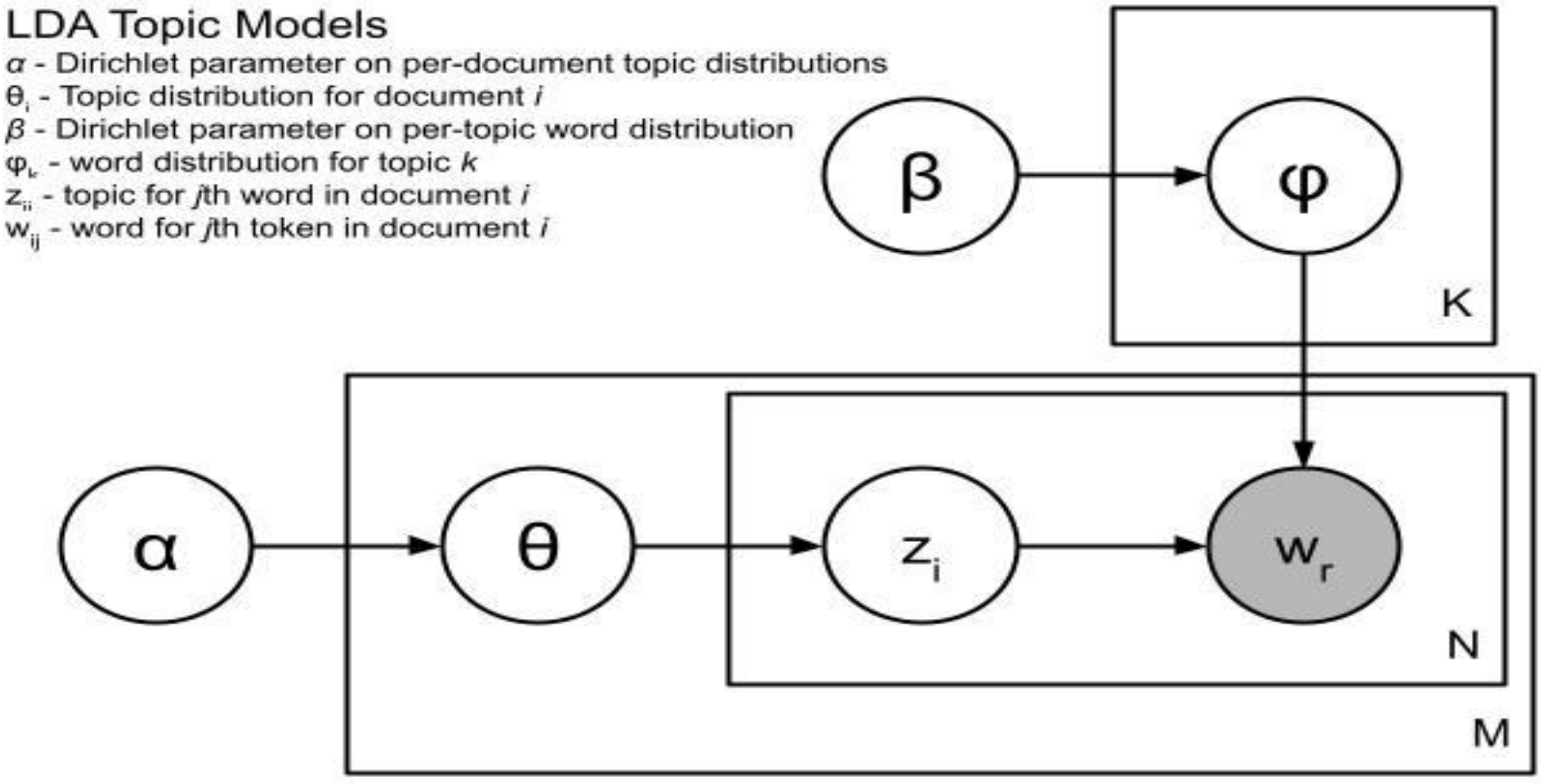
Introduction

- The growing importance of E-commerce in today's world
- Customer reviews help companies analyse their target customer base and improve their business model
- Natural Language Processing - A tool to gauge customer feedback

Theory: Topic Modeling

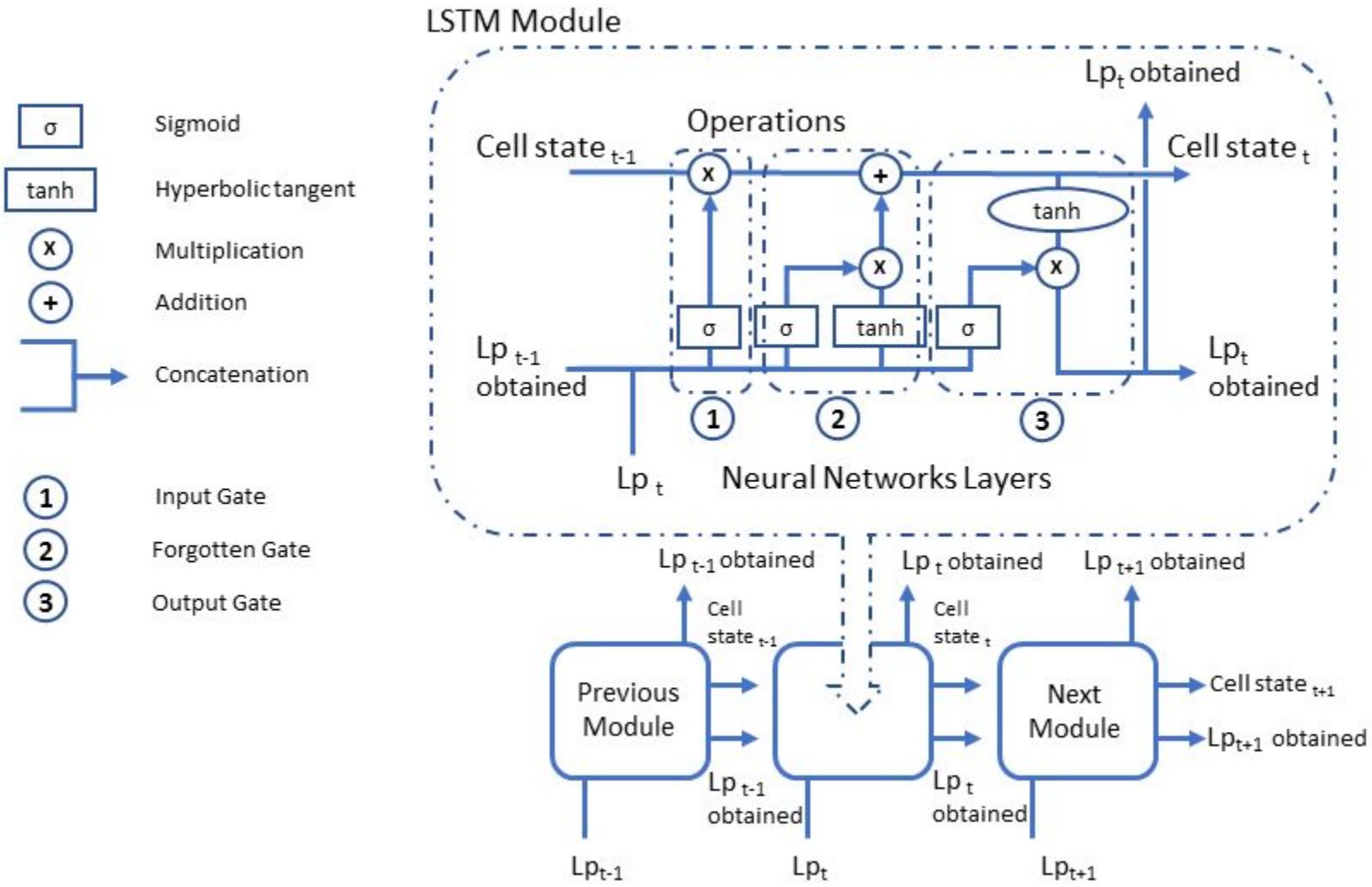
- A machine learning technique that automatically analyzes text data to determine cluster words for a set of documents.
- Unsupervised technique: absence of labelled data
- LDA: mapping each document in the corpus to a set of topics which covers most of the words in the document.
- Hyperparameters: alpha, beta and the number of topics
- The output of the algorithm is a vector that contains the coverage of every topic for the document being modeled.

Theory: Topic Modeling



Latent Dirichlet Allocation Plate Notation

Sentiment Ananlysis Theory: LSTM



Why LSTM?

- LSTM handles long term dependencies in the sentences better than RNNs because it retains only the relevant information.
- Vanishing Gradient Problem
- Exploding Gradient Problem
- Gated Cell

Dataset

- Dataset used is Cellphones and Accessories Review dataset from Stanford.
- We took a subset(11,00,000) of the original dataset containing 11,00,00,000 reviews.
- We took this subset to aid in faster training and more tractability.

Link : https://nijianmo.github.io/amazon/index.html

Data Preprocessing

- Tokenization:
  It is the process of breaking a whole sentence into individuals such as symbols, keywords, and phrases known as a token. In tokenization some characters are removed.

- Stop words:
  Stop words are those objects in a sentence which are not required in any segment of text mining, so usually, these sentences are removed to increase the efficiency of analysis. The idea is simply removing the words that occur commonly across all the documents in the corpus.

Data Preprocessing (Cont.)

- Stemming
  It is the process of alloying the modified forms of a word into a common interpretation. It is a technique used for information retrieval (IR) in text processing based on the statement in documents

- Lemmatization
  In linguistics, it is the process of grouping together the inflected forms of a word so they can be analyzed as a single item, identified by the word's lemma, or dictionary form.

Methodology: Topic Modelling

- After preprocessing, vectorization was done to find the sparcisity matrix
- This matrix gives the percentage of non zero values in the word count matrix.
- The word count matrix gives the frequency of words appearing in a particular document i.e individual reviews.

Methodology: Topic Modelling (cont)

- sklearn library in python was used to prepare the following LDA model for topic modelling.

- The model was analysed using metrics like coherence score and perplexity.

```
lda_model = LatentDirichletAllocation(n_components=6,               # Number of topics
                                      max_iter=10,                  # Max learning iterations
                                      learning_method='online',
                                      random_state=100,             # Random state
                                      batch_size=10000,             # n docs in each learning
                                      evaluate_every = -1,          # compute perplexity every
                                      n_jobs = -1,                  # Use all available CPUs
                                     )
lda_output = lda_model.fit_transform(data_vectorized)

print(lda_model)  # Model attributes
```

Methodology: Topic Modelling (cont)

innovate achieve lead

- To optimise the number of topics Grid search method was employed.

- The optimal number of topics was found to be 6 which were:
  a. Screen
  b. Service
  c. Cover
  d. Battery
  e. User Experience
  f. Sound

- Further observations were made on the LDA.

Methodology: Sentiment Analysis

- Similar preprocessing was required for Sentiment Analysis as for Topic Modelling.
- Dataset was split into train and test in 80-20 ratio.
- GLOVE word embedding was used to convert textual data into numerical input.
- LSTM neural network was built using keras deep learning library in python.

Methodology: Sentiment Analysis (Cont.)

- This model included embedding input layer, hidden LSTM layer and output dense layer.
- The model was trained on the training dataset with batch size of 1000 for 3 epochs.
- Insample validation was done for 20% of the training data.
- The model was evaluated on accuracy metric and the loss function was Binary Cross Entropy Loss.

Evaluation Of LDA

1. Coherence Score:

   It measures the score wrt a single topic, by measuring the degree of semantic similarity between high probability words in each topic. These measurements help distinguish between topics that are semantically interpretable and topics that are artifacts of statistical inference.

2. Perplexity:

   It is a metric that measures the confusion a model faces when presented with new data. A lower perplexity score indicates better generalization performance.

$$PP(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_{i-1})}}$$

Our Results:

Perplexity: 810.3318102454269

Coherence Score: 0.6359785

Evaluation of Sentiment Analysis

- The LSTM model exhibited the following evaluation metrics:

```
Epoch 1/3
723/723 [==============================] - 1054s 1s/step - loss: 0.3214 - acc: 0.8853 - val_loss: 0.2210 - val_acc: 0.9098
Epoch 2/3
723/723 [==============================] - 1163s 2s/step - loss: 0.1971 - acc: 0.9187 - val_loss: 0.1896 - val_acc: 0.9200
Epoch 3/3
723/723 [==============================] - 1146s 2s/step - loss: 0.1759 - acc: 0.9280 - val_loss: 0.1741 - val_acc: 0.9275
7053/7053 [==============================] - 235s 33ms/step - loss: 0.1726 - acc: 0.9284
```

```
Test Score: 0.17262110114097595
Test Accuracy: 0.928405582904815
```

Results - Topic Modelling

- The trained LDA model was used to predict topics for unseen reviews.

- The results found were as follows

```
mytext = ["The flexibilty of the case was poor"]
topic, prob_scores = predict_topic(text = mytext)
index = np.argmax(prob_scores[0])
print(topic_names[index])
```

```
100% 1/1 [00:00<00:00, 33.07it/s]

100% 1/1 [00:00<00:00, 29.02it/s]

Phone Cover
```

Results - Topic Modelling

```
mytext = ["The capacity of the phone to retain power was exceptional, and its built was good"]
topic, prob_scores = predict_topic(text = mytext)
index = np.argmax(prob_scores[0])
print(topic_names[index])
```

```
100% 1/1 [00:00<00:00, 35.62it/s]
100% 1/1 [00:00<00:00, 27.04it/s]
Battery
```

```
mytext = ["The charging was efficient"]
topic, prob_scores = predict_topic(text = mytext)
index = np.argmax(prob_scores[0])
print(topic_names[index])
```

```
100% 1/1 [00:00<00:00, 30.03it/s]
100% 1/1 [00:00<00:00, 22.06it/s]
Battery
```

Results - Topic Modelling

```
mytext = ["The music quality was very good"]
topic, prob_scores = predict_topic(text = mytext)
index = np.argmax(prob_scores[0])
print(topic_names[index])
```

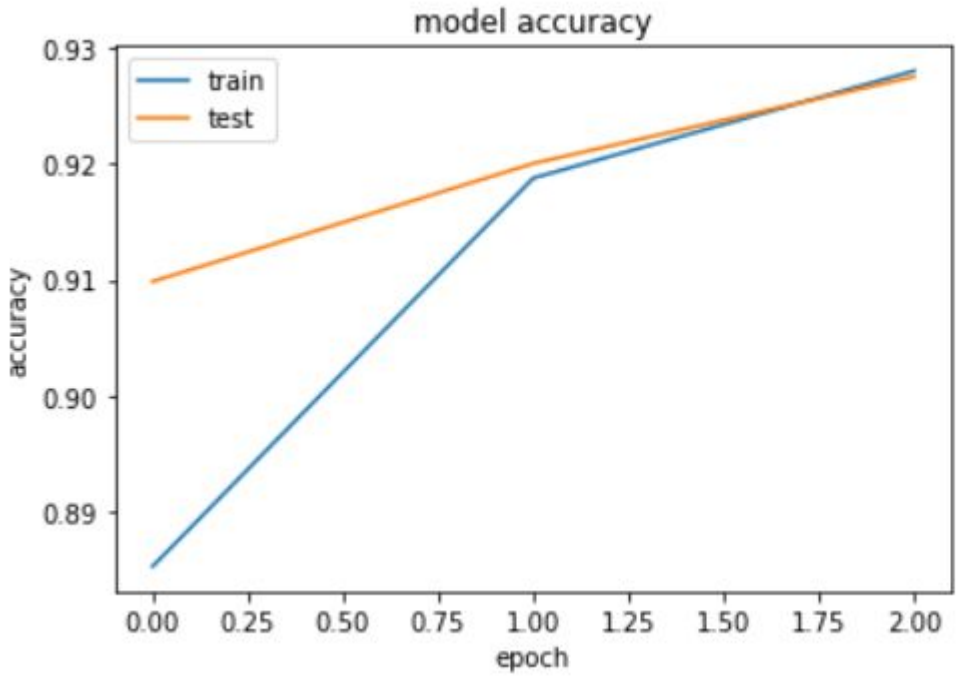
```
100% 1/1 [00:00<00:00, 37.86it/s]
100% 1/1 [00:00<00:00, 33.62it/s]
Sound
```

```
mytext = ["The quality was very good"]
topic, prob_scores = predict_topic(text = mytext)
index = np.argmax(prob_scores[0])
print(topic_names[index])
```

```
100% 1/1 [00:00<00:00, 37.37it/s]
100% 1/1 [00:00<00:00, 24.20it/s]
User Experience
```

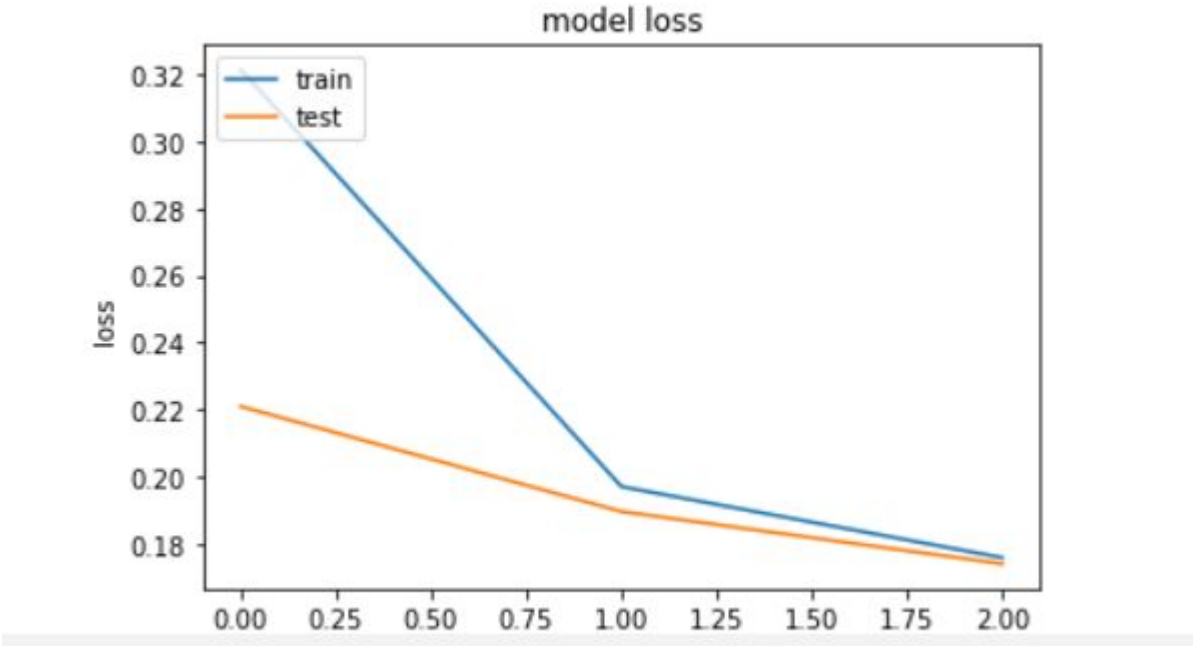
Results- Sentiment Analysis

- The graph for model accuracy was plotted using the matlplotlib python library.



Results- Sentiment Analysis

- The graph for model loss was plotted using the matlplotlib python library.



Conclusion

- The Topic Modeling provided substantial extraction of the important topics in the dataset, as is evident from the results.
- The accuracy of the sentiment analysis model is promising, proving that LSTM is an efficient paradigm for sentiment analysis.

Contributions

Shreyas Dixit	Sentiment Analysis Model
Dhruv Agrawal	Evaluation Metrics Of LDA and Sentiment Models and Results
Akshat Khaitan	Latent Dirichlet Allocation Model

Common Tasks :

1. Presentation
2. Hyperparameter Tuning

Thank You