Abstractive User Review Consolidation

Final Project Presentation

Group 12
Bhairavi Sameer Shah
Diya Liza Varghese
Michelle Elizabeth
Theres Mary Jose

Introduction

- Information is always available in plenty in the modern era of the Internet.
- Important to transform the data in a way that would help consumers understand it in the most comprehensive way.
- Limiting the content into precise and accurate points will reduce the overhead of processing unwanted information for consumers, and the difficulty of providing relevant data for the providers.
- Automated summary generators provide you with a summary of the input text that you provide to the system.
- The automated summary generator can find its application in different forms, in educational fields, content creation, E-commerce, marketing, etc.

Abstract

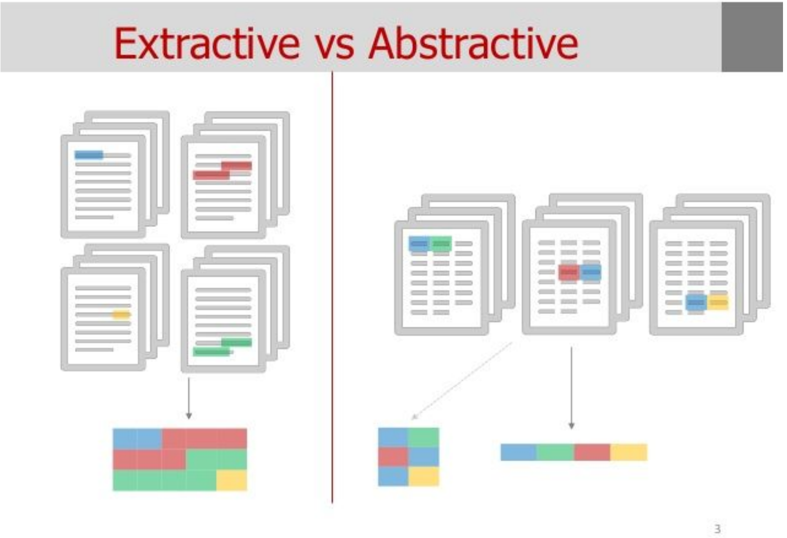
- E-commerce websites such as Amazon allow customers to leave reviews for various products.
- **Hundreds of reviews** for a single product - difficult to go through all the user reviews.
- Each review could be **lengthy and repetitive** - making it **confusing** for customer to make a well-informed decision after reading all the reviews.
- Therefore, automatic review summarization has a huge potential to help customers by providing an authentic summary of the reviews found online on various E-commerce sites.

Problem Statement

- The existing systems for summarization do not consider the **sentiment** of the user reviews which are necessary to produce sound reviews of products.
- Also, some methods consider only **a single document** for summarization while others do not consider **repetition of semantically equivalent words**.
- To overcome these issues, we introduce the system **Abstractive User Review Consolidation**.

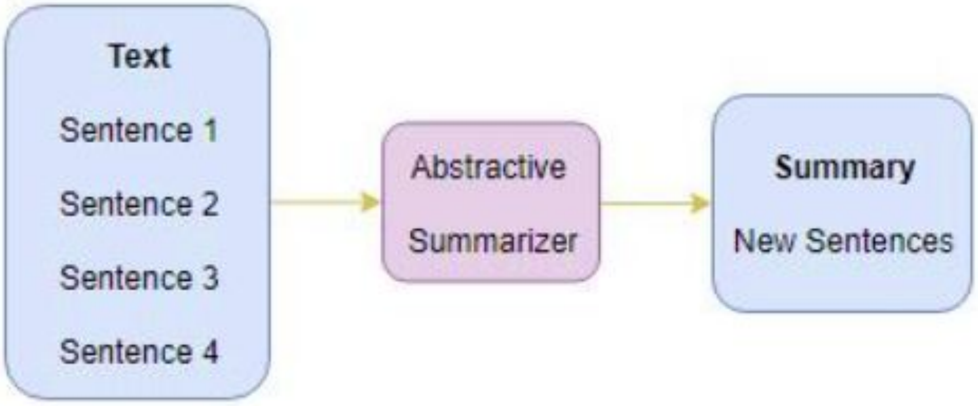
Proposed System

- We propose a **Sequence to Sequence (Seq2Seq) model with Attention Mechanism** which consists of an encoder, decoder, and attention layer to perform **abstractive summarization** of user reviews.
- The dataset includes user reviews of products from E-commerce websites.
- The tweets from social networking platforms like Twitter which are sentimentally rich are used to train a **Naive Bayes model** to identify the sentiment of the review text.
- We use **lemmatization** to avoid repetition of semantically equivalent words.



Abstractive Summarization

- New sentences are generated from the original text, in contrast to the extractive approach where only the sentences that were present are used.
- The sentences generated through abstractive summarization might not be present in the original text.



Why Abstractive Summarisation?

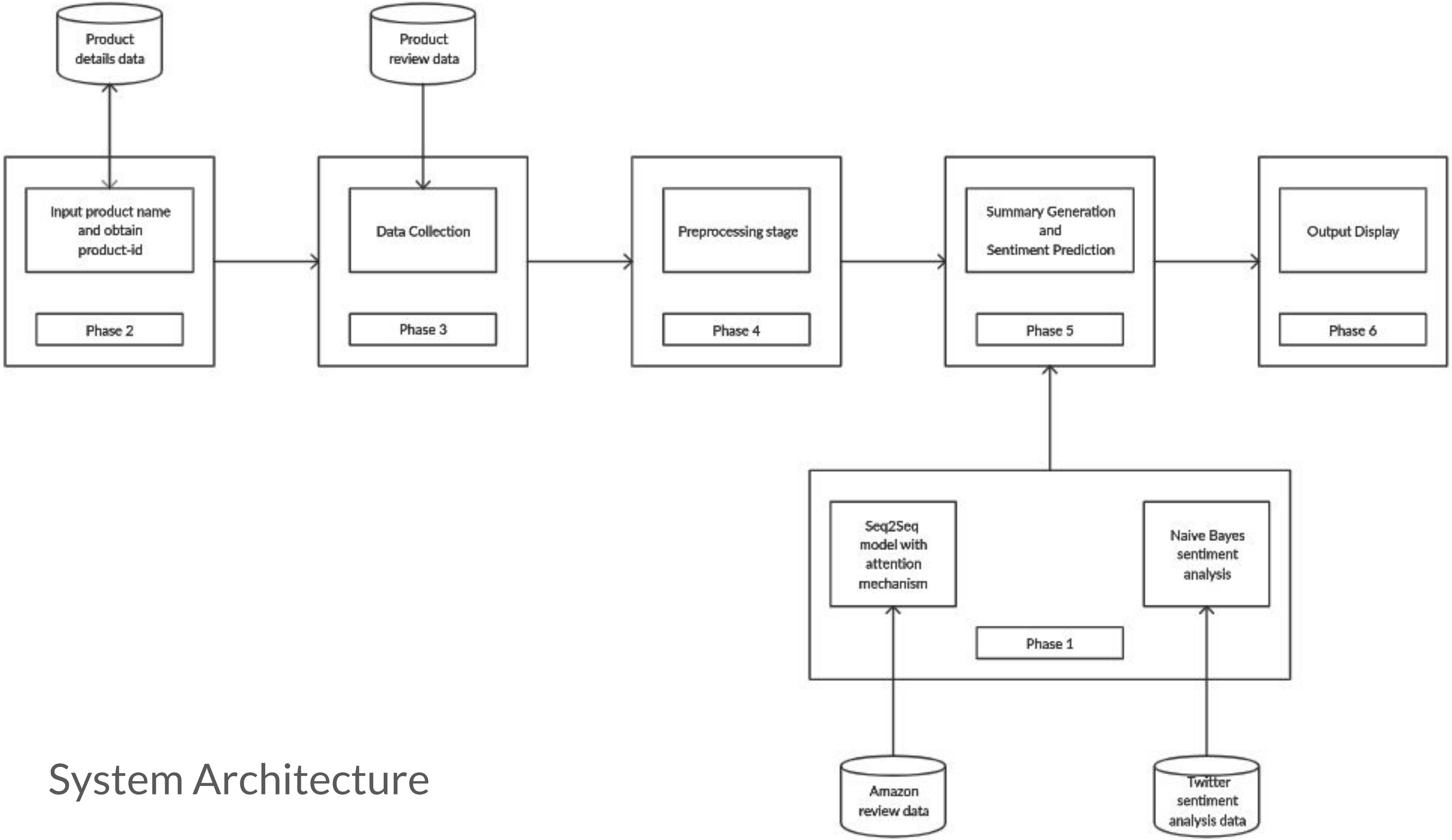
- The abstraction technique entails paraphrasing and shortening parts of the source document.
- The abstractive text summarization algorithms create new phrases and sentences that relay the most useful information from the original text.
- Abstractive summaries attempt to improve the coherence among sentences by eliminating redundancies and clarifying the context of sentences.

System Design

Module Division

1. Data Collection & Preprocessing
2. Model Training
3. Input Module
4. Summary Generation & Sentiment Analysis
5. Output Display

System Architecture



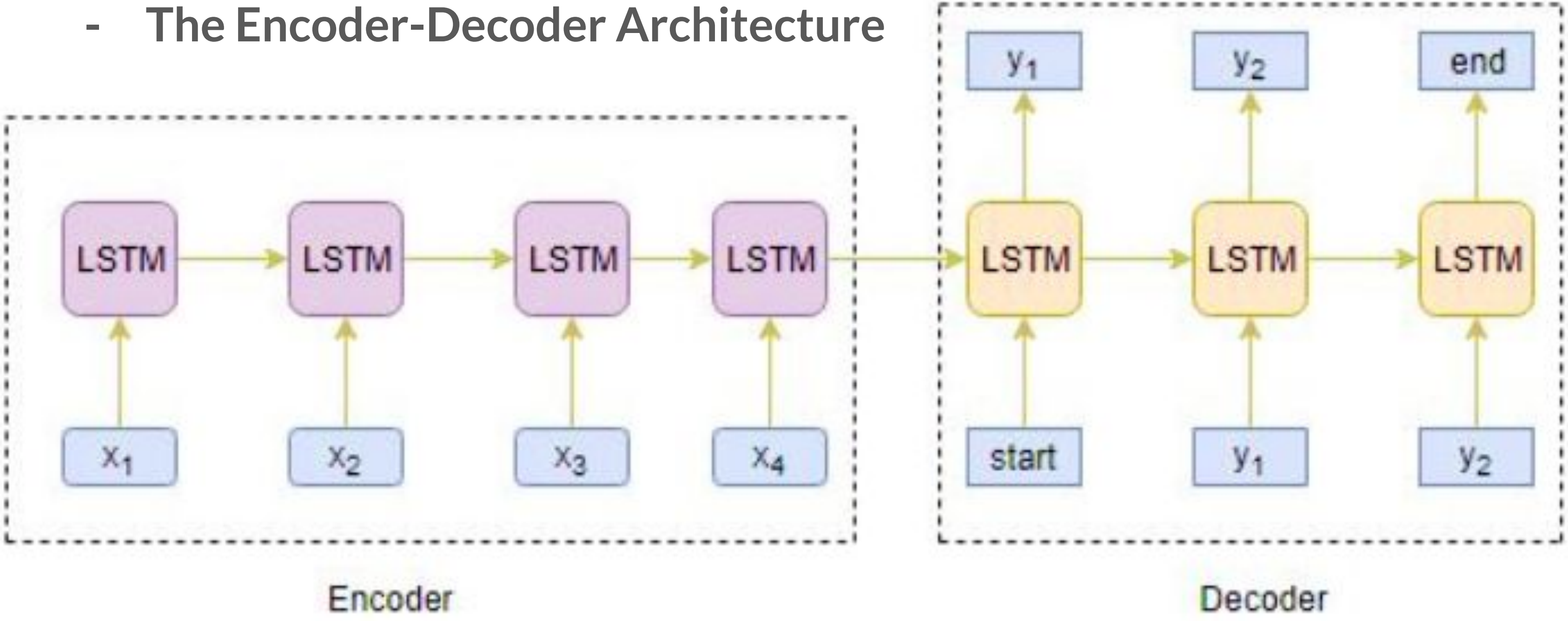
System Architecture

Sequence-to-Sequence (Seq2Seq) Modeling

- Built on problems with sequential information.
- The objective is to build a text summarizer where the input is a long sequence of words (in a text body), and the output is a short summary (which is a sequence as well).
- So, we can model this as a **Many-to-Many Seq2Seq problem**.

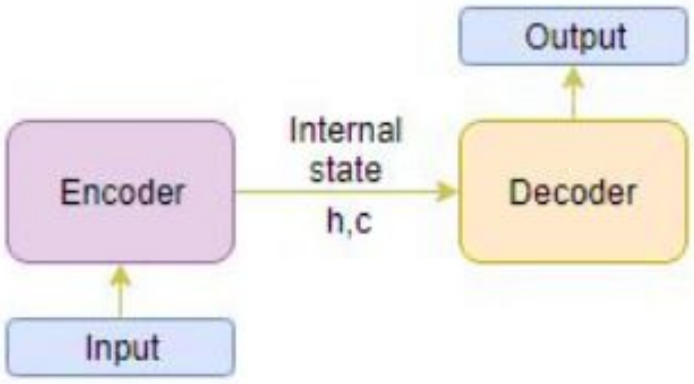
Major Components of Seq2Seq Architecture

- The Encoder-Decoder Architecture



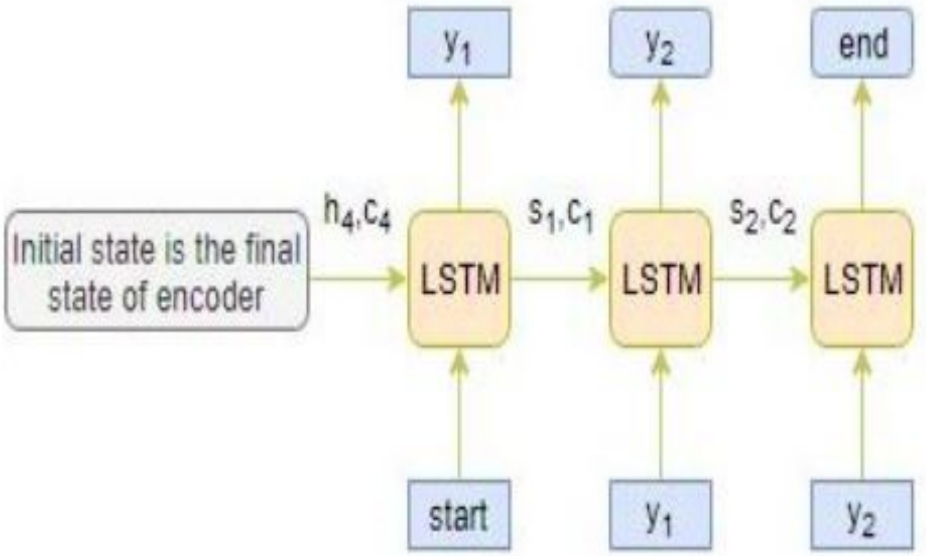
Encoder

- Encoder reads the entire input sequence, one word at a time.
- Processes and captures the contextual information present in the input sequence.
- The hidden state (hi) and cell state (ci) of the last time step are used to initialize the decoder.



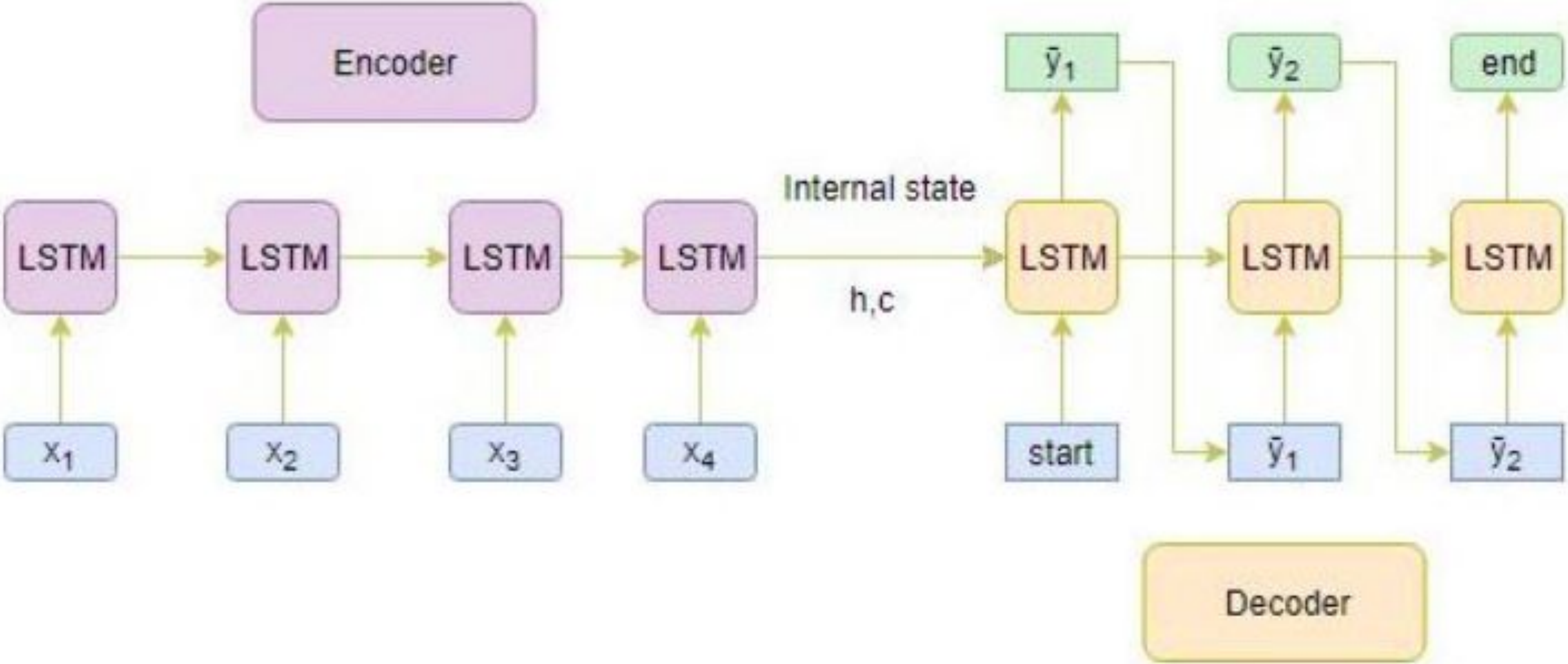
Decoder

- Decoder reads the entire target sequence word-by-word and predicts the same sequence offset by one timestep.
- Trained to predict the next word in the sequence given the previous word.
- The target sequence is unknown while decoding the test sequence.



Long Short Term Memory

- Long Short Term Memory (LSTM) has been used to implement the encoder and decoder components.
- Capable of capturing long term dependencies by overcoming the problem of vanishing gradient.
- Can selectively remember and forget data.



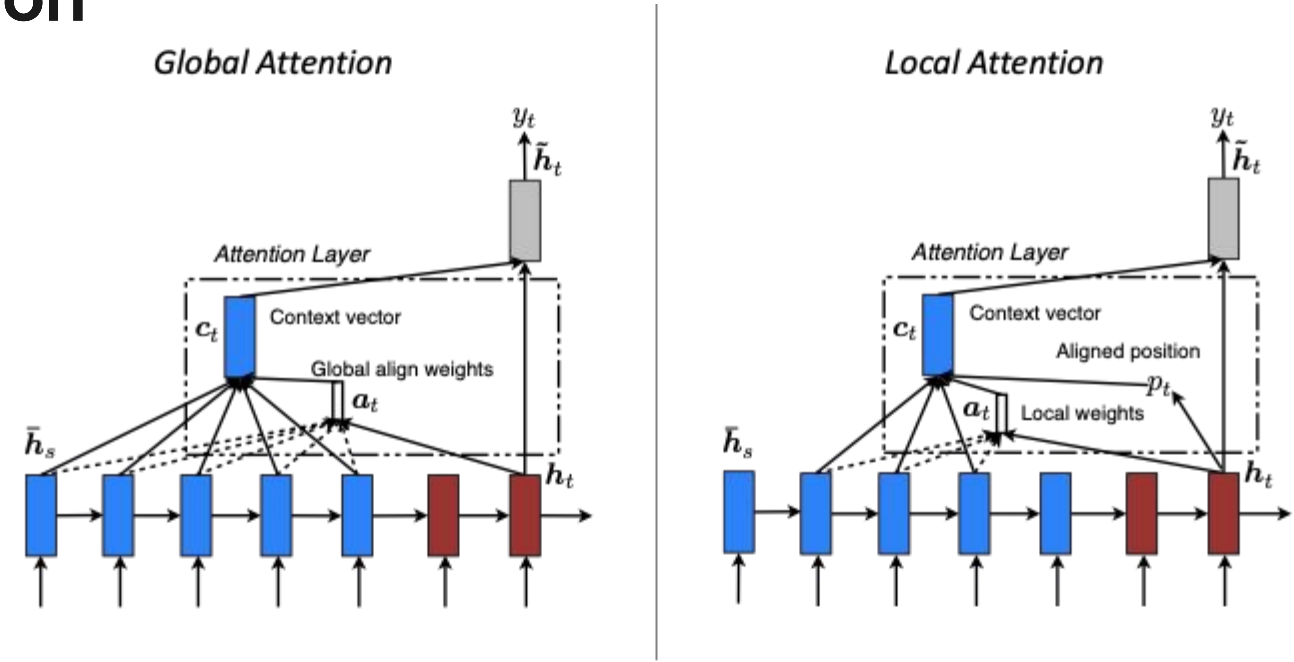
Encoder
LSTM
LSTM
LSTM
LSTM
x_1
x_2
x_3
x_4
Internal state
h,c
$\bar{y}_1$
$\bar{y}_2$
end
LSTM
LSTM
LSTM
start
$\bar{y}_1$
$\bar{y}_2$
Decoder

Attention Mechanism

- Difficult for the LSTM-based encoder to memorize long sequences into a fixed length vector. So, attention mechanism is introduced.
- Aims to predict a word by looking at a few specific parts of the sequence only, rather than the entire sequence.
- Instead of looking at all the words in the source sequence, the importance of specific parts of the source sequence that result in the target sequence can be increased.

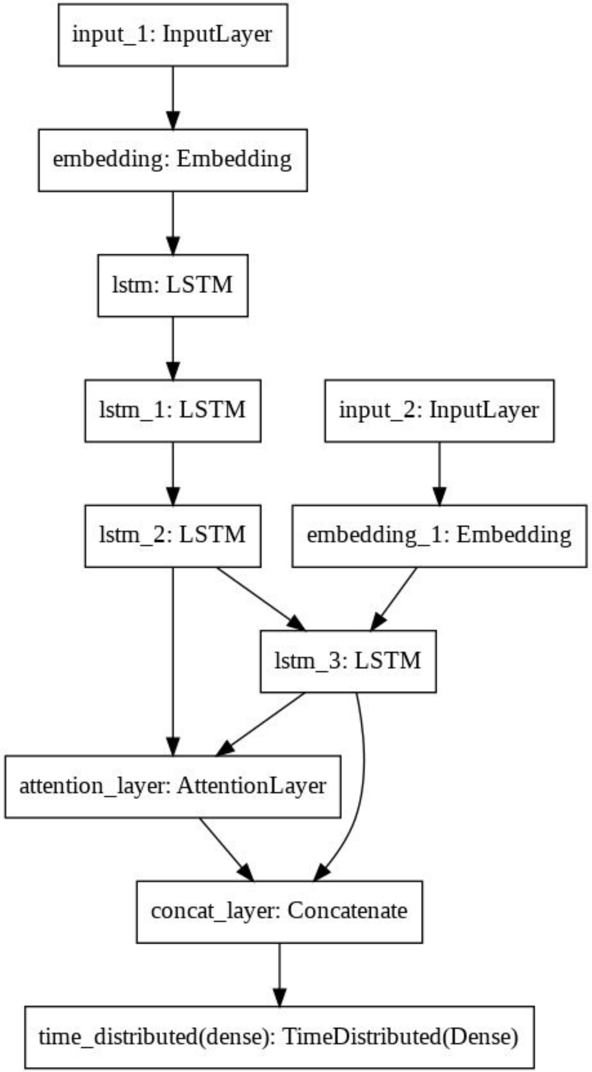
Global Attention

- The attention is placed on all the source positions.
- All the hidden states of the encoder are considered for deriving the attended context vector.



```
______________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==============================================================================================
input_1 (InputLayer)            [(None, 100)]        0
______________________________________________________________________________________________
embedding (Embedding)           (None, 100, 100)     3159500     input_1[0][0]
______________________________________________________________________________________________
lstm (LSTM)                     [(None, 100, 300), ( 481200      embedding[0][0]
______________________________________________________________________________________________
input_2 (InputLayer)            [(None, None)]       0
______________________________________________________________________________________________
lstm_1 (LSTM)                   [(None, 100, 300), ( 721200      lstm[0][0]
______________________________________________________________________________________________
embedding_1 (Embedding)         (None, None, 100)    750900      input_2[0][0]
______________________________________________________________________________________________
lstm_2 (LSTM)                   [(None, 100, 300), ( 721200      lstm_1[0][0]
______________________________________________________________________________________________
lstm_3 (LSTM)                   [(None, None, 300),  481200      embedding_1[0][0]
                                                                 lstm_2[0][1]
                                                                 lstm_2[0][2]
______________________________________________________________________________________________
attention_layer (AttentionLayer ((None, None, 300),  180300      lstm_2[0][0]
                                                                 lstm_3[0][0]
______________________________________________________________________________________________
concat_layer (Concatenate)      (None, None, 600)    0           lstm_3[0][0]
                                                                 attention_layer[0][0]
______________________________________________________________________________________________
time_distributed (TimeDistribut (None, None, 7509)   4512909     concat_layer[0][0]
==============================================================================================
Total params: 11,008,409
Trainable params: 11,008,409
Non-trainable params: 0
______________________________________________________________________________________________
```

Model Summary


input_1: InputLayer
embedding: Embedding
lstm: LSTM
lstm_1: LSTM
input_2: InputLayer
lstm_2: LSTM
embedding_1: Embedding
lstm_3: LSTM
attention_layer: AttentionLayer
concat_layer: Concatenate
time_distributed(dense): TimeDistributed(Dense)

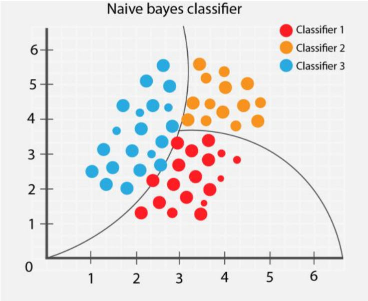
Naive Bayes Model for Sentiment Analysis

- Sentiment analysis - A classification problem.
- Often, better than complex algorithms.
- Twitter sentiment dataset available in NLTK is used to train the Naive Bayes model.
- The model proved to have an accuracy of more than 95% while testing.

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

using Bayesian probability terminology, the above equation can be written as

$$\text{Posterior} = \frac{\text{prior x likelihood}}{\text{evidence}}$$

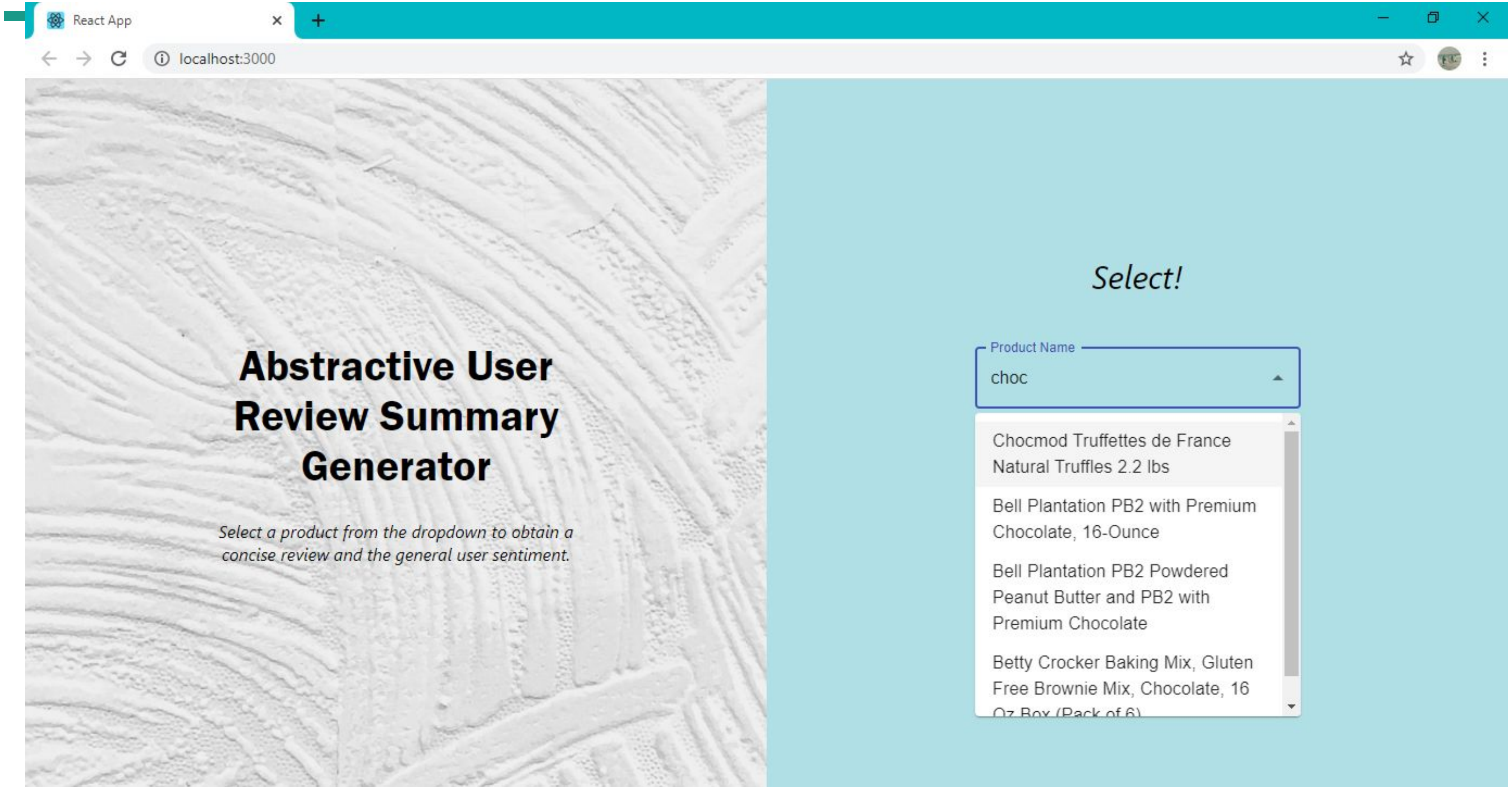


Preparing the Training Data

- The dataset contained text which had sms language and short forms in different slangs which was mapped using a dictionary of abbreviations to make the text more understandable.
- The input text was then cleaned to remove noise by following the preprocessing steps which includes POS tagging, lemmatization, removal of stop words, removal of punctuations.
- The cleaned data was used to train the Naive Bayes model to classify the sentiment of any given text.
- For the seq2seq model, the review text and the expected summary was cleaned as described and used for training.

Testing

Unit Testing : **Input Module**



Unit Testing : **Preprocessing Module**

```
%%time
import pandas as pd
product_id = "B002QWP89S" #User input

df = pd.read_csv(r"/content/gdrive/My Drive/PROJECT | S7-S8/Data/Reviews.csv", nrows=100000)
df.groupby('ProductId').mean()
df_test = (df.loc[df['ProductId'] == product_id])
print(df_test.columns)

pickle_in = open(r"/content/gdrive/My Drive/PROJECT | S7-S8/Pickle Files/nb_sentiment_analysis_final.pkl","rb")
classifier = pickle.load(pickle_in)
df_test['PreprocessedText'] = df_test.Text.apply(lambda x: remove_noise(word_tokenize(str(x))))
df_test['Sentiment'] = df_test.PreprocessedText.apply(lambda x: classifier.classify(dict([tok, True] for tok in x)))
print(df_test[['PreprocessedText', "Sentiment"]])
```

```
                                                                                                                      PreprocessedText
 from, these, use, with, caution., bathroom, if, the, cat, get, too, many, she, have, the, run, ..., bathroom, sheltie, do, good, when, i, up, her, ...]
22.37, for, this, 96-pack, and, it, be, by, far, the, best, price, i, have, find, anywhere, i, wish, these, be, part, of, the, subscribe, save, program]
;gies, do, n't, i, buy, this, for, my, dashchund, and, minpin, and, it, 's, perfect, a, great, price, for, a, great, product, who, could, ask, for, more]
                   [what, can, i, say, dog, love, greenies, they, begg, for, them, all, the, time, they, always, sit, by, the, cupboard, and, ask, for, more]
 lite, for, my, dog, the, package, come, quickly, and, be, package, appropriately, i, be, very, satisfied, with, this, purchase, and, with, the, seller]
                                                                                                                                                ...
 and, they, do, whiten, the, teeth, very, well, see, all, the, great, review, help, your, dog, teeth, if, you, do, n't, want, to, brush, them, like, me]
:smart, they, be, 32.99, plus, tax, for, the, exact, same, amount, thanks, for, the, good, buy., bathroom, bathroom, sincerely, bathroom, danny, knowles]
 where, i, get, my, puppy, and, he, love, them, i, like, the, fact, that, they, help, to, clean, his, teeth, as, well, as, satisfy, his, need, to, chew]
:, fast, and, convenient., bathroom, i, be, a, huge, fan, of, amazon, they, treat, their, customer, right, and, make, all, purchases/returns, very, easy]
 you, buy, for, a, lot, more, at, other, store, not, much, more, to, say, about, i, love, it, she, love, it, and, i, 've, buy, it, again, numerous, time]
```

Unit Testing : **Summary Model**

```
[ ] %%time
    #compile and train the model
    filename = 'model.h5'
    model.compile(optimizer='rmsprop', loss='sparse_categorical_crossentropy')
    es = EarlyStopping(monitor='val_loss', mode='min', verbose=1,patience=2)
    history=model.fit([x_tr,y_tr[:,:-1]], y_tr.reshape(y_tr.shape[0],y_tr.shape[1], 1)[:,1:] ,epochs=5, callbacks=[es], batch_size=624, v
```

```
Epoch 1/5
529/529 [==============================] - 987s 2s/step - loss: 1.8278 - val_loss: 1.6072
Epoch 2/5
529/529 [==============================] - 990s 2s/step - loss: 1.5187 - val_loss: 1.4428
Epoch 3/5
529/529 [==============================] - 998s 2s/step - loss: 1.4035 - val_loss: 1.3665
Epoch 4/5
529/529 [==============================] - 992s 2s/step - loss: 1.3384 - val_loss: 1.3239
Epoch 5/5
529/529 [==============================] - 999s 2s/step - loss: 1.2954 - val_loss: 1.2948
CPU times: user 1h 53min 18s, sys: 19min 19s, total: 2h 12min 37s
Wall time: 1h 23min 7s
```

```
[ ] #Testing
    for i in range(0,100):
        print("Review:",seq2text(x_tr[i]))
        print("Original summary:",seq2summary(y_tr[i]))
        print("Predicted summary:",decode_sequence(x_tr[i].reshape(1,max_text_len)))
        print("\n")
```

```
Review: like coffee best many tried amazon normally buy costco found price ordered delivered door amazon nice full bodied flavor
Original summary: good coffee
Predicted summary:  best coffee ever


Review: received march pictured got sweet pops wanted kind would ordered box cherry stawberry orange grape blue razz berry assortment flavors huh disappointed
Original summary: ordered sweet and sour pops got sweet pops
Predicted summary:  not what ordered


Review: really like indian food find kitchens india dishes fairly authentic tasting meals variety pack good range fairly mild spicy provides something everyone
Original summary: great indian sampler set highly recommended
Predicted summary:  very good indian food highly recommended


Review: ever since tried coffee absolutely loved drink cups coffee per day picky coffee drink anything green moutain columbian get costco went away weekend actu
Original summary: dont drink anything else
Predicted summary:  best coffee ever


Review: breakfast cookies favorite young children friend one ounce cookies right size young children individually wrapped used home take along snack item also u
Original summary: good snack for children
Predicted summary:  great for kids
```

Unit Testing : **Sentiment Analysis Model**

[6]

```
positive_tokens_for_model = get_tweets_for_model(positive_cleaned_tokens_list)
negative_tokens_for_model = get_tweets_for_model(negative_cleaned_tokens_list)

positive_dataset = [(tweet_dict, "Positive") for tweet_dict in positive_tokens_for_model]

negative_dataset = [(tweet_dict, "Negative") for tweet_dict in negative_tokens_for_model]

dataset = positive_dataset + negative_dataset

random.shuffle(dataset)

train_data = dataset[:7000]
test_data = dataset[7000:]

classifier = NaiveBayesClassifier.train(train_data)
print("Score Naive Bayes:", classify.accuracy(classifier, test_data))

pickle.dump(classifier, open("nb.pkl", 'wb'))
files.download('nb.pkl')
```

```
[(':)', 3691), (':-)', 701), (':d', 658), ('thanks', 391), ('follow', 357), ('love', 337), ('...', 290), ('good', 286), ('get', 263), ('thank', 253)]
Score Naive Bayes: 0.996
CPU times: user 13.1 s, sys: 232 ms, total: 13.4 s
Wall time: 17.2 s
```

Unit Testing : **Summary Generation**

	pid	title	summaries	avg_rating	sentiment	price	image
0	B001L4JH5I	Pamela's Products Gluten-free Bread Mix, 4-Pound Bags (Pack of 3)	good but not great. great tasting and low carb. great tasting snack. great tasting and healthy. great tasting. great for cats. great for gluten free. great pizza crust. great product. great snack.	4.618497	1	$37.71	https://images-na.ssl-images-amazon.com/images/I/71crjrvBCFL._SL1500_.jpg
1	B000DZFMEQ	Pamela's Products Gluten Free, Bread Mix, 19-Ounce Packages (Pack of 6)	good but not great. great tasting and easy to make. great taste and texture. great tasting and healthy. great for cats. great gluten free cake. great for my cats. great cat food. great taste. gre...	4.627660	1	$28.68	https://images-na.ssl-images-amazon.com/images/I/71aqMwcuqEL._SL1500_.jpg
2	B003QNJYXM	5 Hour Energy Extra Strength Energy Shots, Berry, 12 pk	good but not great. great tasting and easy to make. great tasting and healthy. great tasting snack. great taste and texture. great taste but not so much. great for kids with allergies. great for k...	4.229167	1	$27.89	https://images-na.ssl-images-amazon.com/images/I/71u0bax1-IL._SL1300_.jpg
3	B0039ZOZ86	Gourmet Basics Smart Fries Original KC BBQ, 3-Ounce Bags (Pack of 12)	good but not great. great taste. great for kids and adults. great flavor. great product. great for the price. great taste but not. great gum. great	4.141892	1	$32.91	https://images-na.ssl-images-amazon.com/images/I/81fXAtPvYHL._SL1500_.jpg

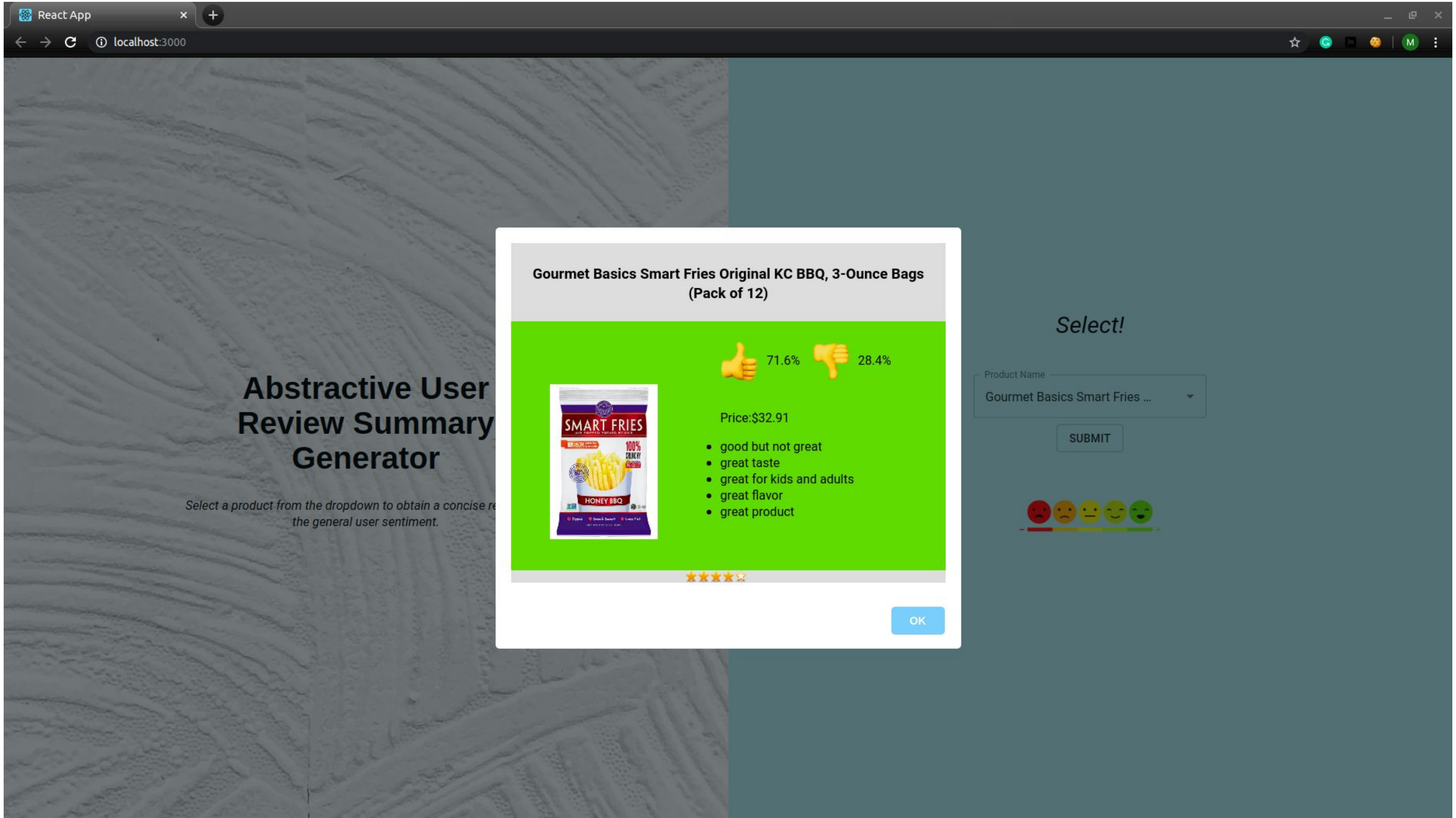
Integration Testing

Select C:\Windows\System32\cmd.exe - flask run

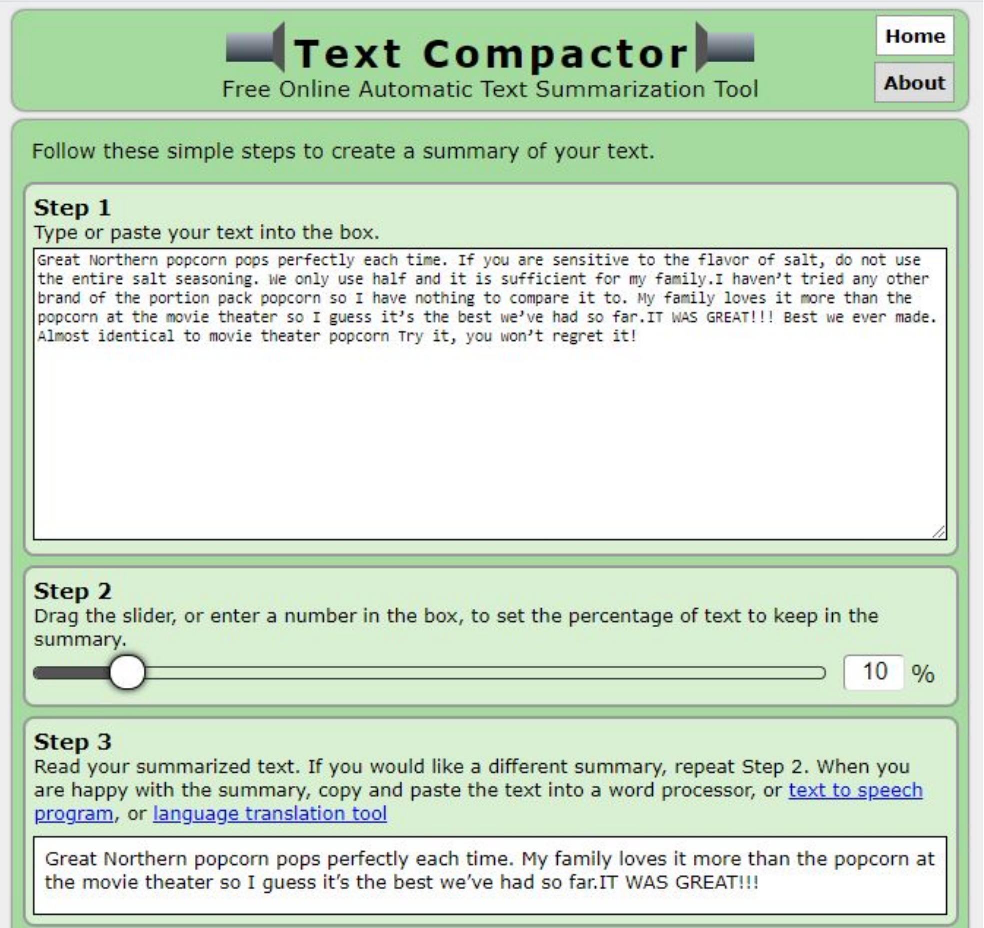
```
33  B004SRH2B6  ...  https://images-na.ssl-images-amazon.com/images...
34  B005GX00BK  ...  https://images-na.ssl-images-amazon.com/images...
35  B007TJGZ5E  ...  https://images-na.ssl-images-amazon.com/images...
36  B008ZRKZSM  ...  https://images-na.ssl-images-amazon.com/images...
37  B000CQG8KS  ...  https://images-na.ssl-images-amazon.com/images...
38  B000CQC05K  ...  https://images-na.ssl-images-amazon.com/images...
39  B0058AMYTC  ...  https://images-na.ssl-images-amazon.com/images...
40  B005HG9ESG  ...  https://images-na.ssl-images-amazon.com/images...
41  B000GAT6NG  ...  https://images-na.ssl-images-amazon.com/images...
42  B000CQIDHE  ...  https://images-na.ssl-images-amazon.com/images...
43  B000WB1YSE  ...  https://images-na.ssl-images-amazon.com/images...
44  B0012BUR8Q  ...  https://images-na.ssl-images-amazon.com/images...
45  B0098WV8F2  ...  https://images-na.ssl-images-amazon.com/images...
46  B002AQP5MK  ...  https://images-na.ssl-images-amazon.com/images...
47  B005GTWCTM  ...  https://images-na.ssl-images-amazon.com/images...
48  B001LG940E  ...  https://images-na.ssl-images-amazon.com/images...
49  B0033HPPIO  ...  https://images-na.ssl-images-amazon.com/images...

[50 rows x 4 columns]
Generate sentiment
{"pid":{"0":"B000DZFMEQ"},"title":{"0":"Pamela's Products Gluten Free, Bread Mix, 19-Ounce Packages (Pack of 6)"},"summaries":{"0":[" good but not great"," great tastin
g and easy to make"," great taste and texture"," great tasting and healthy"," great for cats"," great gluten free cake"," great for my cats"," great cat food"," great t
aste"," great tasting",""]},"avg_rating":{"0":4.6276595745},"sentiment":{"0":1},"price":{"0":"$28.68"},"image":{"0":"https:\/\/images-na.ssl-images-amazon.com\/images\/
I\/71aqMwcuqEL._SL1500_.jpg"}}
127.0.0.1 - - [23/May/2020 12:42:51] "[37mPOST /generatesummary HTTP/1.1[0m" 200 -
```

System Testing



Validation Testing



Conclusion

- A user review consolidation system has been developed using the method of abstractive summarizing to obtain a concise form of the lengthy product reviews available on e-commerce websites along with additional features such as overall rating and sentiment analysis.
- The system uses a dataset of fine food reviews from Amazon to generate the summaries. It also generates the general sentiment of the customers towards each product.
- The average rating based on the numerous reviews available for each product is also calculated.
- The sentiment analysis has been performed by training a model based on supervised learning. The Naive Bayes classifier has been chosen as the machine learning model, trained on twitter sentiment analysis data, which provides an accuracy of more than 95%.
- The abstractive summarizer has been modelled as a Sequence-to-Sequence model which uses attention mechanism to improve accuracy.
- The model gives fairly good summaries of the reviews.

Future Scope

- The model is trained using reviews for Fine Foods.
- It can be extended to include other categories of items available online, thereby generating better summaries due to a larger dataset. It can also be improved to integrate reviews from various online shopping websites to provide an unbiased summary and rating.
- The generalization capability of a deep learning model enhances with an increase in the training dataset size.
- Web scraping can be introduced to perform summarization on real-time data.
- Implementing Bi-Directional LSTM which is capable of capturing the context from both the directions could result in a better context vector.
- The beam search strategy can be applied for decoding the test sequence instead of using the greedy approach (argmax).
- Pointer-generator networks and coverage mechanisms can be implemented in the model to further improve the summary generation capability of the model.

Thank You