ML Internship Project Report

Tweet Sentiment Extraction

Logine Magdi
7163

Muhammad Kotb
19016258

Heba-tullah Mostafa
19016836

September 14, 2023

Contents

1 Problem Overview

1.1 Problem Statement

With all of the tweets circulating every second it is hard to tell whether the sentiment behind a specific tweet will impact a company, or a person's, brand for being viral (positive), or devastate profit because it strikes a negative tone. Capturing sentiment in language is important in these times where decisions and reactions are created and updated in seconds. But, which words actually lead to the sentiment description ?

1.2 Data Description

The dataset used here is from the Kaggle competition Tweet Sentiment Extraction and Figure Eight's Data for Everyone platform . It consists of two data files train.csv and test.csv, where there are 27481 rows in training data and 3534 rows in test data . Columns » textID - unique ID for each piece of text text - the text of the tweet sentiment - the general sentiment of the tweet selected text - [train only] the text that supports the tweet's sentiment .

1.3 Technical Approach

Our base model was a RNN model that take the sentence and its sentiment and output word or phrase best supports that sentiment .Firstly we would prepare our dataset using some preprocessing operations and pass it to the model . The first layer in our model is the embedding layer, after that Bi-LSTM cells receive representations from the embedding layer. Long-range dependencies for sentiment analysis are modeled through Bi-LSTM. It adds input, output and forget gates to a recurrent cell thereby adding recurrent connections to the network to include information about the sequence of words in the data. Finally, we predict the output as an array includes the start and end indices of the selected text .

1.4 Evaluation Metric

The metric in this competition is the word-level Jaccard score. Current highest performing model has achieved a score of 0.73566.

2

Week One Progress

2.1 Data Analysis

Our training dataset consists of 27480 entry , each entry contains 4 columns text id, text, selected text, sentiment .

- (text) column » it contains the original text that we want to select from the text that supports the tweet's sentiment .
- (selected text) column » it contains the ground truth selected text that supports the tweet's sentiment (we would convert it later to array of start and end indices of the selected text in the text column) .
- (sentiment) column » it consists of only three values " positive , negative and neutral " .

Both text and sentiment are passed as inputs to the model and the selected text are used for evaluating the model " as ground truth prediction " .
in the next figure 1 illustration of the composition of the training dataset .

	textID	text	selected_text	sentiment
0	cb774db0d1	I`d have responded, if I were going	I`d have responded, if I were going	neutral
1	549e992a42	Sooo SAD I will miss you here in San Diego!!!	Sooo SAD	negative
2	088c60f138	my boss is bullying me...	bullying me	negative
3	9642c003ef	what interview! leave me alone	leave me alone	negative
4	358bd9e861	Sons of ****, why couldn`t they put them on t...	Sons of ****,	negative
...	...	...	...	...
27476	4eac33d1c0	wish we could come see u on Denver husband l...	d lost	negative
27477	4f4c4fc327	I`ve wondered about rake to. The client has ...	, don`t force	negative
27478	f67aae2310	Yay good for both of you. Enjoy the break - y...	Yay good for both of you.	positive
27479	ed167662a5	But it was worth it ****.	But it was worth it ****.	positive
27480	6f7127d9d7	All this flirting going on - The ATG smiles...	All this flirting going on - The ATG smiles. Y...	neutral

Figure 1: Some Data Samples .

Our data contains about :

- 8582 entry of positive sentiment samples .
- 7781 entry of negative sentiment samples .
- 11117 entry of neutral sentiment samples .

By analyzing the positive and negative samples we found that there is some keywords that appear frequently in text sentences of positive and negative sentiments like love, thank , good , happy , etc for positive sentences and insult, hate , sad , bad , etc for negative .
the next figures show the most frequent words in positive and negative sentences .

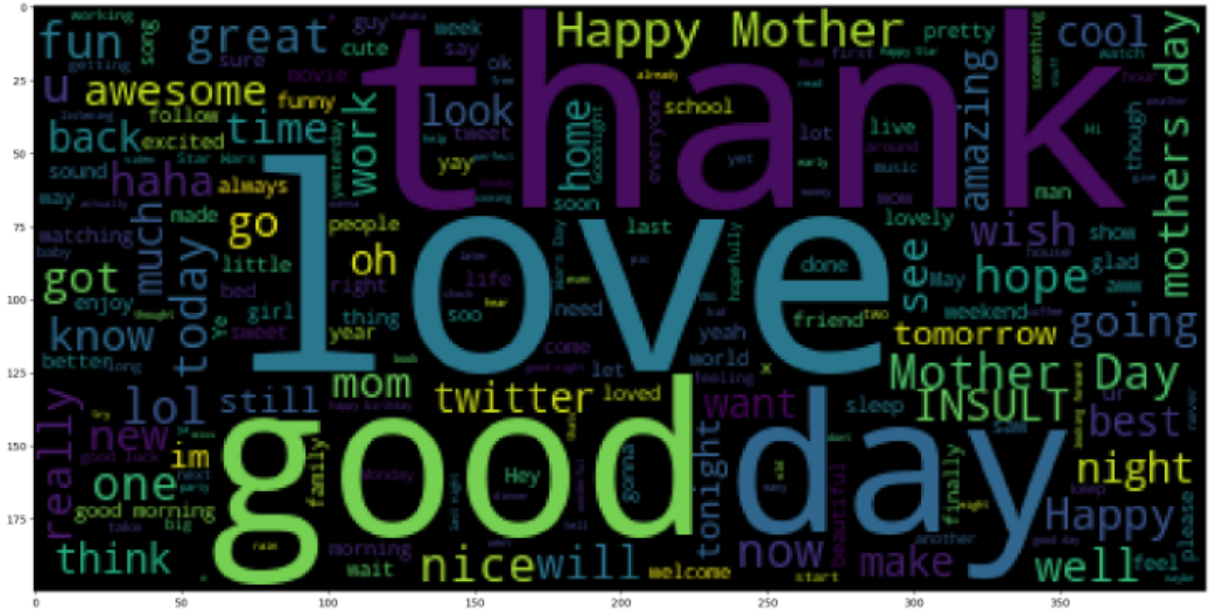


Figure 2: Most frequent words in positive sentences .

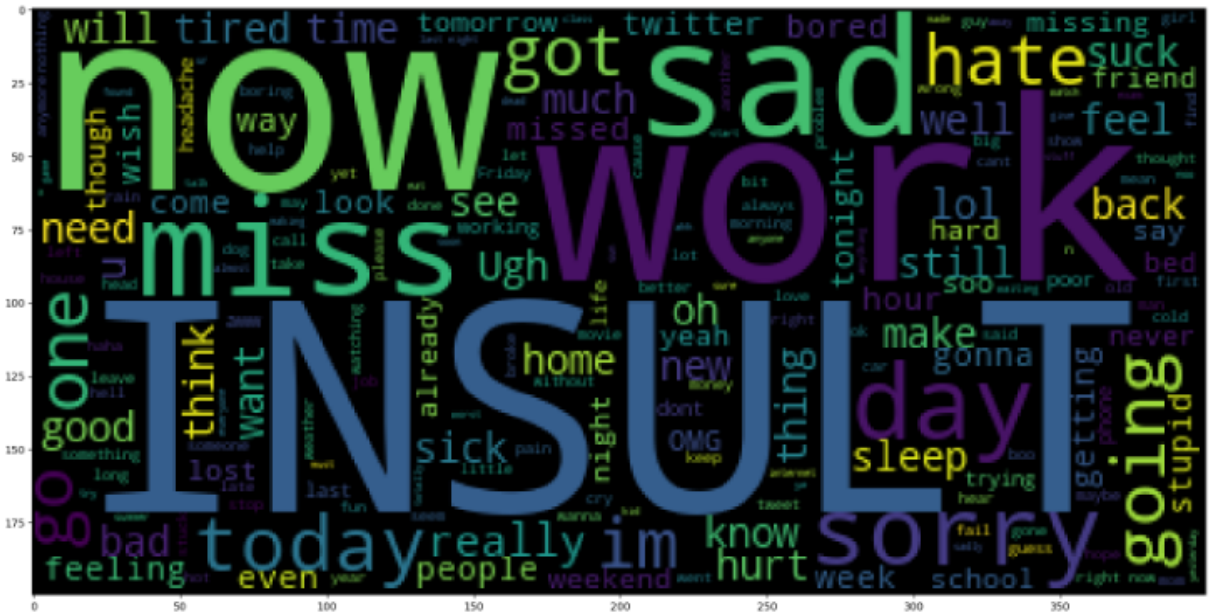


Figure 3: Most frequent words in negative sentences .

2.2 Data Preprocessing

Text pre-processing is a crucial step in natural language processing (NLP) tasks as it helps to clean and prepare raw text data for further analysis and modeling.

Text pre-processing steps are:

- **Dropping nulls**

  Null values or missing data can create inconsistencies and affect the quality of the analysis. By dropping nulls, we ensure that we are working with complete and reliable data.

- **Dropping long sentences**

  Removing non frequent long sentences from tweet dataset is advantageous to reduce padding. Dataset becomes more balanced, potentially improving the stability of the training process. In figure 4 we saw length distribution for training sentences and according to it we removed sentences with a large number of tokens and Figure 5 shows the amount of dropped data .

- **Setting up start and end indices**

  The start index is set where the first token in selected text appears in text (index in text not selected text). End index is set where the last token in selected text appears in text.

- **Expanding contractions and abbreviations**

  Contractions are shortened forms of words, such as "can't" for "cannot" or "I'm" for "I am." Expanding contractions helps in standardizing the text and ensures consistent representation of words, which is important for sentiment analysis task.

- **Expanding abbreviations**

  Abbreviations can be ambiguous and may have multiple meanings. Expanding abbreviations into their full forms, such as "OMG" for "Oh My God" or "BTW" for "By The Way", ensures that the text is correctly interpreted during analysis or model training.

- **Removing hyperlinks**

  Text data contain hyperlinks and URLs that are not relevant to the modeling task,Thus we cleaned the text and focused on the actual content.

- **Spacing punctuation**

  Properly spacing punctuation helps in tokenization, which involves placing spaces before and after punctuation and breaking down the text into individual words or tokens.[Eve]

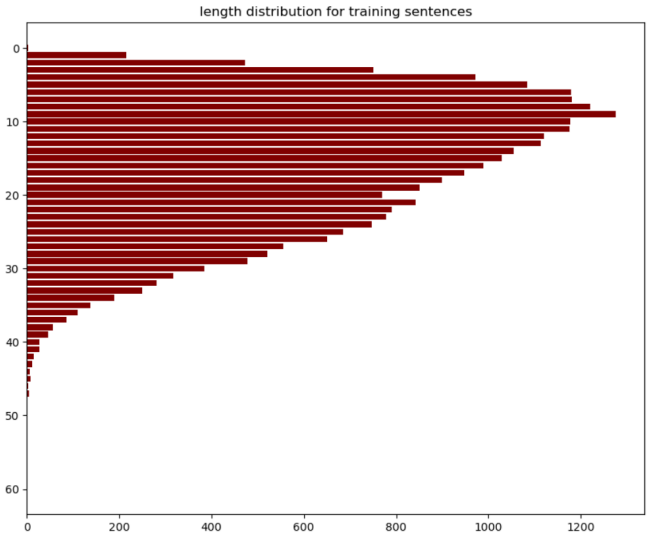


Figure 4: Length distribution for training sentences where y-axis is the number of tokens in each sentence,x-axis represents their frequency .

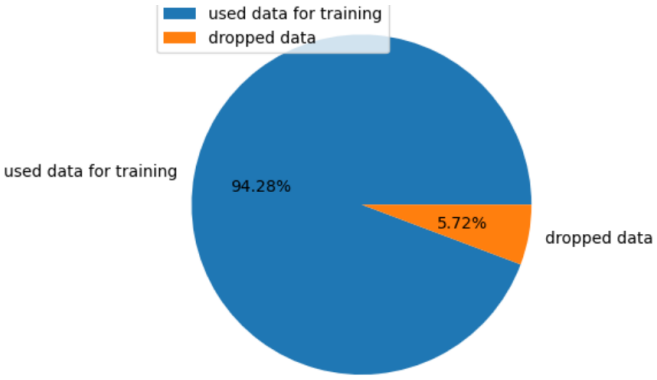


Figure 5: Data after dropping long sentences

2.3 Preparing Embedding Matrix

We perform the following key tasks for text preprocessing and embedding matrix creation:

- Tokenization: Two Tokenizer objects are initialized to tokenize text data from the 'text' and 'sentiment' columns. Vocabulary sizes are computed for both tokenizers.
- Text Sequencing: The 'text' data is tokenized and sequences are generated, ensuring a consistent sequence length of 100.
- Embedding Index: An embedding index is created by parsing pre-trained word embeddings from the 'glove.6B.100d.txt' file. This index associates words with their respective embedding vectors.
- Embedding Matrices: Two embedding matrices are constructed, one for 'text' and another for 'sentiment' data. These matrices contain word embeddings for words found in the dataset, with missing words initialized as zero vectors.

3 Week Two Progress

3.1 Validation and testing datasets preparation

training data set was split into 75% training , 5% validation and 20% testing .

- **training data (data and labels)** : for training the model .
- **validation data (data and labels)** : for tuning the model with the results of metrics (accuracy, loss etc).
- **testing data (data and labels)** : for analysis success and failure cases and test the model performance on the new data (it is different from test data provided by kaggle site) .

training dataset before splitting was (27480 x 4) 27480 rows and 4 columns after splitting it became

- **training data** : 19690 x 3 " 3 columns are for text, sentiment and the ground truth selected text " .
- **validation data** : 1037 x 3 " 3 columns are for text, sentiment and the ground truth selected text " .
- **testing data** : 5182 x 3 " 3 columns are for text, sentiment and the ground truth selected text " .

note » all the preprocessing operations were applied for training, validation and testing data .

3.2 Base Model

3.2.1 Model Architecture

Text and its sentiment are passed to our model after padding as lists with shapes (19690 x 30) and (19690 x 1) respectively .

- text is passed to embedding layer to convert each word in the text to its corresponding representation in Glove100 dictionary to allow the network to learn more about the relationship between inputs and to process the data more efficiently . (Each word is represented as vector of 100 instance .)
- sentiment also is passed to embedding layer to convert it to its corresponding representation .
- The embedded text is passed to a GRU network to recognize data's sequential characteristics and use patterns to predict the next likely scenario .
- The output of the GRU network and the embedded sentiment are flattened and concatenated in one tensor .
- The concatenated tensor is passed to a dense layer with 16 unit and ReLU activation followed by a dropout layer with rate 0.5 .
- The output of the dropout layer passed to a normalization layer to speed up and stabilize the learning process .

- A dense layer is used with 8 units and ReLU activation followed by another dense layer with only 2 units which is the prediction of our model , which represents the start and end indices of the selected text that supports the sentiment of each text .

the next figure 6 explains the model architecture and how the input is processed in our model .

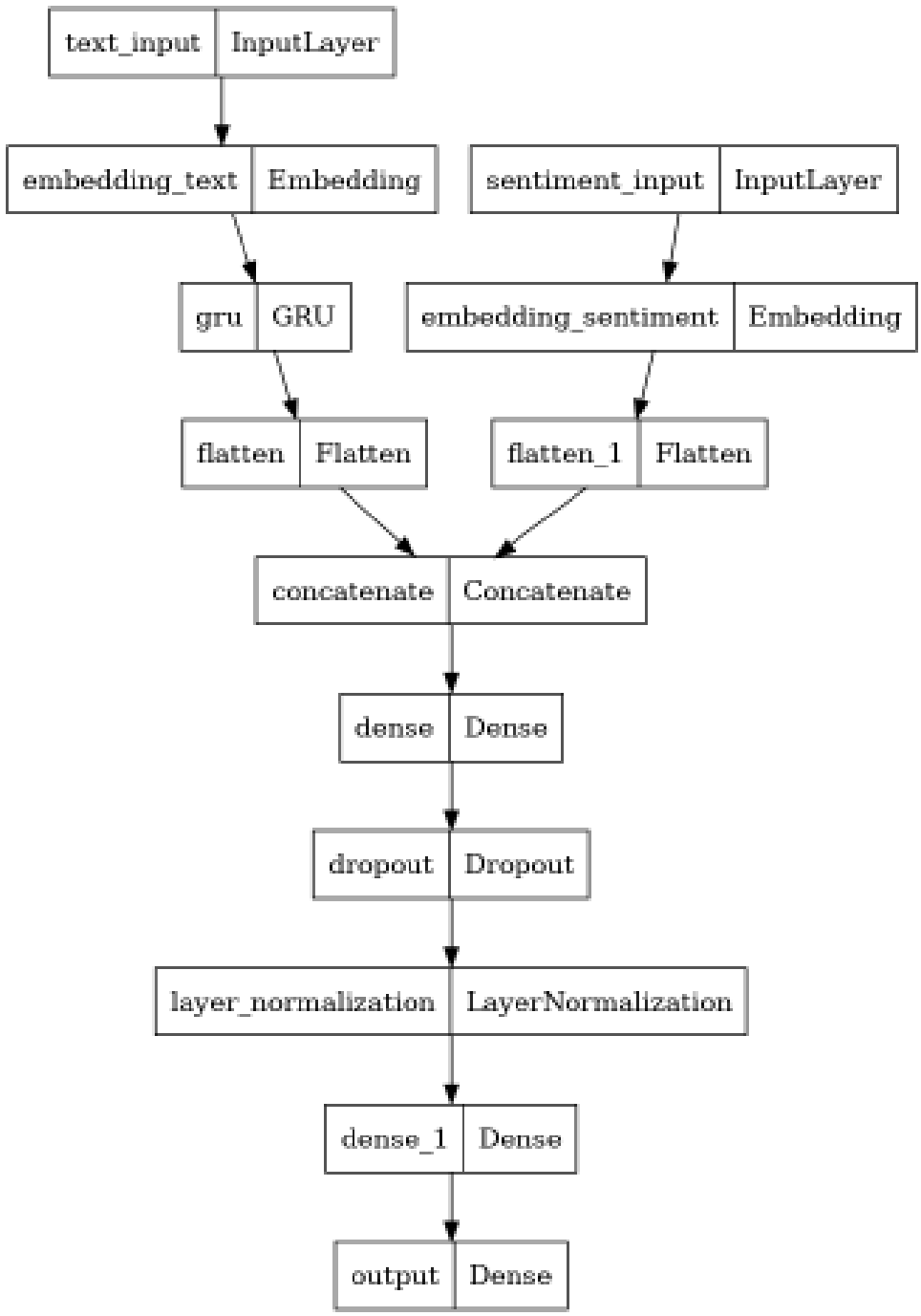


Figure 6: Model Architecture.

the next figure 7 explains the model architecture in details with the number of parameters in each layer .

```
Model: "model"
_________________________________________________________________________________________________
 Layer (type)                Output Shape              Param #     Connected to
=================================================================================================
 text_input (InputLayer)     [(None, 30)]              0           []

 embedding_text (Embedding)  (None, 30, 100)           1484600     ['text_input[0][0]']

 sentiment_input (InputLayer)  [(None, 1)]             0           []

 bidirectional (Bidirectional)  (None, 30, 64)         34048       ['embedding_text[0][0]']

 embedding_sentiment (Embedding  (None, 1, 100)        500         ['sentiment_input[0][0]']
 )

 flatten (Flatten)           (None, 1920)              0           ['bidirectional[0][0]']

 flatten_1 (Flatten)         (None, 100)               0           ['embedding_sentiment[0][0]']

 concatenate (Concatenate)   (None, 2020)              0           ['flatten[0][0]',
                                                                    'flatten_1[0][0]']

 dense (Dense)               (None, 16)                32336       ['concatenate[0][0]']

 dropout (Dropout)           (None, 16)                0           ['dense[0][0]']

 layer_normalization (LayerNorm  (None, 16)            32          ['dropout[0][0]']
 alization)

 dense_1 (Dense)             (None, 8)                 136         ['layer_normalization[0][0]']

 output (Dense)              (None, 2)                 18          ['dense_1[0][0]']

=================================================================================================
Total params: 1,551,670
Trainable params: 67,070
Non-trainable params: 1,484,600
_________________________________________________________________________________________________
```

Figure 7: Model Details.

3.3 Base Model Output Analysis

3.3.1 Loss and Accuracy Analysis

After training the model with 100 epoch we get these results :

- **loss (mean square error loss "mse") :** dropped from 114.03 to 32.39 for training and from 93.09 to 38.46 for validation
- **accuracy :** started with 18.5% and finally we got 81.4% for training and started with 20.8% and ended up with 79.17% for validation data .

the next figures 9 show the Loss and Accuracy plots along epochs .

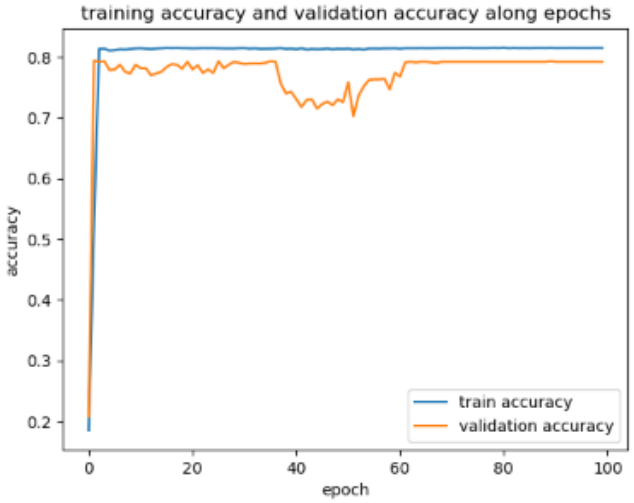


Figure 8: Accuracy plot .

Figure 9: Loss Plot .

from the graphs :

- the training accuracy graph started to be smooth after epoch 5 .
- the validation graph started to be smooth after epoch 50 .
- from the loss graph we can conclude that the model suffers from a little overfitting .

3.3.2 Failure and Success cases

After training the model the final weights were saved to be used later in prediction on the test dataset .
As mentioned before our test dataset is 5182 row after predicting the start and end indices we analyzed the output and compared it with the ground truth values , we can conclude that : there some cases the model predicts all the selected text start and end indices perfectly , some cases it predicts only part of the selected text correctly and fails to predict the other part (Intersection case) and there are some cases it fails to predict the selected text correctly .

from all the test data we got :

- **127 samples was successfully predicted (about 2.5 %) .**
- **3722 samples the model predicted part of it successfully (about 71.8 %) .**
- **1333 samples model failed to predict (about 25.7 %) .**

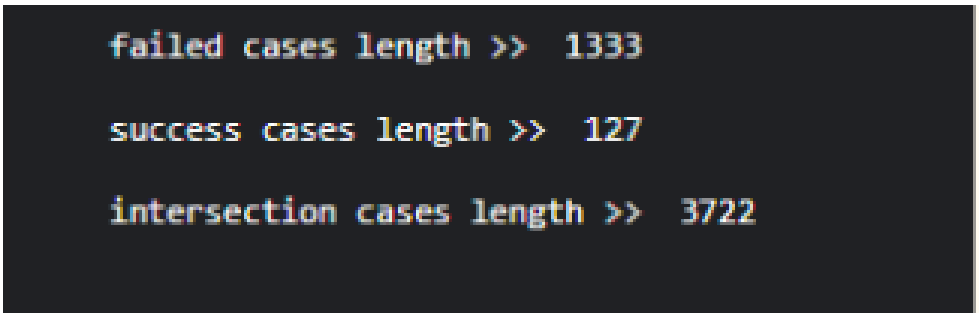


Figure 10: statistics of the model output.

this image 11 contains some success cases of the model where the predicted indices represent the start and end index of the predicted selected text by the model .

	text	sentiment	predicted indices	ground truth indices
0	work day of	neutral	[0, 2]	[0, 2]
1	think about you	neutral	[0, 2]	[0, 2]
2	hey cool :	neutral	[0, 2]	[0, 2]
3	miss you music	neutral	[0, 2]	[0, 2]
4	jackson rathbone !	neutral	[0, 2]	[0, 2]
5	grr . we do not finish until july	negative	[0, 1]	[0, 1]
6	la bind ! !	neutral	[0, 2]	[0, 2]
7	feel pretty good this morning ! let we hope ...	positive	[0, 2]	[0, 2]
8	yellow for ?	neutral	[0, 2]	[0, 2]
9	guinness at coogar	neutral	[0, 2]	[0, 2]
10	one more final	neutral	[0, 2]	[0, 2]
11	goosh ! someone pay miss you lastfm subscrip...	positive	[0, 1]	[0, 1]
12	happy birthday ! !	positive	[0, 2]	[0, 2]
13	thank you	positive	[0, 1]	[0, 1]
14	will miss jay leno . .	negative	[1, 1]	[1, 1]
15	ugh . kind of bored .	negative	[0, 1]	[0, 1]
16	insult it . you do not look well . you hav...	negative	[0, 2]	[0, 2]
17	that suck . but woot for misha .	neutral	[0, 2]	[0, 2]
18	just have kfc	neutral	[0, 2]	[0, 2]
19	oh good . I get time out be the gooseberry a...	positive	[0, 2]	[0, 2]

Figure 11: some success "correctly predicted" cases of the model.

this image 12 contains some cases where the model failed to predict correctly any part of the selected text

.

	text	sentiment	predicted indices	ground truth indices
0	heaven , not good I can empathise . finger...	positive	[12, 19]	[0, 1]
1	miss you wack friend be all raid miss you kitc...	positive	[12, 19]	[0, 1]
2	my english be break	negative	[1, 1]	[3, 3]
3	# iusedtobescaredof the girl in the year abov...	negative	[1, 4]	[0, 1]
4	this cigarette be significant other shout out ...	positive	[0, 2]	[3, 7]
5	pull out the breakfast sausage for mother day ...	positive	[4, 7]	[9, 9]
6	do crossfit run today . . . agitate miss y...	negative	[2, 5]	[7, 7]
7	do not win a lammy last night but happy for sc...	positive	[3, 7]	[8, 8]
8	wow . my teacher just call I a skunk cuz of ...	positive	[1, 5]	[0, 1]
9	I ante meridiem amplitude modulation wat...	negative	[5, 9]	[11, 18]
10	good morning by the way - a public holiday i...	positive	[7, 14]	[22, 24]
11	you be always amusing .	positive	[0, 1]	[3, 4]
12	oh no ! my fan break noo ! great now I hav...	negative	[12, 29]	[6, 7]
13	be very very tired just want time out sleep	negative	[0, 2]	[10, 11]
14	! ! ! ! glad you be alright !	positive	[0, 2]	[4, 8]
15	fudge . . . . just bs ' d that whole pape...	negative	[5, 10]	[22, 22]
16	welcome time out glasgow felix , sorry I can...	negative	[2, 5]	[5, 12]
17	want time out go out tonight but can not get home	negative	[3, 6]	[6, 10]
18	the last song all american reject	positive	[4, 9]	[16, 19]
19	oh I see who you mean now - tht heltershelte...	negative	[7, 14]	[15, 15]

Figure 12: some failure "wrongly predicted" cases of the model.

this image 13 contains some cases the model predict correctly only part of the right prediction (based on ground truth values)the predicted indices represent the start and end index of the predicted selected text by the model .

NOTE : the ground truth was provided as text in column selected text in the training data frame but to compare it with the model output we convert it to an array contains the start and end index of the selected text from the original text column

	text	sentiment	predicted indices	ground truth indices
0	insult . . . I have get significant other ...	neutral	[1, 24]	[0, 29]
1	yes I ante meridiem amplitude modulation	neutral	[0, 2]	[0, 5]
2	I feel like miss you dream just get crush	negative	[0, 2]	[0, 8]
3	want time out go home .	neutral	[0, 2]	[0, 5]
4	listen time out dashboard confessional & cou...	neutral	[2, 18]	[0, 18]
5	poor baby girl chloe be freak out because of a...	neutral	[1, 31]	[0, 13]
6	star trek wait and see grtsat bggete drunk now	positive	[0, 3]	[0, 8]
7	you well come back soon !	positive	[1, 1]	[0, 5]
8	happy mother day time out heidi klum	positive	[0, 2]	[0, 0]
9	I really wish I could make it ! a hr . dri...	positive	[3, 6]	[0, 7]
10	lalalaland . . . why ante meridiem ampl...	neutral	[1, 30]	[0, 19]
11	on miss you where be you time out school !	neutral	[1, 7]	[0, 9]
12	thank for the follow miss you new twitpeep !	negative	[0, 2]	[0, 1]
13	have break off the facebook wedding significan...	negative	[7, 14]	[6, 10]
14	_ eclectic but then you leave	neutral	[0, 2]	[0, 5]
15	have find a free wifi point . . . and info...	positive	[4, 9]	[0, 13]
16	whatever be one , but whatever be not the sa...	neutral	[2, 15]	[0, 17]
17	omg I want time out go too ! hahaha	positive	[0, 1]	[0, 8]
18	there be a guy in miss you house significant o...	neutral	[1, 24]	[0, 22]
19	conference call arrange for today just blow mi...	neutral	[2, 12]	[0, 13]

Figure 13: some cases the model predicted correctly part of the prediction .

From the following graphs we can summarize the model performance :

- The best performance of the model is on the neutral data
- Most of the Failure cases are from positive and negative data .

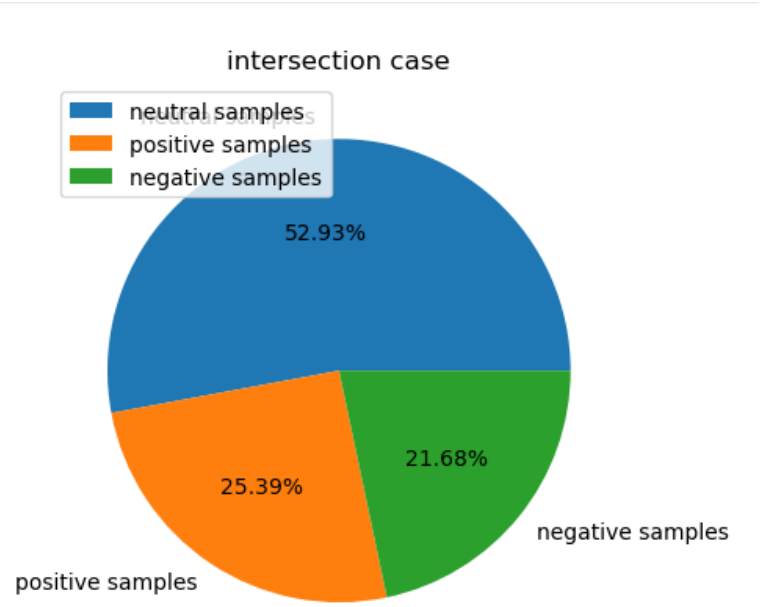
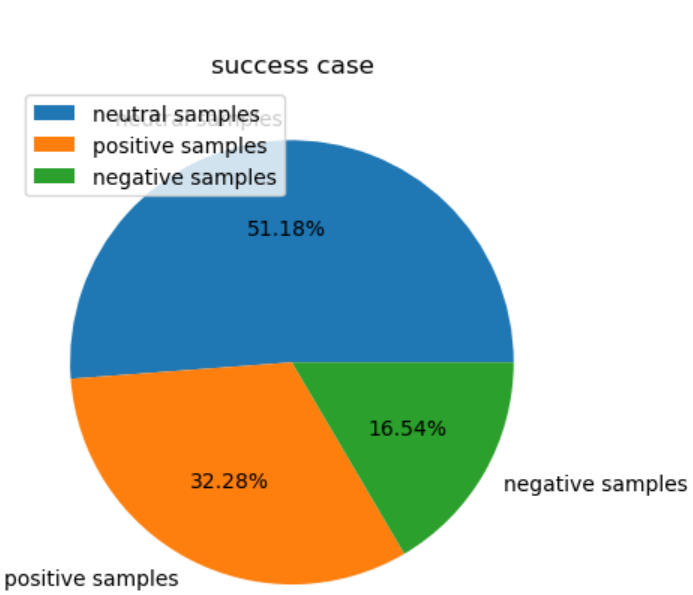


Figure 14: Intersection Samples Distribution among the Sentiments .



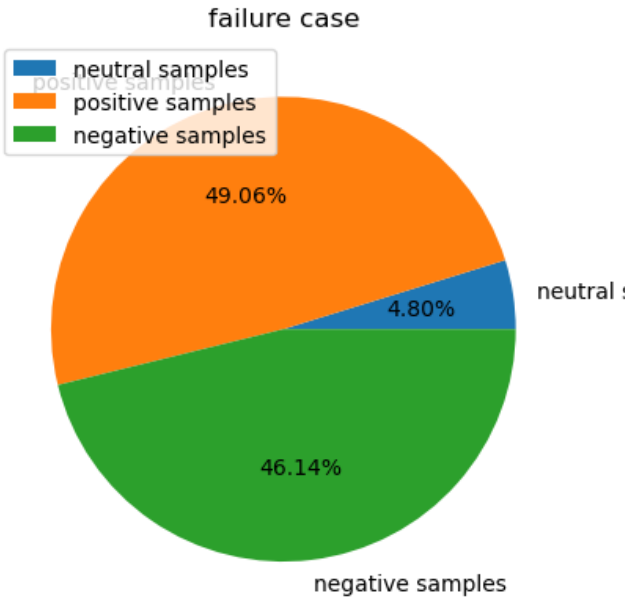


Figure 15: Success Samples Distribution among the Sentiments .

Figure 16: Failure Samples Distribution among the Sentiments .

4 Week Three Progress

4.1 Applying Jaccard Metric

As mentioned in **section 1.4** our metric in this competition is word-level Jaccard score, it is a metric used to determine the similarity between two text document means how the two text documents close to each other in terms of their context that is how many common words are exist over total words. The Jaccard Similarity score is in a range of 0 to 1. If the two documents are identical, Jaccard Similarity is 1, and score is 0 if there are no common words between two documents .

$$J(doc_1, doc_2) = \frac{doc_1 \cap doc_2}{doc_1 \cup doc_2}$$

Figure 17: Jaccard score calculation .

4.2 DistilBert pre-trained Model

We fine-tuned a DistilBERT model for question answering tasks using pre-trained models from Hugging Face. DistilBertForQuestionAnswering is a model built on top of the DistilBERT architecture, specifically designed for question answering tasks. It takes advantage of the powerful capabilities of DistilBert in natural language understanding and adds a custom tailored output layer designed for question answering.
Leveraging the flexibility of PyTorch Lightning, we created a custom dataset, module, and model to optimize our performance. Our approach enabled us to refine the model's capabilities and better suit it to our specific application requirements.[Huga]

4.2.1 Technical Approach

We created a custom dataset called TweetExtractionData for training our model as a question-answering model using the Transformers library in PyTorch. AutoTokenizer and DistilBert For Question Answering classes are imported .

4.1.1.1 Tweet Extraction Custom Dataset

Tweet Extraction Data is our custom dataset; it is a subclass of the torch.utils.data.Dataset class. The dataset consists of three main components: an initialization function **(init)**, a length function **(len)**, and a sampling

function **(getitem)**. The init function initializes instance variables such as the tokenizer and data, while the len function returns the total number of rows in the dataset. The getitem function retrieves a single sample from the dataset, preprocesses the tweet text using the tokenizer, extracts the answer span, and creates a dictionary containing various information about the tweet, including the sentiment, text, answer, labels, start position, end position, input IDs (tokenized sequence), and attention mask.

- **Tokenizer**

  We load tokenizer from Auto Tokenizer using the pre-trained distilbert base uncased model. It takes the sentiment (question) and text as inputs and returns a tuple of tensors, including a tokenized input sequence, attention masks, and offset mappings. The tokenized input sequence represents each token in the vocabulary, while the attention masks indicate which tokens should be attended to and which should be ignored, such as padding tokens.
- **Offset Mappings**

  Offset mappings ,returned from tokenizer, provide the starting position of each token in the original text, allowing us to calculate the start and end indices of the tokens that correspond to the answer span. These indices serve as our labels or ground truth when evaluating the performance of our model.

  Note » By utilizing the same tokenizer used by the model, we can ensure consistency in the tokenization process
- **Dictionary**

  getitem method returns a dictionary, for every row of data in the data frame.It contains various information about the tweet, including the sentiment, text, answer, labels, start position, end position, input IDs (tokenized sequence), and attention mask.

4.1.1.2 Data Module Class

Data Module Class provides a convenient way to manage data loading and preparation for PyTorch Lightning models. It uses the Tweet Extraction Data class to create the dataloaders for the training, validation, and test sets. The Data Module Class class has the following inputs:[Liga]

- Train data, val data, test data: Pandas data frames containing datasets.
- Tokenizer: A Hugging Face AutoTokenizer object.
- Batch size train, batch size val: The batch size for the training and validation dataloader.

4.1.1.3 Implementing Jaccard Score Metric

We implemented our custom Jaccard metric class that inherits from the Metric class, which provides some basic functionality for tracking metrics. It calculates the Jaccard similarity coefficient between the predicted and ground truth labels. The Jaccard similarity coefficient is a measure of overlap between two sets. In this case, the two sets are the predicted and the ground truth words.

4.1.1.4 Tweet Model Class

The Tweet Model class is a PyTorch Lightning model that is responsible for training and testing our distil Bert pre-trained model. The TweetModel class has the following methods:

- **Forward() :**
  It is fed with the input ids, attention masks, and start and end positions to the DistilBertForQuestionAnswering model and returns the loss and start and end logits.
- **Training Step() :**
  It is called for each batch of the training set. It calculates the loss and Jaccard similarity coefficient for the batch, and logs the loss and Jaccard to the progress bar and logger.
- **Validation Step() :**
  It is called for each batch of the validation set. It calculates the loss and Jaccard similarity coefficient for the batch, and logs the loss and Jaccard to the progress bar and logger.
- **Predict Step() :**
  It is called for each batch of the test set. It returns the predicted start and end positions for the batch.
- **Configure Optimizers() :**
  It returns Adam optimizer.

4.3 Distil Bert Model Output Analysis

4.3.1 Loss and Jaccard Scores

After training the model with 100 epoch we get these results:

- **Loss :** changed from 1.097 to 0.001143 for training and from 0.915 to 2.88 for validation
- **Jaccard :** started with 64.86% and finally we got 94.68% for training and started with 66% and ended up with 68.28% for validation data .

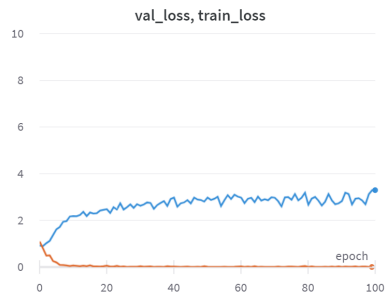


Figure 18: Training and Validation loss

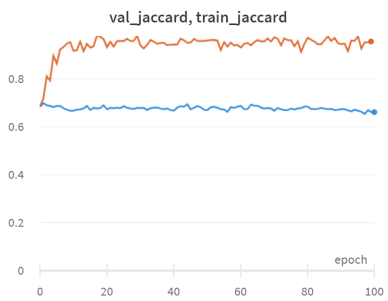


Figure 19: Training and Validation Jaccard Score

4.3.2 Mean Jaccard Values

Mean Jaccard Score for all data	0.6811
Mean Jaccard Score for neutral data	0.9697
Mean Jaccard Score for positive data	0.4654
Mean Jaccard Score for negative data	0.5018

Table 1: DistilBERT Model resulted total mean Jaccard scores across the test data.

4.3.3 Failure and Success cases

Figures 21 and 20 present examples illustrating both failed cases with a Jaccard score of 0 and successful cases with a Jaccard score of 1.

Predicted	Ground Truth	jaccard score	sentiment	text
clive it ' s my birthday pat me	Clive it ' s my birthday pat me	1.0	neutral	Clive it ' s my birthday pat me
is texting	is texting	1.0	neutral	is texting
do you have any idea when the (not so)...	Do you have any idea when the (not so) pat...	1.0	neutral	Do you have any idea when the (not so) pat...
tell him where . . .	Tell him where . . .	1.0	neutral	Tell him where . . .
ohshnapss . is she pissed at blair as usual...	OHSHNAPSS . is she pissed at blair as usual ...	1.0	neutral	OHSHNAPSS . is she pissed at blair as usual ...
...	...	...	...	...
excited	excited	1.0	positive	bit excited are u bradie lol
just investigated whether i could change my us...	Just investigated whether I could change my us...	1.0	neutral	Just investigated whether I could change my us...
no one i know likes boiled peanuts t .	no one I know likes boiled peanuts t .	1.0	neutral	no one I know likes boiled peanuts t .
aww) where ' d you get that ? hugh i...	aww) where ' d you get that ? hugh is so t...	1.0	neutral	aww) where ' d you get that ? hugh is so t...
. . i ' m a buffalo worshipper	. . i ' m a buffalo worshipper . . . may...	1.0	neutral	. . i ' m a buffalo worshipper . . . may...

Figure 20: Cases with jaccard score of one

Predicted	Ground Truth	Jaccard score	sentiment	text
document is not yet complete	dirty	0.0	negative	Want to get my hands dirty with FubuMVC . bu...
well thats even worse	it gets hurt everyone is in pain ,	0.0	negative	Well thats even worse cuz when it gets hurt ev...
good	goo	0.0	positive	good point ! ! Mine is on its way . How did...
insult	poo	0.0	negative	was gonna go to my brothers show but still fee...
shut up fool . .	? i dontlike the fact	0.0	negative	shut up fool . . . . where you been at ? ...
useful	Perfect	0.0	positive	Could be useful Tutorials & Resources for...
cute matt .	thats a good movie	0.0	positive	ZOMG SO CUTE MATT . INSULT thats a good movie
i love music so much	pain	0.0	negative	I love music so much that i ' ve gone through ...
i incredily love	entertaining ,	0.0	positive	i incredily love reading your tweets ! they '...
than	Thanx sis	0.0	positive	Thanx sis I ' ll b sure to let them know how m...

Figure 21: Cases with jaccard score of zero

4.4 Base Model Modifications

4.4.1 Architecture Modifications

In week two we used GRU network to recognize data's sequential characteristics and patterns but in this week we tried Bidirectional LSTM network instead of GRU which give us more accurate results

the next figure show the new architecture :

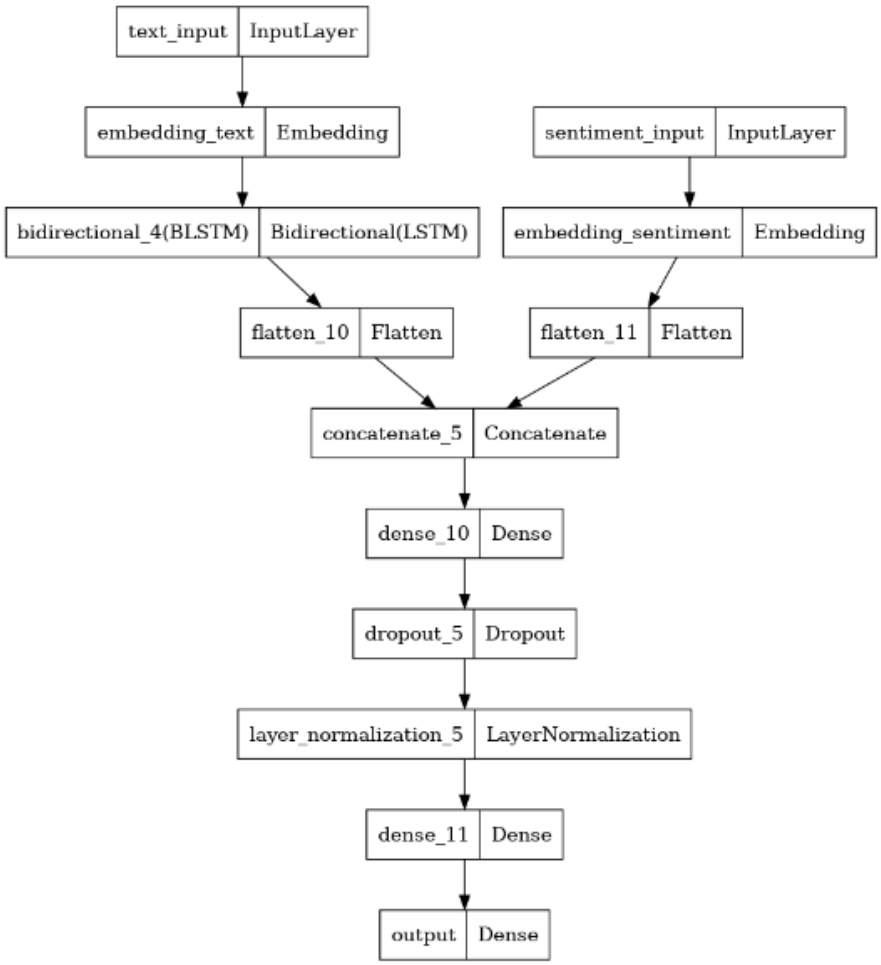


Figure 22: Model Architecture with Bi-LSTM .

4.4.2 GRU performance vs. Bi-LSTM performance

after training both models with 150 epoch we get the next results

- training accuracy for both models approximately the same .
- validation acuuracy for Bi-LSTM model reached 81.68% but GRU was 79.5% .
- training loss for Bi-LSTM model dropped to 27.22 , and for GRU decreased to 34.7 .
- validation loss for both models was about 39.27

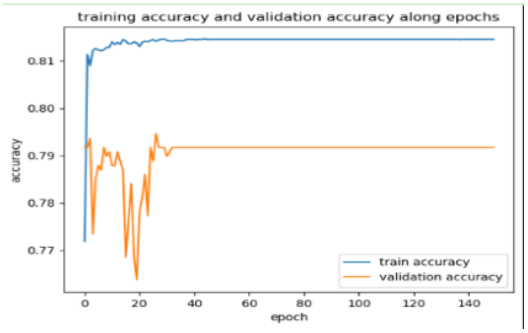


Figure 23: GRU accuracy along epochs plot .

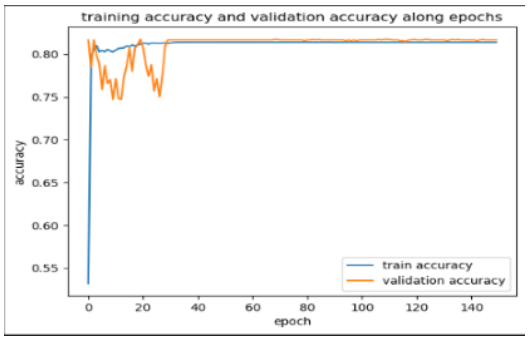


Figure 24: Bi-LSTM accuracy along epochs Plot .

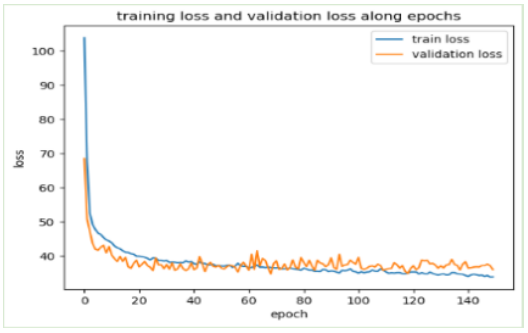


Figure 25: GRU loss along epochs plot .

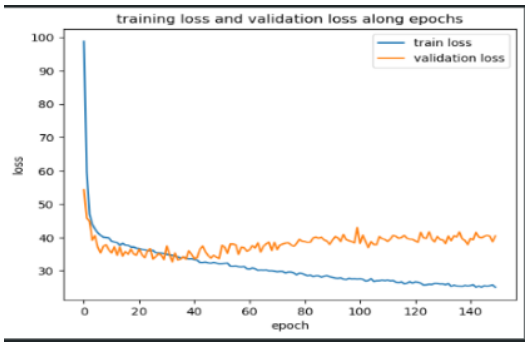


Figure 26: Bi-LSTM loss along epochs Plot .

Using the weights of training for prediction on the test dataset we get :

- **success cases " perfectly predicted " :** for GRU case we got 108 sample ,for Bi-LSTM we got 214 .
- **Intersection case "predict correctly only part of the prediction " :** for GRU case we got 3550 sample but for Bi-LSTM we got 3723 .
- **Failure case :** for GRU case we got 1524 sample but for Bi-LSTM we got 1245 .

From the previous results we can conclude that Bi-LSTM model performance is better than GRU model performance on both training and testing data .

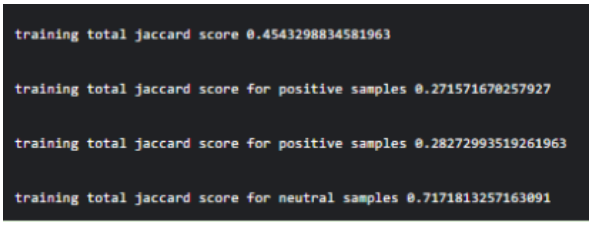


Figure 27: GRU training Jaccard scores .

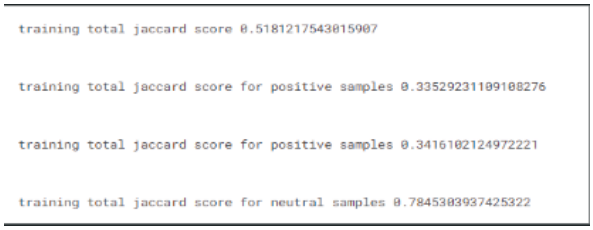


Figure 28: Bi-LSTM training Jaccard scores .

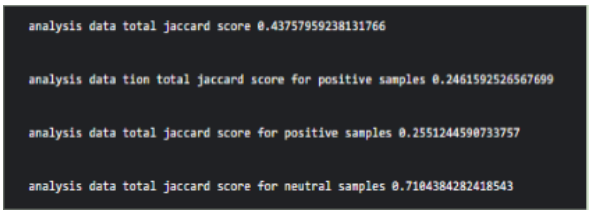


Figure 29: GRU testing "analysis" Jaccard scores .

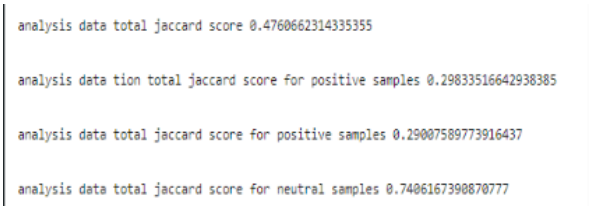


Figure 30: Bi-LSTM testing "analysis" Jaccard scores .

The previous figures show the total jaccard score for each model and the scores for each sentiment separately .

4.4.3 Bi-LSTM model Jaccard scores Analysis

By analyzing the achieved scores on testing data and plotting the distribution of that scores we can notice :

- the model prediction on the neutral data is more efficient than the positive and negative data and we can notice that from the jaccard scores of neutral as we achieved about 0.74 on the other hand we achieved 0.29 and 0.3 on positive and negative samples .

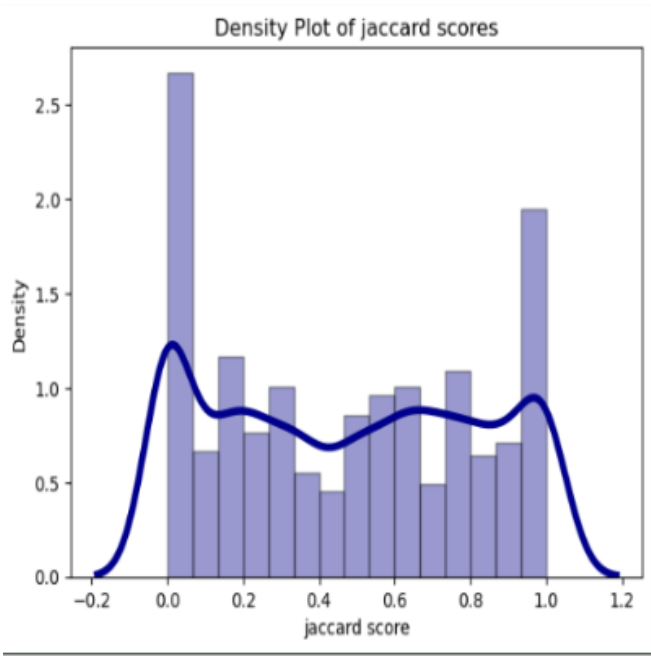


Figure 31: Jaccard scores Distribution for testing data .

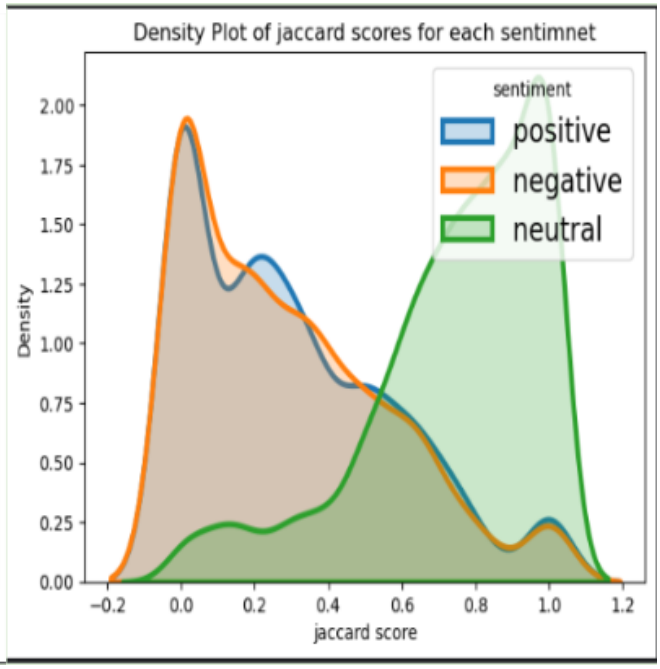


Figure 32: Jaccard scores Distribution for each sentiment on testing data .

4.4.4 Failure and Success Cases from Training and Testing Data

	text	sentiment	predicted_start	predicted_end	predicted_selected_text	true_start	true_end	true_selected_text	jaccard
6329	wait and see be sarcy as usual	negative	1	3	and see be	4	4	sarcy	0.000000
24729	haha hey , well if you be in the elevator yo...	neutral	0	16	haha hey , well if you be in the elevator you ...	0	15	haha hey , well if you be in the elevator you ...	1.000000
19602	such a fun night where in the hell bekah just ...	positive	0	1	such a	2	2	fun	0.000000
2530	I love youu nick santino ! thirteen dayss	positive	0	2	I love youu	0	2	I love youu	1.000000
8389	be really , really bored . . . I guess I...	negative	1	3	really , really	0	5	be really , really bored .	0.400000
9526	lol ! thank glad I have the tear in my eye o...	neutral	0	29	lol ! thank glad I have the tear in my eye off...	0	32	lol ! thank glad I have the tear in my eye off...	0.892857
864	well now I do	neutral	1	3	now I do	0	3	well now I do	0.750000
26336	be charleston bind for the day	neutral	1	5	charleston bind for the day	0	5	be charleston bind for the day	0.833333
21364	makeup + cute dress = I ante meridiem am...	positive	1	4	+ cute dress =	2	3	cute dress	0.500000
17550	woo I ante meridiem amplitude modulation s...	positive	0	1	woo I	0	1	woo I	1.000000
24501	man I need find a siitter val still be not ins...	negative	6	13	val still be not insult wit I lol	10	10	insult	0.125000
2580	sorry time out hear that man he be be the insu...	negative	1	4	time out hear that	0	0	sorry	0.000000

Figure 33: Some Samples with prediction and corresponding Jaccard score from training data .

	text	sentiment	predicted_start	predicted_end	true_start	true_end	predicted_selected_text	true_selected_text	jaccard
0	insult . . . I have get significant other ...	neutral	0	28	0	29	insult . . . I have get significant other shou...	insult . . . I have get significant other shou...	0.956522
1	I ake information technology you be at work th...	positive	6	13	0	19	at work then and not laze at home	I ake information technology you be at work th...	0.368421
2	I feel like miss you dream just got crush	negative	2	6	0	8	like miss you dream just	I feel like miss you dream just got crush	0.555556
3	want time out go home .	neutral	1	4	0	5	time out go home	want time out go home .	0.666667
4	listen time out dashboard confessional & cou...	neutral	0	15	0	18	listen time out dashboard confessional & count...	listen time out dashboard confessional & count...	0.882353
5	heaven , not good I can empathise . finger...	positive	2	5	0	1	not good I can	heaven ,	0.000000
6	poor baby girl chloe be freak out because of a...	neutral	0	31	0	13	poor baby girl chloe be freak out because of a...	poor baby girl chloe be freak out because of a...	0.541667
7	star trek wait and see grtsat bggete drunk now	positive	2	5	0	8	wait and see grtsat	star trek wait and see grtsat bggete drunk now	0.444444
8	you well come back soon !	positive	1	3	0	5	well come back	you well come back soon !	0.500000
9	I kno I ante meridiem amplitude modulation ...	neutral	1	31	0	32	kno I ante meridiem amplitude modulation behin...	I kno I ante meridiem amplitude modulation beh...	0.964286
10	I really wish I could make it ! a hr . dri...	positive	6	13	0	7	it ! a hr . drive just be	I really wish I could make it !	0.153846
11	work day of	neutral	0	2	0	2	work day of	work day of	1.000000

Figure 34: Some Samples with prediction and corresponding Jaccard score from testing data .

4.5 Roberta Base Model

4.5.1 Preparing Input

Before feeding the network with the training data, Five steps had to be done. Tokenizing, padding, alignment and encoding

- ByteLevelBPT Tokenzier from Hugging face python package was used, with roberta base vocab and merges from Kaggle Datasets. The input has to be tokenized so that every token has an id that is encoded with it. Sentiments are encoded into ids using the tokenizer.
- Padding was done to ensure that all examples has a length of maximum 128 tokens, having a constant length of inputs is crucial to the model due to it being a sequential model.
- Text alignment was important so that the model focuses only on the input text, with the help of attention mask, the model is able to train only on start and end indexes of the input text.
- Inputs, sentiments, starting indexes and ending indexes are encoded using the tokenizer as 2d numpy arrays with the first dimension as the size of the training data, the second dimension as MAX length of 128 tokens.

The previous operations was also done on the validation data and on the test data. The data was encapsulated as tensorflow datasets to utilize shuffling and batching provided by the tensorflow packages.

4.5.2 Model architecture

Tensorflow pre-trained Roberta Base Model was used wtih its pre-trained configurations. The input was input ids and attention mask ids, that were encoded using the tokenizer as was explained in the previous section. The start and end logits that is output of the Roberta Base Model were used as inputs to various layers making two branches of sequential layers, the first one is to predict the starting index of the select text and the second one is to predict the ending index.

- Dropout layers are used to reduce over-fitting and to ensure that the model generlizes on the training data, validation and test data.
- Convolutional layers are used to extract and generalize information from the text, just as it is used with images.
- Leaky RLU as a activation function after the conv layers
- Softmax Activation function is used an output layer, each output of the two branches classifying from 128 classes.

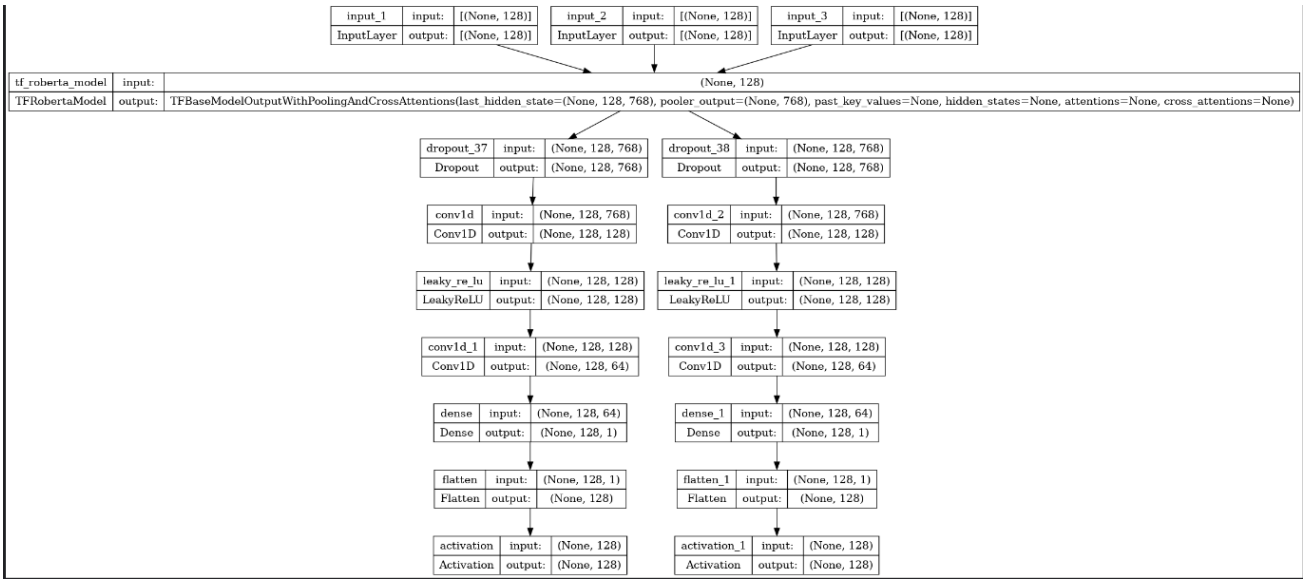


Figure 35: Roberta Base Model Architecture

4.5.3 Training

- The model was trained with 20 epochs and validated using the validation data
- Categorical Cross Entropy is used as a function loss. Label smoothing is used with hyper-parameter set to 0.2 to reduce the noise of the labels and overconfidence.
- Learning rate is set to 0.00003.
- ADAM is used as an optimize.

```
Epoch 1/20
653/653 [==============================] - 384s 502ms/step - loss: 4.5534 - activation_loss: 2.2946 - activation_1_loss: 2.2588 - val_loss: 4.2088 - val_activation_loss: 2.1107 - val_activation_1_loss: 2.0981
Epoch 2/20
653/653 [==============================] - 318s 486ms/step - loss: 4.1622 - activation_loss: 2.1010 - activation_1_loss: 2.0612 - val_loss: 4.1809 - val_activation_loss: 2.1004 - val_activation_1_loss: 2.0806
Epoch 3/20
653/653 [==============================] - 317s 486ms/step - loss: 4.0394 - activation_loss: 2.0392 - activation_1_loss: 2.0002 - val_loss: 4.1997 - val_activation_loss: 2.0928 - val_activation_1_loss: 2.1068
Epoch 4/20
653/653 [==============================] - 317s 486ms/step - loss: 3.9642 - activation_loss: 2.0000 - activation_1_loss: 1.9642 - val_loss: 4.2235 - val_activation_loss: 2.1069 - val_activation_1_loss: 2.1166
Epoch 5/20
653/653 [==============================] - 318s 486ms/step - loss: 3.8483 - activation_loss: 1.9445 - activation_1_loss: 1.9038 - val_loss: 4.3114 - val_activation_loss: 2.1540 - val_activation_1_loss: 2.1575
Epoch 6/20
653/653 [==============================] - 318s 487ms/step - loss: 3.7318 - activation_loss: 1.8838 - activation_1_loss: 1.8480 - val_loss: 4.3986 - val_activation_loss: 2.1910 - val_activation_1_loss: 2.2076
Epoch 7/20
653/653 [==============================] - 318s 486ms/step - loss: 3.6212 - activation_loss: 1.8282 - activation_1_loss: 1.7930 - val_loss: 4.5176 - val_activation_loss: 2.2526 - val_activation_1_loss: 2.2650
Epoch 8/20
653/653 [==============================] - 317s 486ms/step - loss: 3.5105 - activation_loss: 1.7697 - activation_1_loss: 1.7408 - val_loss: 4.6247 - val_activation_loss: 2.3049 - val_activation_1_loss: 2.3198
Epoch 9/20
653/653 [==============================] - 317s 486ms/step - loss: 3.4086 - activation_loss: 1.7164 - activation_1_loss: 1.6922 - val_loss: 4.7343 - val_activation_loss: 2.3314 - val_activation_1_loss: 2.4029
Epoch 10/20
653/653 [==============================] - 317s 486ms/step - loss: 3.3280 - activation_loss: 1.6722 - activation_1_loss: 1.6559 - val_loss: 4.7990 - val_activation_loss: 2.4001 - val_activation_1_loss: 2.3989
Epoch 11/20
653/653 [==============================] - 317s 486ms/step - loss: 3.2660 - activation_loss: 1.6416 - activation_1_loss: 1.6244 - val_loss: 4.7975 - val_activation_loss: 2.4084 - val_activation_1_loss: 2.3891
Epoch 12/20
653/653 [==============================] - 317s 486ms/step - loss: 3.2072 - activation_loss: 1.6092 - activation_1_loss: 1.5979 - val_loss: 4.9489 - val_activation_loss: 2.4567 - val_activation_1_loss: 2.4922
Epoch 13/20
653/653 [==============================] - 322s 493ms/step - loss: 3.1595 - activation_loss: 1.5836 - activation_1_loss: 1.5759 - val_loss: 4.9970 - val_activation_loss: 2.4938 - val_activation_1_loss: 2.5032
Epoch 14/20
653/653 [==============================] - 317s 486ms/step - loss: 3.1297 - activation_loss: 1.5657 - activation_1_loss: 1.5640 - val_loss: 5.0443 - val_activation_loss: 2.5384 - val_activation_1_loss: 2.5058
Epoch 15/20
653/653 [==============================] - 317s 486ms/step - loss: 3.0982 - activation_loss: 1.5514 - activation_1_loss: 1.5468 - val_loss: 5.1446 - val_activation_loss: 2.5669 - val_activation_1_loss: 2.5777
Epoch 16/20
653/653 [==============================] - 317s 486ms/step - loss: 3.0821 - activation_loss: 1.5441 - activation_1_loss: 1.5380 - val_loss: 5.1563 - val_activation_loss: 2.6268 - val_activation_1_loss: 2.5295
Epoch 17/20
653/653 [==============================] - 317s 486ms/step - loss: 3.0687 - activation_loss: 1.5355 - activation_1_loss: 1.5332 - val_loss: 5.1267 - val_activation_loss: 2.5616 - val_activation_1_loss: 2.5652
Epoch 18/20
653/653 [==============================] - 317s 486ms/step - loss: 3.0521 - activation_loss: 1.5284 - activation_1_loss: 1.5237 - val_loss: 5.1705 - val_activation_loss: 2.6162 - val_activation_1_loss: 2.5543
Epoch 19/20
653/653 [==============================] - 317s 486ms/step - loss: 3.0426 - activation_loss: 1.5210 - activation_1_loss: 1.5216 - val_loss: 5.2703 - val_activation_loss: 2.6464 - val_activation_1_loss: 2.6239
Epoch 20/20
653/653 [==============================] - 317s 485ms/step - loss: 3.0241 - activation_loss: 1.5138 - activation_1_loss: 1.5103 - val_loss: 5.2832 - val_activation_loss: 2.6409 - val_activation_1_loss: 2.6423
```

Figure 36: Roberta Base Model Training

4.6 Roberta Base Model Output Analysis

4.6.1 Success and Failure Cases

0	0	5634	56b9191817	If it makes you feel any better.. My Saturday...	If it makes you feel any better.. My Saturday ...	neutral	if it makes you feel any better.. my saturday...	1.0
1	1	19247	111a65cf26	`auto-resolve` is that a Geek/tech answer to ...	`auto-resolve` is that a Geek/tech answer to m...	neutral	`auto-resolve` is that a geek/tech answer to ...	1.0
2	2	12755	2a547eba64	oooh yeaah fooood time I`ve found my seat at t...	oooh yeaah fooood time I`ve found my seat at t...	neutral	oooh yeaah fooood time i`ve found my seat at ...	1.0
3	3	13071	db5466d7e4	P9 for Danica and your team... Not the end of ...	P9 for Danica and your team... Not the end of ...	neutral	p9 for danica and your team... not the end of...	1.0
6	6	13116	d3889fedf4	not really sure. need to deposit and save som...	not really sure.	negative	not really sure.	1.0
...	...	...	...	...	...	...	...	...
5490	5490	25629	dde52024e4	Finally at home. Who decides it`s time for mor...	Finally at home. Who decides it`s time for mor...	neutral	finally at home. who decides it`s time for mo...	1.0
5491	5491	18539	013b686a0a	i`m searching followers	i`m searching followers	neutral	i`m searching followers	1.0
5492	5492	11516	b2c78b1572	goodnight twitter, ill see you after 10 + hour...	goodnight twitter, ill see you after 10 + hour...	neutral	goodnight twitter, ill see you after 10 + hou...	1.0
5494	5494	23739	0f0ce4a8fb	I`m so pumped for the day!	I`m so pumped for the day!	neutral	i`m so pumped for the day!	1.0
5495	5495	23891	844c3344c5	Sweetie, if you refuse to offend, who will?	Sweetie, if you refuse to offend, who will? W	neutral	sweetie, if you refuse to offend, who will? w	1.0

Figure 37: Roberta Base Success Cases

2	2	4582	da68fc0719	dude, I can safely say I was blown away when ...	dude, I can safely say I was blown away when I...	positive	blown away	0.181818
3	3	7793	5344597aef	this is very true about ! but you do have to ...	funny!	positive	was pret-ty funny! im	0.250000
5	5	906	caf284161d	Star trek was good times.	good times.	positive	good	0.500000
6	6	26230	01cf51125c	check out review for the movie Fighting - htt...	Hilarious	neutral	check out review for the movie fighting - htt...	0.000000
7	7	23230	e9778e805e	Content content content .... gah! Story of my...	- thx	positive	- thx for reminding me.	0.400000
...	...	...	...	...	...	...	...	...
5489	5489	9239	7544811702	_precious06 sooo mad	_precious06 sooo mad	neutral	sooo mad	0.666667
5492	5492	9266	4df8509eb1	I`m watching some of your videos in YouTube. ...	I`m watching some of your videos in YouTube. Y...	positive	. you`re funny david. oh and talented	0.375000
5493	5493	25141	0b1abd00c2	I promise to post new mini magical village tod...	the weather is perfect for it	positive	perfect for it	0.500000
5494	5494	24495	ead4e636bc	happened about three weeks ago. Why, is there...	, is there a serial card fraudster on the loose?	negative	serial card fraudster on the loose?	0.600000
5496	5496	24915	2632db8d87	ohhh ok. thats upsetting sorry for wasting y...	. thats upsetting sorry	negative	thats upsetting	0.500000

Figure 38: Roberta Base Failure Cases

4.6.2 Loss and Jaccard Scores

After training the model with 20 epochs we get these results

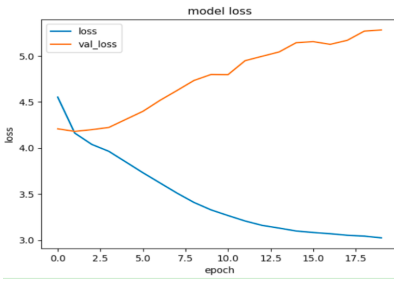


Figure 39: Training and Validation loss

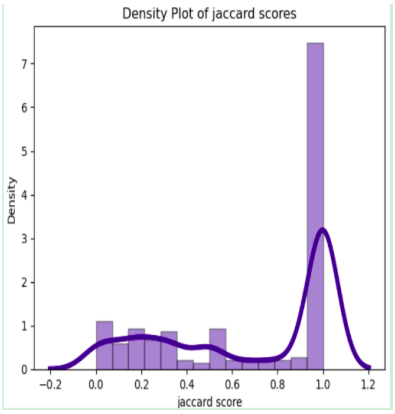


Figure 40: Training and Validation Jaccard Score

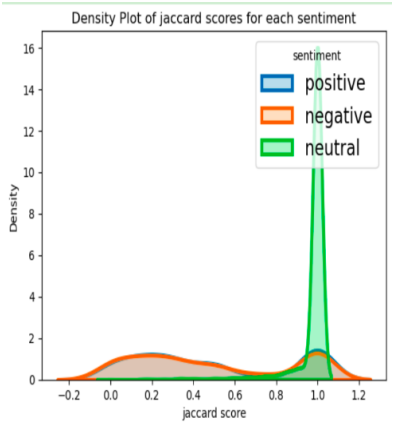


Figure 41: Training and Validation Jaccard Score

Mean Jaccard Score for all data	0.6852
Mean Jaccard Score for neutral data	0.9718
Mean Jaccard Score for positive data	0.4916
Mean Jaccard Score for negative data	0.4828

Table 2: The resulted total mean Jaccard scores across the test data

5

Week Four Progress

5.1 Tiny Roberta pre-trained Model

The "tinyroberta-squad2" model is a distilled variant of the "deepset/roberta-base-squad2" model, which has been specifically fine-tuned using the Squad2.0 dataset. This distilled version serves as a more compact and efficient alternative to the original "roberta" model while retaining its question-answering capabilities.
It incorporates a span classification head on top, enabling it to extract answers from given passages for extractive question-answering task. For our specific problem, we performed fine-tuning on the aforementioned "tinyroberta-squad2" model , then we treated the sentiment as the question and the text as the context from which we extracted the answer.[Hugb]

5.1.1 Model Architecture

We utilized the reproducibility advantage of PyTorch Lightning to effortlessly switch to Tiny Roberta model. and package imports, specifically adopting the compact RoBERTa variant. Our TweetExtractionData custom dataset, data module class, and Jaccard score metric remained unaltered from distil Bert . The sole modification occurred within the Tweet model class, where we defined pretrained tinyroberta AutoTokenizer and AutoModelForQuestionAnswering .
TinyRoberta model takes the sentiments,text,ground truth start and end positions as inputs and calculate loss and predicted output .[Ligb]

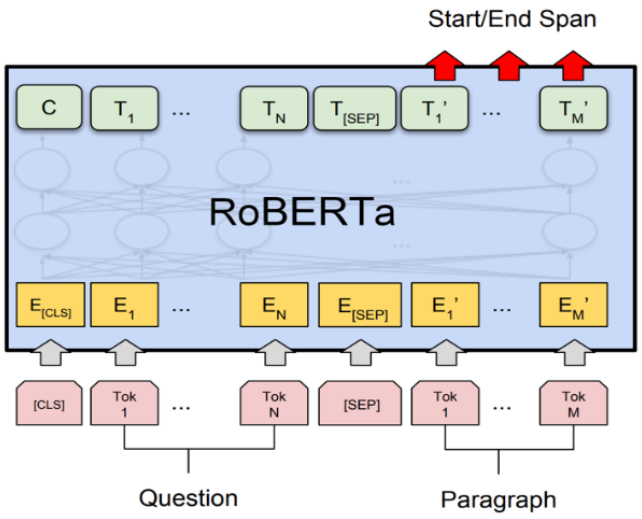


Figure 42: Roberta for QA task

5.2 Tiny Roberta Model Output Analysis

5.2.1 Loss and Jaccard Scores

After training the model with 100 epoch we get these results:

- **Loss :** changed from 1.202 to 0.009905 for training and from 0.9106 to 3.437 for validation
- **Jaccard :** started with 63.03% and finally we got 94.86% for training and started with 64.01% and ended up with 65.09% for validation data .

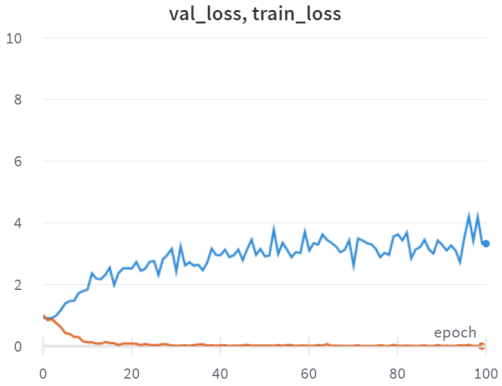


Figure 43: Training and Validation loss

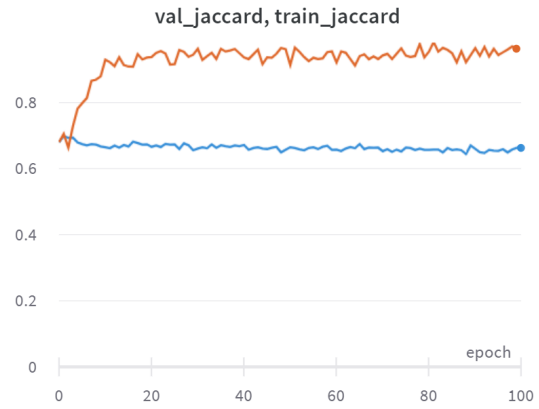


Figure 44: Training and Validation Jaccard Score

5.2.2 Mean Jaccard Values

Mean Jaccard Score for all data	0.6585
Mean Jaccard Score for neutral data	0.9596
Mean Jaccard Score for positive data	0.4472
Mean Jaccard Score for negative data	0.4567

Table 3: The resulted total mean Jaccard scores across the test data.

5.2.3 Failure and Success cases

Figure 45 presents examples illustrating both cases with a Jaccard score from 0 to 1.

Predicted	Ground Truth	Jaccard score	sentiment	alltext
wishes he could be with that special someone ...	special	0.111111	positive	wishes he could be with that special someone ....
the queen of sass oh scenie ! ! !	you the the Queen of Sass oh Scenie ! ! !	0.875000	positive	you the the Queen of Sass oh Scenie ! ! !
falling apart . '	' I only think of you as breaking my heart , ...	0.181818	negative	' I only think of you as breaking my heart , ...
she ` s been awesome	she ` s been awesome	1.000000	positive	saying goodbye to for a year she ` s been aw...
i wish	wish	0.500000	positive	Off to Dollarama - - I wish I had a new job
i hate my arrival in the employee parking lot !	I hate my arrival	0.400000	negative	I hate my arrival in the employee parking lot !
funny	funny	1.000000	positive	had a funny time at neball against PLC score ...
cool	cool beans ,	0.333333	positive	cool beans , yeah man - no prob at all
* * * * ,	Sons of * * * * ,	0.500000	negative	Sons of * * * * , why couldn ` t they pu...

Figure 45: Sample cases

5.3 Roberta Large Model

5.3.1 Roberta Large vs Roberta Base

- **Layers**: Roberta Base has 768 Hidden layers while Roberta Large has 1024
- **Parameters**: Roberta Base has approximalty 125M parameters, while Roberta Large has 355M

All that was done in Roberta Base Model Section was done without any change, to the Roberta Large Model. Making the All the steps identical including Input preparing, Model Architecture which is also the same and with same Layers used after utilizing the pre-trained model, Training with the same hyper-paremetrs, loss function, optimizer and number of epochs

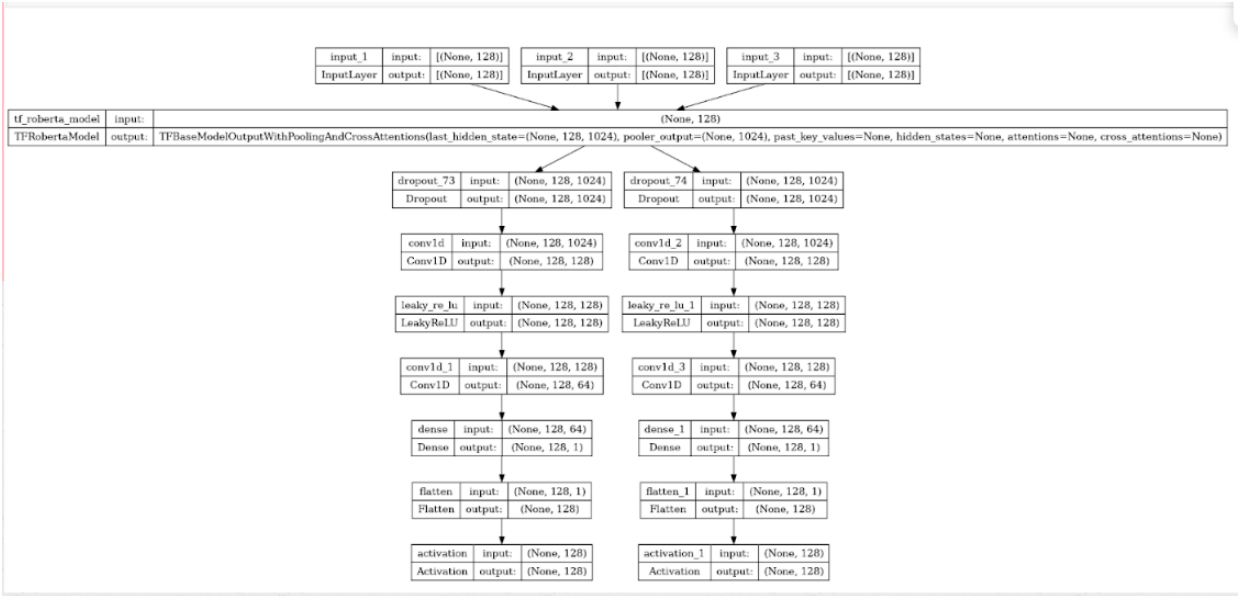


Figure 46: Roberta Large Model Architecture

5.4 Roberta Large Model Output Analysis

5.4.1 Loss and Jaccard Scores

After training the model with 20 epochs we get these results

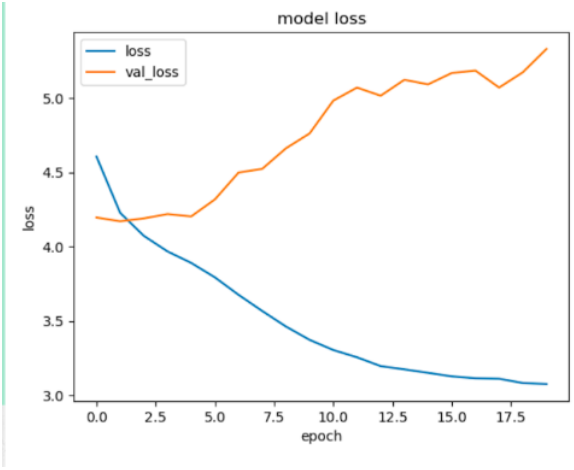


Figure 47: Training and Validation loss

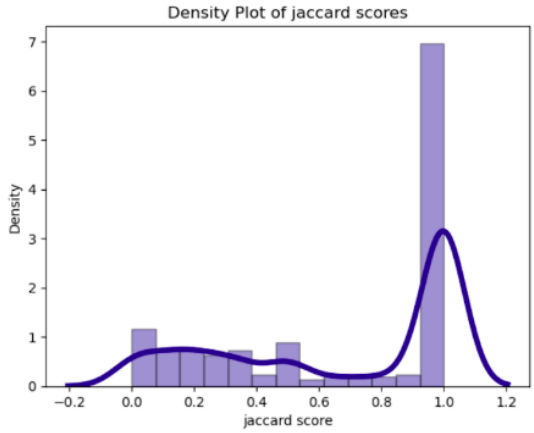


Figure 48: Training and Validation Jaccard Score

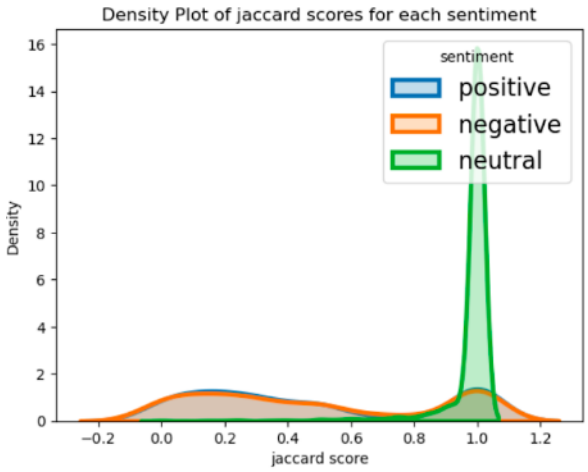


Figure 49: Training and Validation Jaccard Score

- The resulted total mean Jaccard scores across the test data was **0.681**.
- with the Positive sentiment score equals to **.475**.
- with the Negative sentiment score equals to **.482**.
- with the Neutral sentiment score equals to **.972**.

6

Comparison between all models

6.1 Mean Jaccard Score values for test data

Mean Jaccard Score for all data	0.4761
Mean Jaccard Score for neutral data	0.7406
Mean Jaccard Score for positive data	0.2983
Mean Jaccard Score for negative data	0.2901

Table 4: Base Model resulted total mean Jaccard scores across the test data.

Mean Jaccard Score for all data	0.6811
Mean Jaccard Score for neutral data	0.9697
Mean Jaccard Score for positive data	0.4654
Mean Jaccard Score for negative data	0.5018

Table 5: DistilBERT Model resulted total mean Jaccard scores across the test data.

Mean Jaccard Score for all data	0.6852
Mean Jaccard Score for neutral data	0.9719
Mean Jaccard Score for positive data	0.4916
Mean Jaccard Score for negative data	0.4829

Table 6: Roberta base resulted total mean Jaccard scores across the test data.

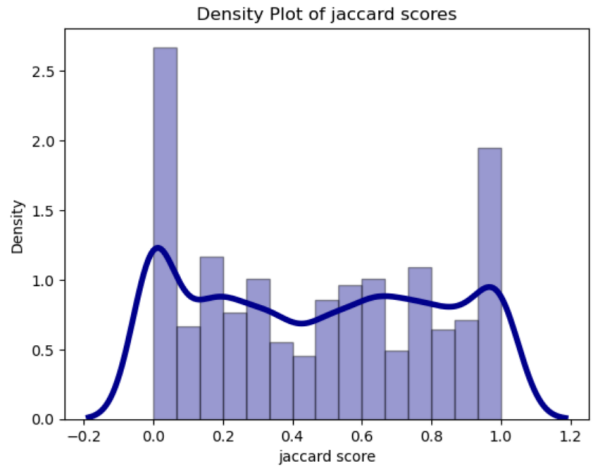
Mean Jaccard Score for all data	0.6585
Mean Jaccard Score for neutral data	0.9596
Mean Jaccard Score for positive data	0.4472
Mean Jaccard Score for negative data	0.4567

Table 7: Tiny Roberta resulted total mean Jaccard scores across the test data.

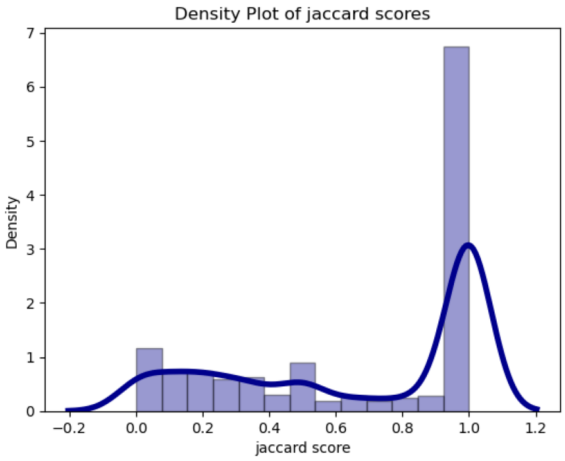
Mean Jaccard Score for all data	0.6811
Mean Jaccard Score for neutral data	0.9724
Mean Jaccard Score for positive data	0.4758
Mean Jaccard Score for negative data	0.4821

Table 8: Large Roberta resulted total mean Jaccard scores across the test data.

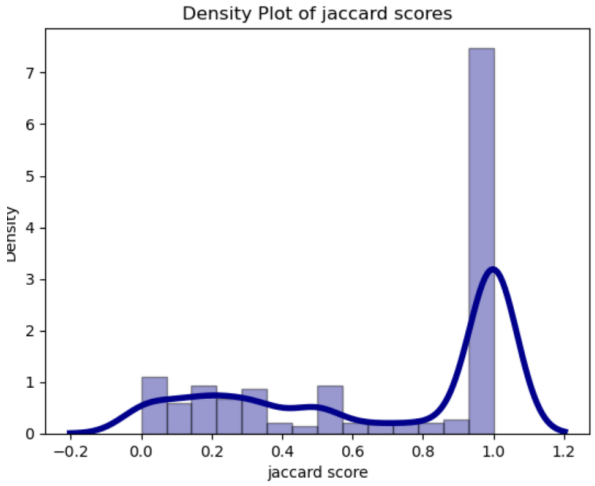
6.2 Density Plots of Jaccard Score values for test data



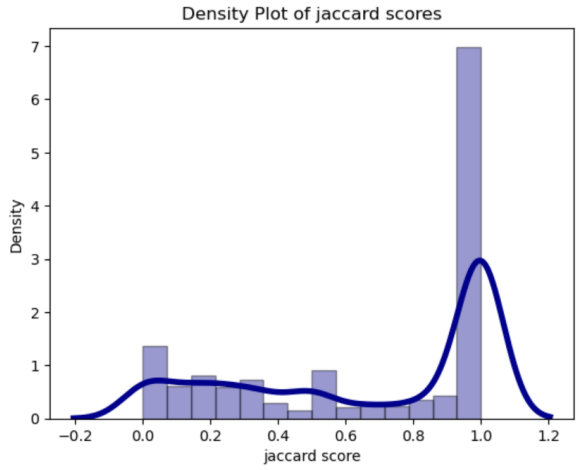
(a) Base Model



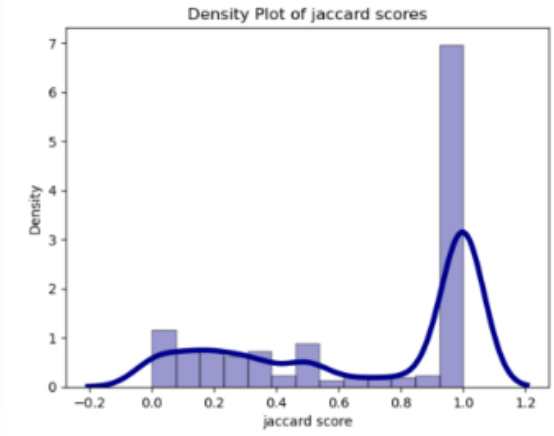
(b) DistilBert model



(c) RoBerta base model



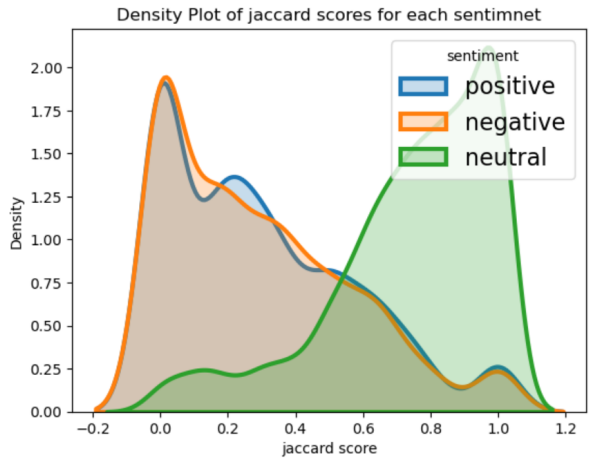
(d) Tiny RoBerta



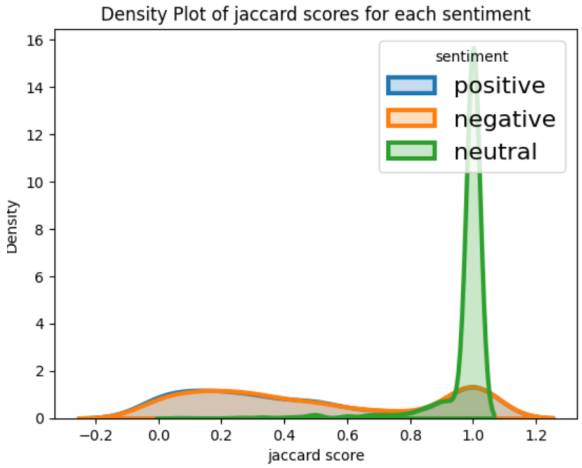
(e) Large RoBerta

Figure 50: Density of Jaccard scores accross test data

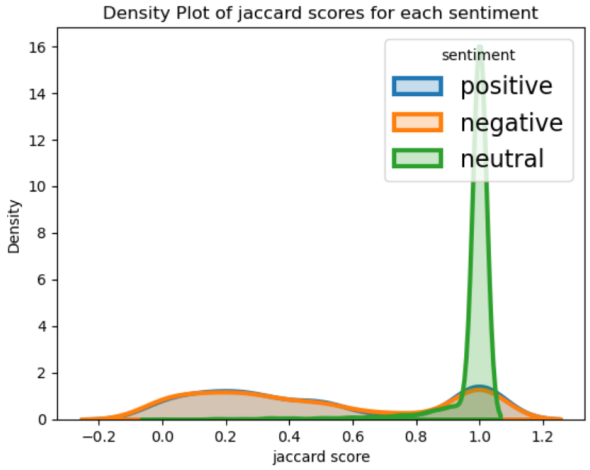
6.3 Density Plots of Jaccard Score values for each sentiment



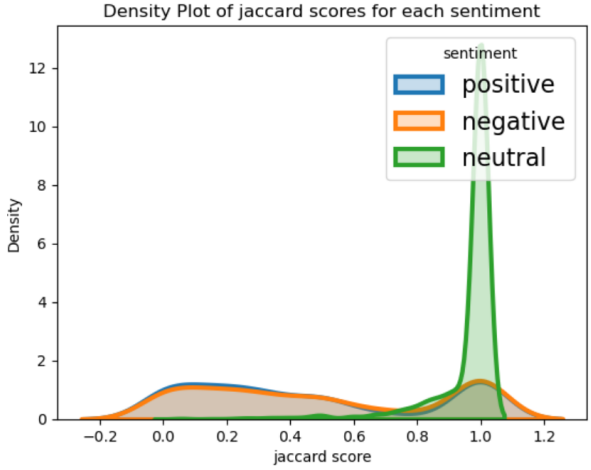
(a) Base Model



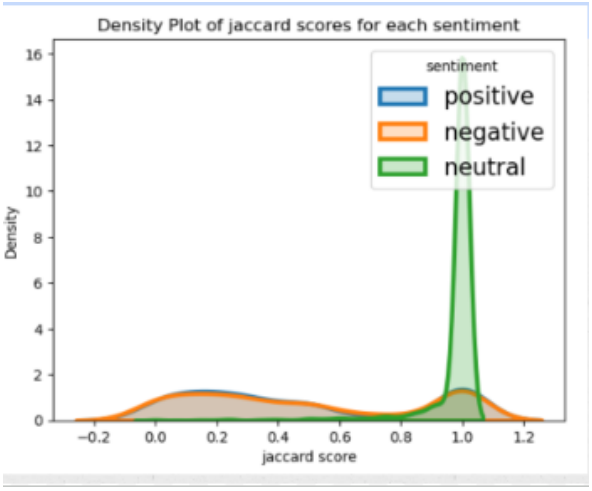
(b) DistilBert model



(c) RoBerta base model



(d) Tiny RoBerta



(e) Large RoBerta

Figure 51: Density of Jaccard scores for each sentiment accross test data

7 Conclusion

- Based on the data presented in Tables 4, 5 , 6 , 7 and 8 ,we can observe that the base model utilizing LSTM and GRU, along with the tiny Roberta model, yielded the lowest Jaccard scores. Conversely, the highest score was achieved by the Roberta base model, which exhibited only marginal differences compared to the distilbert model. While the tiny Roberta model obtained lower scores than the aforementioned models, it still outperformed the base model.
- The analysis of density plots for Jaccard scores revealed a notable difference in scores for neutral text. In order to understand the cause of this discrepancy, we investigated whether it was a result of data imbalance or other data-related issues. To explore this further, we decided to exclude neutral text and re-examine the density plots of Jaccard scores.
- Based on the analysis of figures 52 and 53 , Positive and negative classes exhibit similar scores and density even when the neutral class is excluded. This observation led us to dismiss the possibility of data imbalance as the underlying cause of the discrepancy.
- Further investigation into the original answer and selected text revealed a noteworthy finding: the existence of numerous words contributed to the same sentiment.This discovery suggests that the presence of these sentiment-inducing words might be a potential factor contributing to the observed divergence in scores.

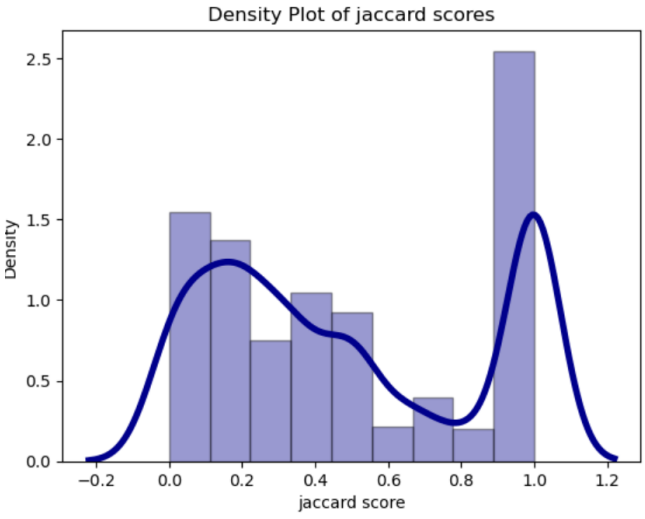


Figure 52: Jaccard scores for texts without neutral sentiment

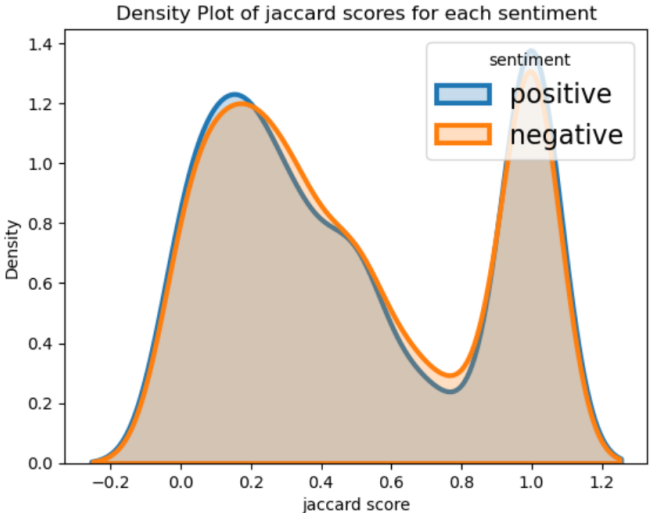


Figure 53: Jaccard scores for positive and negative sentiments

References

[Eve] Scale Virtual Events. *Guide to Text Preprocessing*. URL: %5Curlhttps://exchange.scale.com/public/blogs/preprocessing-techniques-in-nlp-a-guide.

[Huga] HuggingFace. *DistilBertForQuestionAnswering*. URL: https://huggingface.co/docs/transformers/model_doc/distilbert.

[Hugb] HuggingFace. *TinyRoberta*. URL: https://huggingface.co/deepset/tinyroberta-squad2.

[Liga] PyTorch Lightning. *Lightning Data Module*. URL: %5Curl%7Bhttps://lightning.ai/docs/pytorch/stable/data/datamodule.html%7D.

[Ligb] PyTorch Lightning. *Lightning Module*. URL: https://lightning.ai/docs/pytorch/stable/common/lightning_module.html.