UNIVERSITY OF ATHENS
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

Deep Learning for NLP

Student name: *Chatzispyrou Michail*
sdi: 1115202000212

Course: *Artificial Intelligence II (M138, M226, M262, M325)*
Semester: *Fall Semester 2023*

Contents

1. Abstract

This work studied the development of a sentiment classifier, using Feed Forward Neural Networks, for a twitter dataset regarding the Greek general elections. Each tweet consists of an ID, its actual text and lastly the class it belongs to: POSITIVE, NEUTRAL, NEGATIVE. The classifier handles all the classes. The experiments performed for this development uti- lized a variety of methods, including data processing, word embeddings and hyper-parameter tuning.

2. Data processing and analysis

2.1. Pre-processing

Pre-processing is an essential component of Artificial Intelligence (AI) and Machine Learning (ML), since it has the ability to convert raw input data into a more refined and useful form. Also, it can improve the quality and validity of the input data, which enhances both the efficiency and precision of the algorithms that come after.

An extensive data cleaning procedure was used to improve the dataset's quality and analytical appropriateness in order to create the current model. The act of removing links was crucial in getting rid of unnecessary site addresses and maintaining the dataset's primary emphasis. Similarly, by removing potential distractions from metadata, the systematic elimination of mentions and hashtags improved the dataset's readability. Text capitalization was standardized to lowercase in order to preserve uniformity throughout the dataset. Moreover, a further refinement phase required keeping only Latin or Greek-language words in order to guarantee linguistic consistency and remove non-alphabetic characters. Lemmatization was included as an additional option to the data cleaning process to further simplify the dataset in order to enable a more accurate and comprehensive analysis in the present study. Additionally, the removal of stopwords improved dataset conciseness by excluding common words with limited analytical value. Finally, accents were removed to ensure uniformity in text representation.

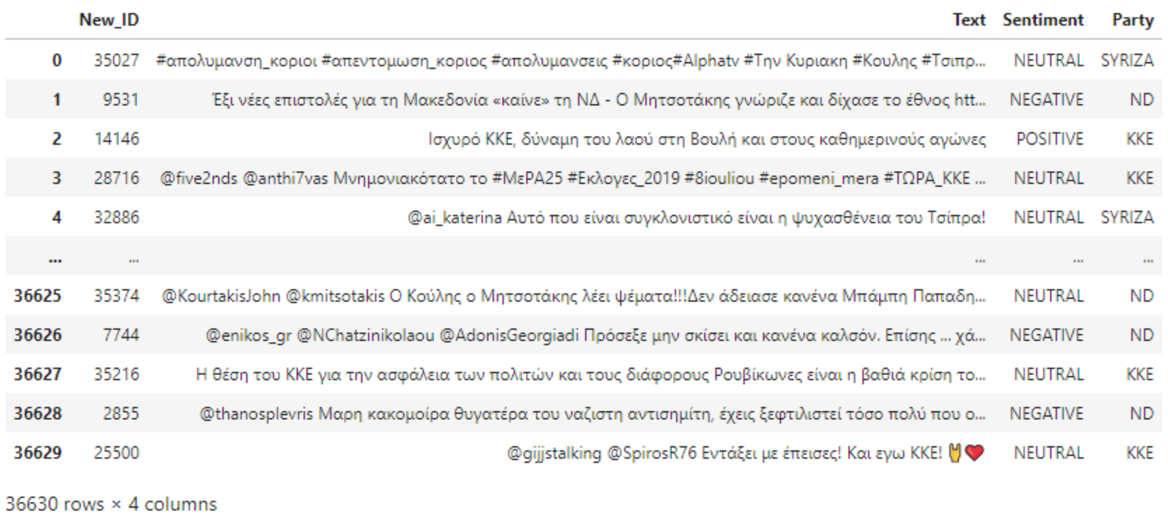
In the previous assignment, we investigated different pre-processing methods to improve our models' performance. We looked into three distinct strategies:

- No Pre-Processing (NP), the dataset was intact
- Basic Pre-Processing (BP), the dataset underwent the following modifications:
  - Removing links
  - Removing hashtags
  - Removing mentions
  - Removing stopwords
  - Removing accents
  - Lowercase
  - Removing non Latin or Greek-language words
- Basic Pre-Processing & Lemmatization (PL), all the Basic Pre-Processing with the addition Lemmatization

Following extensive testing, it became clear that, of the three approaches, Basic Pre-Processing (BP) produced the most encouraging outcomes. These results have led us to decide to simplify our strategy for this task. In order to preserve uniformity and enable a more precise comparison amongst models, we will just utilize the fundamental pre-processing method.

2.2. Analysis

We use a variety of tools to navigate the data's complexities as we begin our initial exploration phase and reveal its mysteries. Printing the dataframe (1) itself is the first step that is both straightforward and essential. Having a basic knowledge of the dataset's structure from this first look lets us examine its elements, column names, and the type of data it contains.



	New_ID	Text	Sentiment	Party
0	35027	#απολυμανση_κοριοι #απεντομωση_κοριος #απολυμανσεις #κοριος#Alphatv #Την Κυριακη #Κουλης #Τσιπρ...	NEUTRAL	SYRIZA
1	9531	Έξι νέες επιστολές για τη Μακεδονία «καίνε» τη ΝΔ - Ο Μητσοτάκης γνώριζε και δίχασε το έθνος htt...	NEGATIVE	ND
2	14146	Ισχυρό ΚΚΕ, δύναμη του λαού στη Βουλή και στους καθημερινούς αγώνες	POSITIVE	KKE
3	28716	@five2nds @anthi7vas Μνημονιακότατο το #ΜεΡΑ25 #Εκλογες_2019 #8iouliou #epomeni_mera #ΤΩΡΑ_ΚΚΕ ...	NEUTRAL	KKE
4	32886	@ai_katerina Αυτό που είναι συγκλονιστικό είναι η ψυχασθένεια του Τσίπρα!	NEUTRAL	SYRIZA
...	...	...	...	...
36625	35374	@KourtakisJohn @kmitsotakis Ο Κούλης ο Μητσοτάκης λέει ψέματα!!!Δεν άδειασε κανένα Μπάμπη Παπαδη...	NEUTRAL	ND
36626	7744	@enikos_gr @NChatzinikolaou @AdonisGeorgiadi Πρόσεξε μην σκίσει και κανένα καλσόν. Επίσης ... χά...	NEGATIVE	ND
36627	35216	Η θέση του ΚΚΕ για την ασφάλεια των πολιτών και τους διάφορους Ρουβίκωνες είναι η βαθιά κρίση το...	NEUTRAL	KKE
36628	2855	@thanosplevris Μαρη κακομοίρα θυγατέρα του ναζιστη αντισημίτη, έχεις ξεφτιλιστεί τόσο πολύ που ο...	NEGATIVE	ND
36629	25500	@gijjstalking @SpirosR76 Εντάξει με έπεισες! Και εγω ΚΚΕ! ✌❤️	NEUTRAL	KKE

36630 rows × 4 columns

Figure 1: Train DataFrame

A snapshot of our dataset, consisting of a significant 36630 rows and carefully arranged into 4 columns, appears as we print the dataframe (1). Every tweet is given a unique identity by the first column. The tweets themselves are visible in the second column, which provides an insight into the textual world. Each tweet is a story contained inside the boundaries of the dataset. Furthermore, in the third column, we find the sentiment classes of neutral, positive, and negative. The last touch is found in the fourth column, where each tweet is matched with a particular political party based on its political resonance.

Next, we turn our attention to visualization, creating a bar plot (2) that provides a broad overview of the political fields included in the dataset. This visual aid offers a brief overview of how tweets are distributed among different political parties.

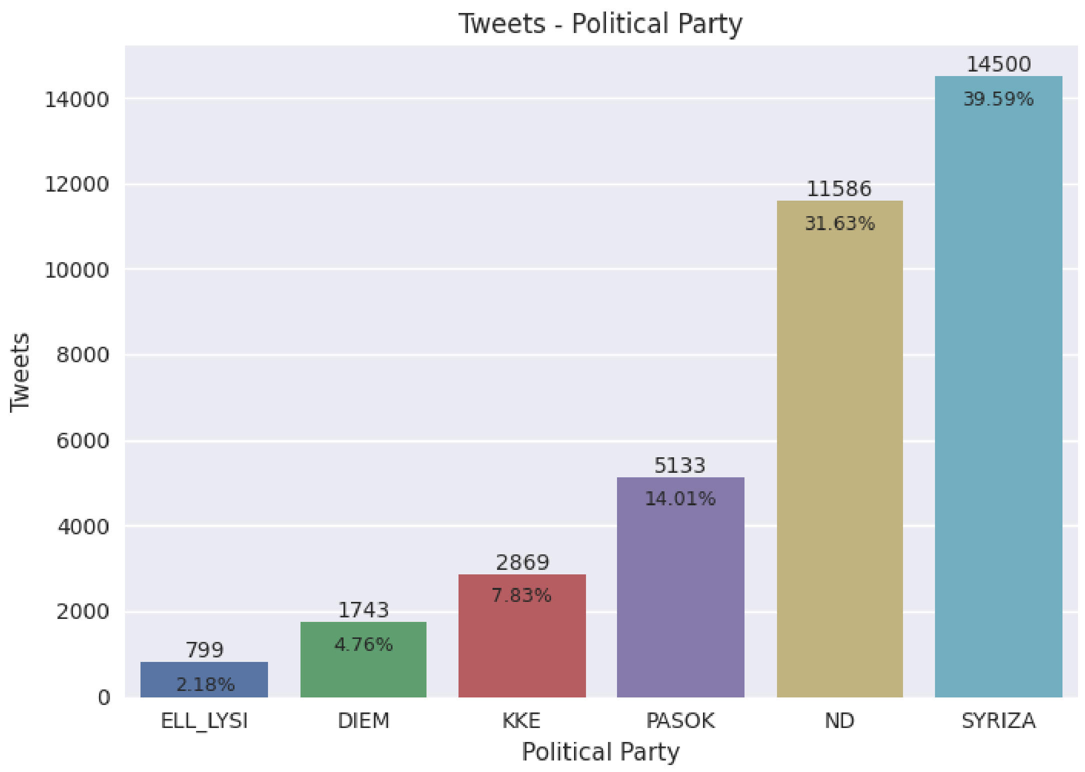


Figure 2: Tweets-Political Party Bar Plot

It can be observed that about 40% of the tweets examined deal with topics related to the political party called "SYRIZA". The New Democracy (ND) party comes in second, taking about 32% of the conversation on Twitter. The collection also contains references to "ELL_LYSI", which accounts for a little over 2% of the total and "PASOK," which accounts for a noteworthy 14%. Remarkably, the collection also includes references to "DIEM," but with a lesser frequency—less than 5%. At roughly 8%, the Communist Party of Greece (KKE) holds a significant portion.

To further our investigation, we create two separate bar plots, each of which highlights a different aspect of our dataset. The first bar plot (3) provides an in-depth comprehension of the distribution of neutral, positive, and negative attitudes inside each individual political party. At the same time, the second bar plot (4) turns our attention to a wider view, showing how tweets are distributed throughout the dataset in various sentiment categories.

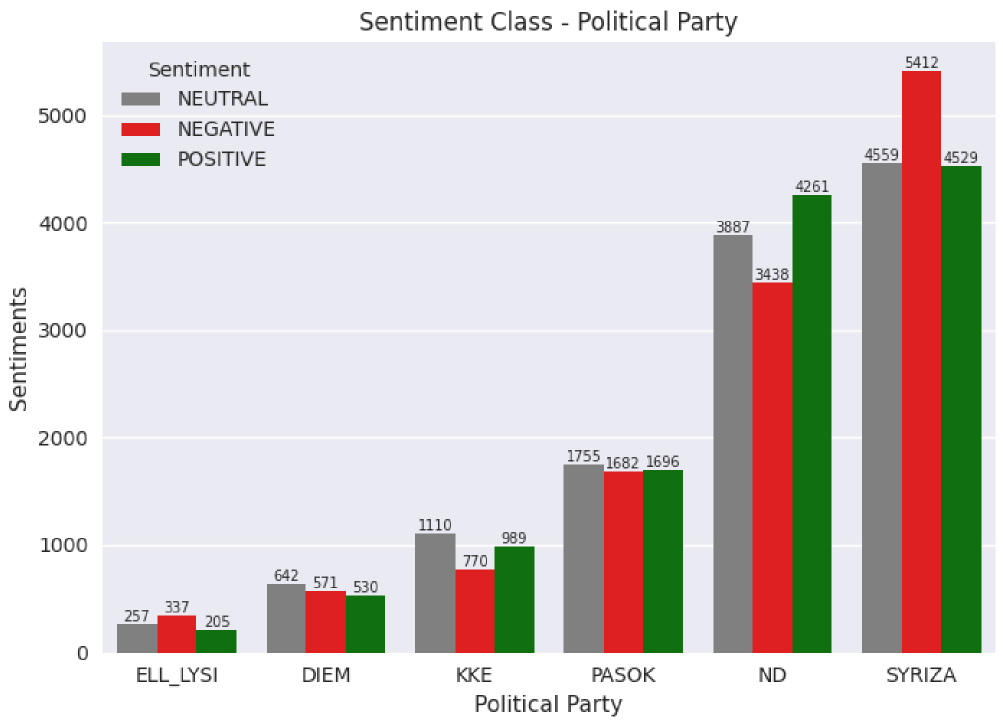


Figure 3: Sentiments-Political Party Bar Plot

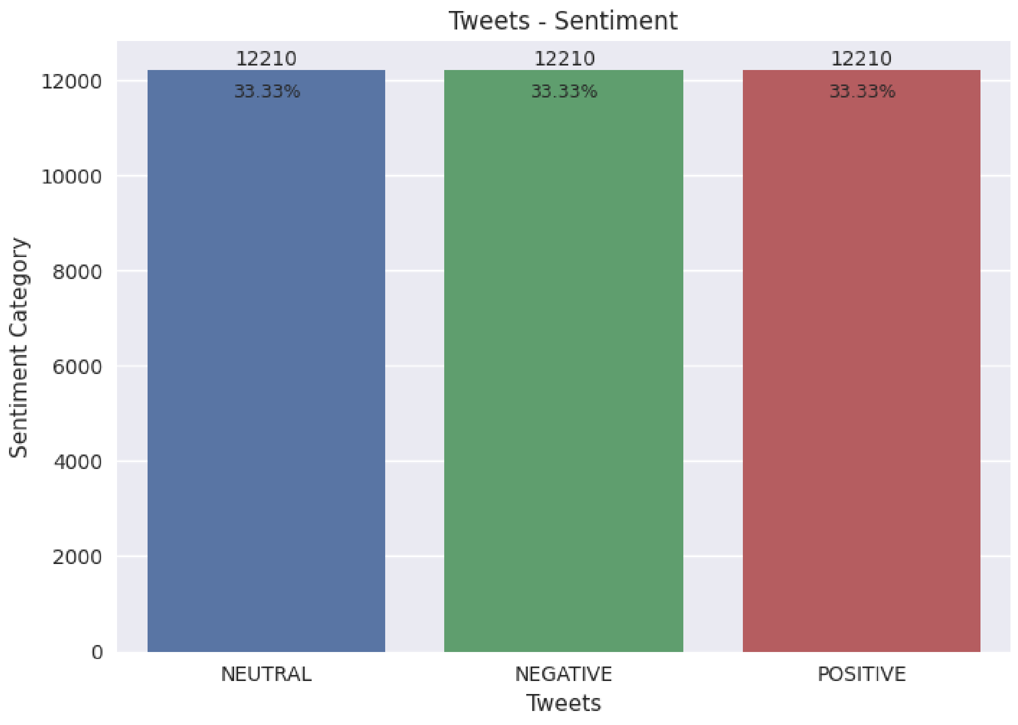


Figure 4: Tweets-Sentiment Bar Plot

Analysis reveals that while sentiment categories within each political party are not exactly balanced, they are admirably close to being balanced. This adds to an overall balanced dataset along with a fairly distributed distribution of tweets within every sentiment category. Fairness and ease of model training are two benefits of a balanced dataset. Unbiased exposure to every class ensures that no class has an undue influence on models during training. Finally, evaluation criteria that facilitate easy interpretation, such as accuracy, become more trustworthy measures of model performance.

2.3. Data partitioning for train, test and validation

The data sets chosen were the ones that the instructors provided. This decision was made in order to prevent any further complications regarding the development of the code. A ratio of 0.156 is attained by utilizing data from the validation set (5232) and training set (33630).

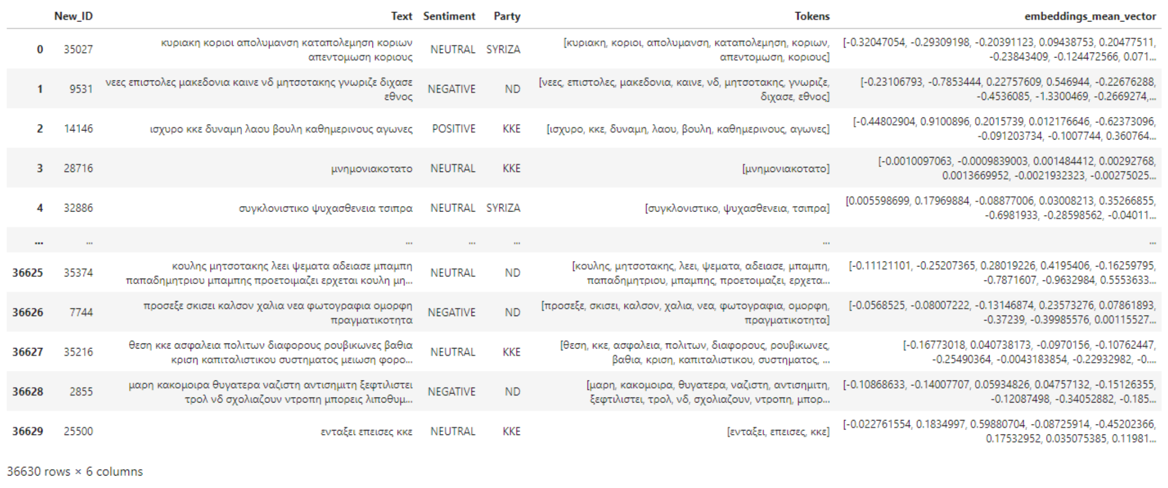
2.4. Vectorization

In our previous assignment, we looked at a variety of text representation methods to convert textual data into numerical formats appropriate for machine learning applications, including the TF-IDF vectorizer, CountVectorizer, and HashVectorizer. Every approach has advantages and disadvantages of its own, providing different insights on how best to convey the main ideas of the supporting material.

However for this project, we've gone in a new direction by using word embeddings in our study. Dense vector representations of words in a continuous vector space, where words with related meanings are positioned closer to one another, are called word embeddings. Word embeddings, in contrast to conventional vectorization algorithms, capture contextual information and semantic relationships, offering a deeper understanding of the underlying language.

We trained our model for this task using the well-known word embedding method, word2vec. Word2vec uses a given corpus's co-occurrence patterns to learn distributed representations of words. The end product is a 300-dimensional vector that captures the syntactic context and semantic meaning of each word.

We first tokenized the tweets in the pre-processing phase of our research, breaking them into individual words or tokens to make subsequent analysis easier. Building on this foundation, we have now employed a technique that makes use of the tokenization procedure to determine the mean vector for every tweet. Using the word embeddings that the word2vec model yields, we compute the mean vector for each tweet in the dataset. The mean vector is calculated by averaging the vectors representing each word in the tweet. Using this technique, we can create a representative vector for every tweet that captures all of the words' combined semantic information. These mean vectors are then saved in our dataframe.



	New_ID	Text	Sentiment	Party	Tokens	embeddings_mean_vector
0	35027	κυριακη κοριοι απολυμανση καταπολεμηση κοριων απεντομωση κοριους	NEUTRAL	SYRIZA	[κυριακη, κοριοι, απολυμανση, καταπολεμηση, κοριων, απεντομωση, κοριους]	[-0.32047054, -0.29309198, -0.20391123, 0.09438753, 0.20477511, -0.23843409, -0.124472566, 0.071...
1	9531	νεες επιστολες μακεδονια καινε νδ μητσοτακης γνωριζε διχασε εθνος	NEGATIVE	ND	[νεες, επιστολες, μακεδονια, καινε, νδ, μητσοτακης, γνωριζε, διχασε, εθνος]	[-0.23106793, -0.7853444, 0.22757609, 0.546944, -0.22676288, -0.4536085, -1.3300469, -0.2669274,...
2	14146	ισχυρο κκε δυναμη λαου βουλη καθημερινους αγωνες	POSITIVE	KKE	[ισχυρο, κκε, δυναμη, λαου, βουλη, καθημερινους, αγωνες]	[-0.44802904, 0.9100896, 0.2015739, 0.012176646, -0.62373096, -0.091203734, -0.1007744, 0.360764...
3	28716	μνημονιακοτατο	NEUTRAL	KKE	[μνημονιακοτατο]	[-0.0010097063, -0.0009839003, 0.001484412, 0.00292768, 0.0013669952, -0.0021932323, -0.00275025...
4	32886	συγκλονιστικο ψυχασθενεια τσιπρα	NEUTRAL	SYRIZA	[συγκλονιστικο, ψυχασθενεια, τσιπρα]	[0.005598699, 0.17969884, -0.08877006, 0.03008213, 0.35266855, -0.6981933, -0.28598562, -0.04011...
...	...	...	...	...	...	...
36625	35374	κουλης μητσοτακης λεει ψεματα αδειασε μπαμπη παπαδημητριου μπαμπης προετοιμαζει ερχεται κουλη μη...	NEUTRAL	ND	[κουλης, μητσοτακης, λεει, ψεματα, αδειασε, μπαμπη, παπαδημητριου, μπαμπης, προετοιμαζει, ερχετα...	[-0.11121101, -0.25207365, 0.28019226, 0.4195406, -0.16259795, -0.7871607, -0.9632984, 0.5553633...
36626	7744	προσεξε σκισει καλσον χαλια νεα φωτογραφια ομορφη πραγματικοτητα	NEGATIVE	ND	[προσεξε, σκισει, καλσον, χαλια, νεα, φωτογραφια, ομορφη, πραγματικοτητα]	[-0.0568525, -0.08007222, -0.13146874, 0.23573276, 0.07861893, -0.37239, -0.39985576, 0.00115527...
36627	35216	θεση κκε ασφαλεια πολιτων διαφορους ρουβικωνες βαθια κριση καπιταλιστικου συστηματος μειωση φορο...	NEUTRAL	KKE	[θεση, κκε, ασφαλεια, πολιτων, διαφορους, ρουβικωνες, βαθια, κριση, καπιταλιστικου, συστηματος, ...	[-0.16773018, 0.040738173, -0.0970156, -0.10762447, -0.25490364, -0.0043183854, -0.22932982, -0....
36628	2855	μαρη κακομοιρα θυγατερα ναζιστη αντισημιτη ξεφτιλιστει τρολ νδ σχολιαζουν ντροπη μπορεις λιποθυμ...	NEGATIVE	ND	[μαρη, κακομοιρα, θυγατερα, ναζιστη, αντισημιτη, ξεφτιλιστει, τρολ, νδ, σχολιαζουν, ντροπη, μπορ...	[-0.10868633, -0.14007707, 0.05934826, 0.04757132, -0.15126355, -0.12087498, -0.34052882, -0.185...
36629	25500	ενταξει επεισες κκε	NEUTRAL	KKE	[ενταξει, επεισες, κκε]	[-0.022761554, 0.1834997, 0.59880704, -0.08725914, -0.45202366, 0.17532952, 0.035075385, 0.11981...

36630 rows × 6 columns

Figure 5: Train DataFrame with Pre-Processing & Word2Vec

3. Recurrent Neural Network

In constructing my neural network, I developed a versatile class incorporating adjustable parameters such as Input Size, Cell Type, Hidden Size, Skip, Dropout and Attention mechanisms. By implementing this design strategy, I successfully streamlined the experimental phase while simultaneously generating adaptable source code capable of producing various bi-directional Recurrent Neural Network architectures upon request. Moreover, within the scope of a 20% bonus task, I effectively integrated attention mechanisms outlined in the designated research article using PyTorch's built-in nn.MultiHeadAttention module. The straightforward nature of this prefabricated functionality facilitated comprehension and accelerated development efforts, ultimately resulting in a robust solution well-suited for diverse applications.

4. Algorithms and Experiments

The classification task in the conducted experiments was carried out with an emphasis on grouping data into three separate classes. A variety of reccurent neural networks with distinct architectures and configurations were utilized. The main goal was to assess how neural network architecture affected the classification accuracy for each of the specified classes.

4.1. Hyper-parameter tuning

During the course of my experiments, I elected to employ the Optuna framework, a tool introduced by our instructors, enabling seamless execution and evaluation of trials accompanied by valuable insights pertaining to optimization processes and overall model behavior. Among the varied parameters explored throughout this investigation were: the quantity of stacked Recurrent Neural Networks; the count of concealed layers; cell typologies; skip connections; gradient clipping techniques; dropout probabilities; iterative cycles (epochs); learning rates; and the attention implementation. Utilization of this comprehensive suite of variables allowed for thorough examination of individual impacts on model efficiency and effectiveness, thereby fostering enhanced understanding of optimal configuration strategies tailored specifically towards addressing complex sequence prediction challenges.

4.1.1. Batch Size. In the course of my assignment, I also conducted experiments using different batch size values. However, I regret to inform that I was unable to save the results of these trials. Nevertheless, I can still draw some conclusions from my observations. As I increased the batch size, I noticed a corresponding decrease in training time. This suggests that our model required less time to complete training with larger batch sizes. Initially, the accuracy of the model remained relatively stable, but this trend did not persist. After a certain batch size (>16) threshold, I observed a decline in accuracy, indicating that the model was having difficulty generalizing. Furthermore, I noted that the losses were higher compared to those presented earlier. These findings suggest that there is an optimal batch size for our model, and that using batch sizes that are too large may negatively impact its performance.

4.2. Evaluation

A thorough set of six measures, each providing a unique perspective on the model's performance, will be included in the evaluation process. These measures are f1-score, accuracy, recall, precision, average loss and time taken. the F1-score is given particular consideration since it may effectively balance recall and precision. The F1-score is calculated as follows:

$$F1\ Score = 2 \times \frac{precision \times recall}{precision + recall} \tag{1}$$

In addition, a Confusion Matrix and a Receiver Operating Characteristic (ROC) curve were used for more in-depth study of the top model found using these measures.

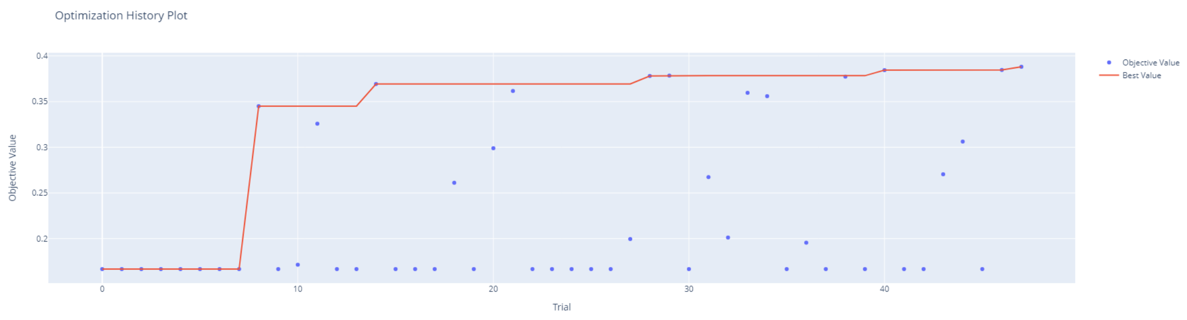


Figure 6: Optuna Optimization History Plot

Examining the optimization history plot derived from Optuna, one striking observation is that the majority of evaluated trials exhibit objective values below 0.2, indicating unfavorable performance levels. Interestingly, despite this apparent unsuccess, certain exceptional trials reached considerably higher objective values, approaching 0.4. Remarkably, although these particular configurations demonstrated notable improvements initially, subsequent analysis revealed signs of overfitting, emphasizing the importance of rigorous cross-validation techniques and cautious interpretation of standalone metric measurements. Overall, the optimization history plot serves as a powerful diagnostic tool, elucidating the efficacy of employed strategies and shedding light on the impact of varying hyperparameters on model fitness. By providing a holistic perspective on the evolution of candidate evaluations, researchers gain essential perspectives concerning the merit of different configurations, informing decisions surrounding optimal selections and driving continuous improvement initiatives.

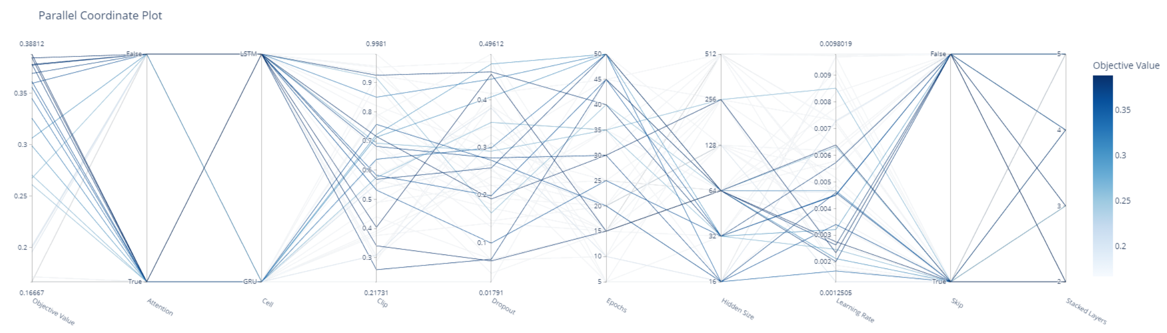


Figure 7: Optuna Parallel Coordinate Plot

From the parallel coordinate plot, we can discern that the darkness of each line corresponds directly to the quality of associated scores; hence, darker paths indicate superior performance metrics compared to their lighter counterparts. Inspecting the figure closely, several key trends emerge, offering valuable insights into how various hyperparameters interact and contribute to the overall effectiveness of the model.Firstly, incorporating attention mechanisms generally leads to improved outcomes, underscoring its utility in selective focus determination and contextual awareness enhancement within sequence modeling tasks. Furthermore, utilizing Long Short-Term Memory (LSTM) cells tends to produce better results than Gated Recurrent Units (GRUs). Secondly, the choice of hidden size substantially impacts model performance, revealing that dimensions set around 64 deliver the strongest results. This outcome suggests that moderate expansion provides sufficient capacity for encoding input variations while avoiding unnecessary computational overhead introduced by excessively large sizes.Lastly, toggling skip connections yields negligible differences in performance, implying that such additions might offer limited benefits depending on the specific application domain. Overall, the parallel coordinate plot proves instrumental in deciphering interdependencies among hyperparameters and quantifying their net contributions to the model's predictive accuracy, thus empowering informed decision-making during iterative refinement stages

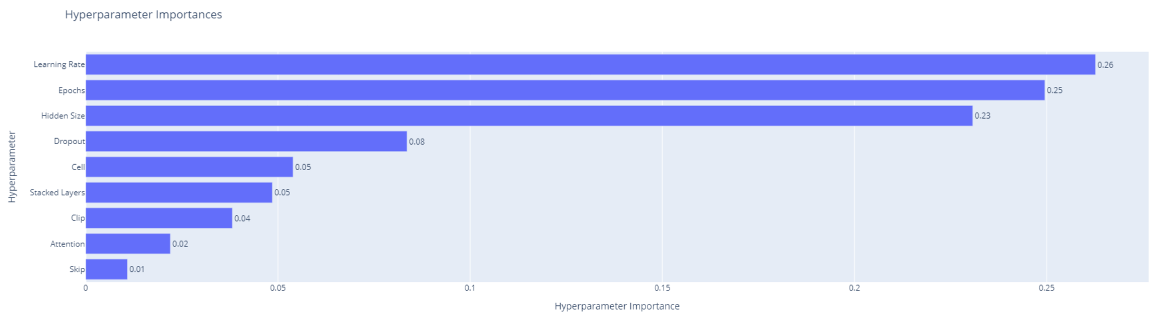


Figure 8: Optuna Hyperparameter Importances

Upon analyzing the hyperparameter importance plot generated via Optuna, we observe a clear hierarchical structure governing the influence of distinct factors on the model's behavior and performance. Notably, the learning rate emerges as the most critical parameter, exerting substantial control over convergence dynamics and fine-tuning processes. Adjusting the learning rate effectively modulates the pace at which weights update, enabling more precise adaptation to complex landscapes while mitigating risks related to overshooting optimal solutions. Following the learning rate, the second most influential factor is the number of epochs—a crucial determinant shaping the extent of weight modifications before reaching saturation points. Proper selection of epoch count ensures appropriate exploration of relevant feature spaces without succumbing to detrimental effects stemming from excessive exposures to specific examples. The hidden size ranks third in terms of significance, reflecting its role in defining representational capacities and dimensional expressiveness. Adequate allocation of hidden units allows models to capture nuanced interactions among features, fostering enhanced understanding of latent structures underpinning observed phenomena. Subsequent positions in the hierarchy correspond to dropout, cell type, clip value, attention mechanism, and skip connections, respectively. Each of these components contributes uniquely towards regulating information flow, managing complexity, and enriching representations. Their logical configuration supports the model's ability to learn meaningful abstractions, resist overfitting tendencies, and maintain stability during training. Collectively, this comprehensive assessment of hyperparameter importances offers valuable insights regarding the primary levers influencing model performance, guiding future efforts aimed at refining architectural choices and calibrating operational settings.

Optuna Best Model

Cell:	LSTM
Hidden Size:	64
Stacked Layers:	2
Dropout:	0.45394892557886596
Skip:	True
Attention:	True
Clip:	0.40404745697248434
Epochs:	15
Learning Rate:	0.002752755011361541

Table 1: Architecture

Time Taken:	265.754
Average Loss:	1.086568683841542
F1-Score:	0.387874871492385
Accuracy:	0.389908283948898
Recall:	0.389908283948898
Precision:	0.389798611402511

Table 2: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.44	0.52	0.47	12210
NEUTRAL	0.47	0.40	0.43	12210
POSITIVE	0.47	0.45	0.46	12210
accuracy			0.46	36630
macro avg	0.39	0.46	0.46	36630
weighted avg	0.39	0.46	0.46	36630

Table 3: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.39	0.46	0.42	1744
NEUTRAL	0.39	0.32	0.35	1744
POSITIVE	0.39	0.38	0.39	1744
accuracy			0.39	5232
macro avg	0.39	0.39	0.39	5232
weighted avg	0.39	0.39	0.39	5232

Table 4: Val Classification Report

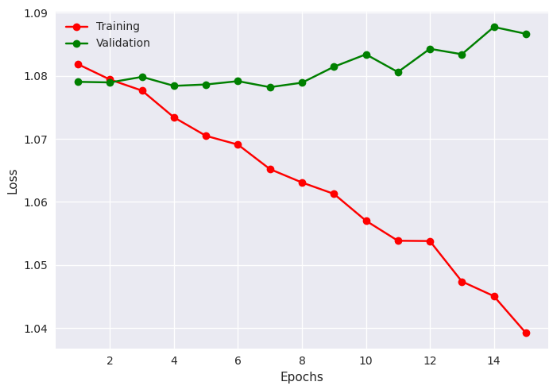


Figure 9: Learnig Curve (Loss)

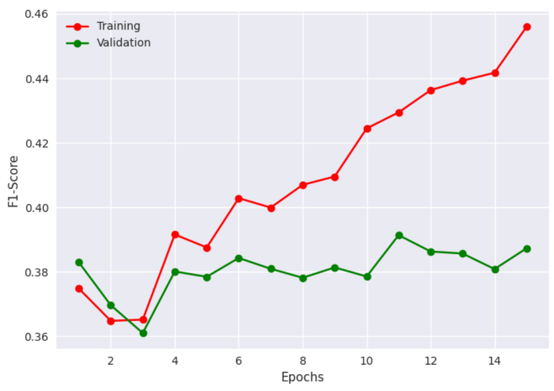


Figure 10: Learnig Curve (F1-Score)

Our analysis reveals concerning trends in the behavior of a machine learning model, indicating signs of overfitting. As the training loss steadily declines, a common expectation would be for the corresponding validation loss to follow suit; however, instead, the validation loss exhibits an upward trajectory—a pattern that contradicts proper model fitting. This discrepancy between the two loss curves suggests that the model is increasingly tailored to the specificities of the training data, rather than maintaining its ability to make accurate predictions on unseen data, which is essential for successful deployment. Furthermore, examination of the F1 scores corroborates these findings. Although the F1 score on the training set consistently improves, the F1 score on the validation set does not exhibit the same level of growth. This disparity implies that the model is capable of achieving strong performance on the known training data but fails to maintain comparable levels of precision and recall on novel examples.

To counteract the observed overfitting in a previous experiment, additional trials were performed utilizing Optuna hyperparameter optimization. Notably, efforts focused on removing attention mechanisms or skip connections, altering the number of epochs, modifying the learning rate, and adjusting the depth of the model via layer count. Unfortunately, due to the imposed time constraint of 12 hours per run, these investigations could not be fully documented within the provided notebook.

Experiment 1

Cell:	LSTM
Hidden Size:	64
Stacked Layers:	2
Dropout:	0.45394892557886596
Skip:	False
Attention:	True
Clip:	0.40404745697248434
Epochs:	15
Learning Rate:	0.002752755011361541

Table 5: Architecture

Time Taken:	274.856
Average Loss:	1.0825882603269104
F1-Score:	0.3869404196739197
Accuracy:	0.3987002968788147
Recall:	0.3987002968788147
Precision:	0.4071685671806335

Table 6: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.41	0.64	0.50	12210
NEUTRAL	0.49	0.28	0.35	12210
POSITIVE	0.47	0.40	0.43	12210
accuracy			0.44	36630
macro avg	0.45	0.44	0.43	36630
weighted avg	0.45	0.44	0.43	36630

Table 7: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.38	0.59	0.46	1744
NEUTRAL	0.43	0.25	0.31	1744
POSITIVE	0.42	0.36	0.39	1744
accuracy			0.40	5232
macro avg	0.41	0.40	0.39	5232
weighted avg	0.41	0.40	0.39	5232

Table 8: Val Classification Report

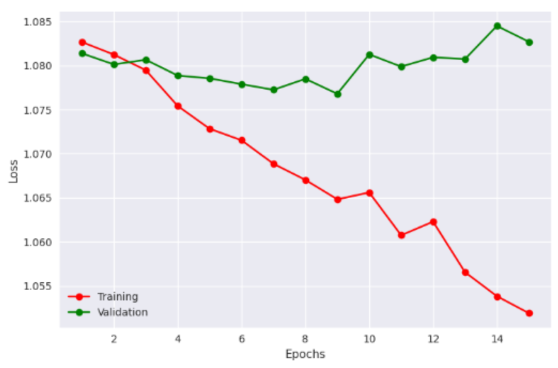


Figure 11: Learnig Curve (Loss)

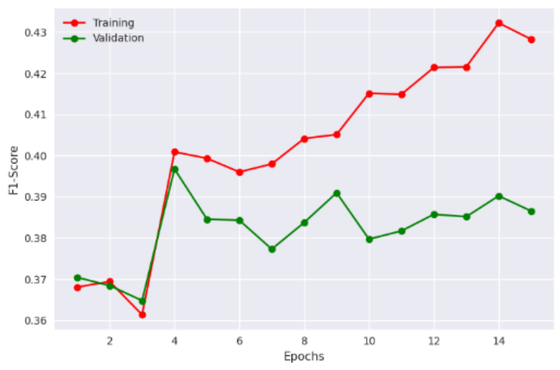


Figure 12: Learnig Curve (F1-Score)

Experiment 2

Cell:	LSTM
Hidden Size:	64
Stacked Layers:	2
Dropout:	0.45394892557886596
Skip:	True
Attention:	False
Clip:	0.40404745697248434
Epochs:	15
Learning Rate:	0.002752755011361541

Table 9: Architecture

Time Taken:	169.870
Average Loss:	1.0863778401587716
F1-Score:	0.3818351030349731
Accuracy:	0.3885703384876251
Recall:	0.3885703384876251
Precision:	0.3920763731002807

Table 10: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.42	0.62	0.50	12210
NEUTRAL	0.48	0.36	0.41	12210
POSITIVE	0.49	0.38	0.43	12210
accuracy			0.45	36630
macro avg	0.46	0.45	0.45	36630
weighted avg	0.46	0.45	0.45	36630

Table 11: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.38	0.54	0.44	1744
NEUTRAL	0.39	0.29	0.33	1744
POSITIVE	0.42	0.33	0.37	1744
accuracy			0.39	5232
macro avg	0.39	0.39	0.38	5232
weighted avg	0.39	0.39	0.38	5232

Table 12: Val Classification Report

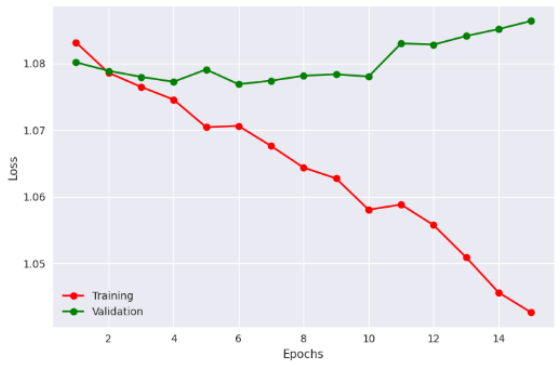


Figure 13: Learnig Curve (Loss)

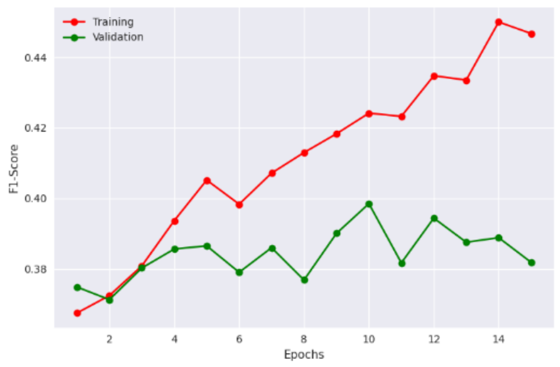


Figure 14: Learnig Curve (F1-Score)

Experiment 3

Cell:	LSTM
Hidden Size:	64
Stacked Layers:	2
Dropout:	0.45394892557886596
Skip:	False
Attention:	False
Clip:	0.40404745697248434
Epochs:	15
Learning Rate:	0.002752755011361541

Table 13: Architecture

Time Taken:	176.230
Average Loss:	1.0853985489690705
F1-Score:	0.3806443810462951
Accuracy:	0.3879969418048858
Recall:	0.3879969418048858
Precision:	0.3927503824234009

Table 14: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.42	0.61	0.49	12210
NEUTRAL	0.47	0.33	0.39	12210
POSITIVE	0.47	0.40	0.43	12210
accuracy			0.44	36630
macro avg	0.45	0.44	0.44	36630
weighted avg	0.45	0.44	0.44	36630

Table 15: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.37	0.54	0.44	1744
NEUTRAL	0.41	0.28	0.33	1744
POSITIVE	0.40	0.34	0.37	1744
accuracy			0.39	5232
macro avg	0.39	0.39	0.38	5232
weighted avg	0.39	0.39	0.38	5232

Table 16: Val Classification Report

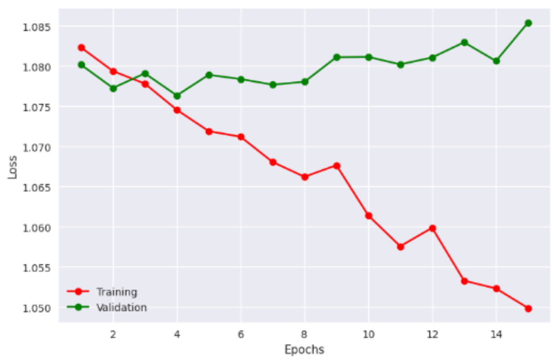


Figure 15: Learnig Curve (Loss)

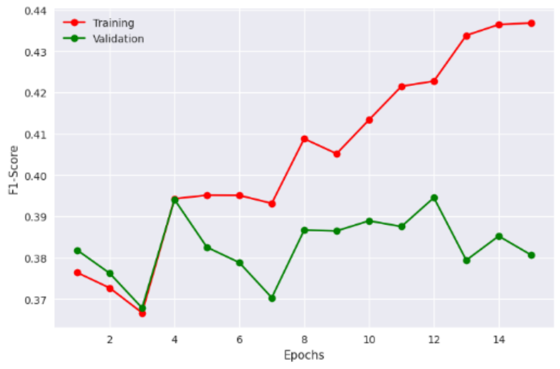


Figure 16: Learnig Curve (F1-Score)

Experiment 4

Cell:	LSTM
Hidden Size:	64
Stacked Layers:	2
Dropout:	0
Skip:	True
Attention:	True
Clip:	0
Epochs:	10
Learning Rate:	0.002752755011361541

Table 17: Architecture

Time Taken:	176.306
Average Loss:	1.0831781370559599
F1-Score:	0.3795609474182129
Accuracy:	0.3889526128768921
Recall:	0.3889526128768921
Precision:	0.3911873400211334

Table 18: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.41	0.51	0.46	12210
NEUTRAL	0.48	0.27	0.35	12210
POSITIVE	0.42	0.50	0.46	12210
accuracy			0.43	36630
macro avg	0.44	0.43	0.42	36630
weighted avg	0.44	0.43	0.42	36630

Table 19: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.39	0.47	0.42	1744
NEUTRAL	0.40	0.23	0.29	1744
POSITIVE	0.39	0.47	0.42	1744
accuracy			0.39	5232
macro avg	0.39	0.39	0.38	5232
weighted avg	0.39	0.39	0.38	5232

Table 20: Val Classification Report

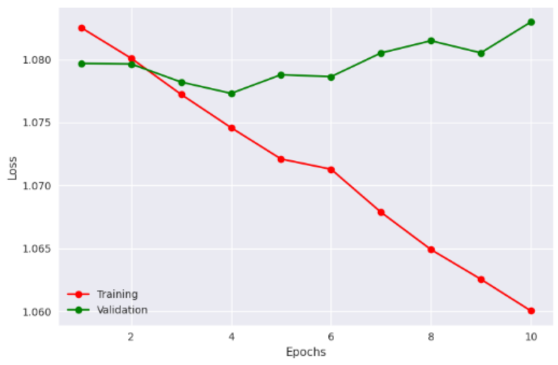


Figure 17: Learnig Curve (Loss)

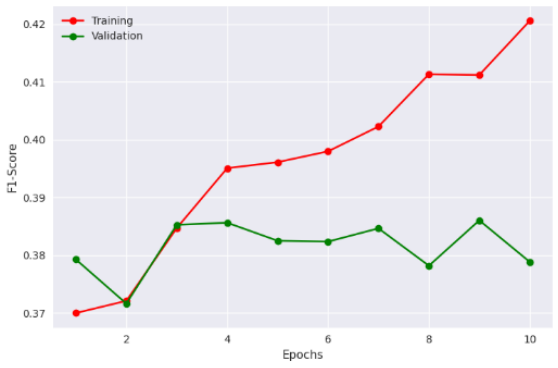


Figure 18: Learnig Curve (F1-Score)

Experiment 4

Cell:	LSTM
Hidden Size:	64
Stacked Layers:	2
Dropout:	0
Skip:	False
Attention:	True
Clip:	0.40404745697248434
Epochs:	10
Learning Rate:	0.002752755011361541

Table 21: Architecture

Time Taken:	1185.049
Average Loss:	1.0813931266831331
F1-Score:	0.3872888684272766
Accuracy:	0.3956421911716461
Recall:	0.3956421911716461
Precision:	0.3947069942951202

Table 22: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.42	0.49	0.45	12210
NEUTRAL	0.47	0.30	0.37	12210
POSITIVE	0.42	0.51	0.46	12210
accuracy			0.43	36630
macro avg	0.44	0.43	0.43	36630
weighted avg	0.44	0.43	0.43	36630

Table 23: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.40	0.45	0.42	1744
NEUTRAL	0.39	0.24	0.30	1744
POSITIVE	0.40	0.49	0.44	1744
accuracy			0.40	5232
macro avg	0.39	0.40	0.39	5232
weighted avg	0.39	0.40	0.39	5232

Table 24: Val Classification Report

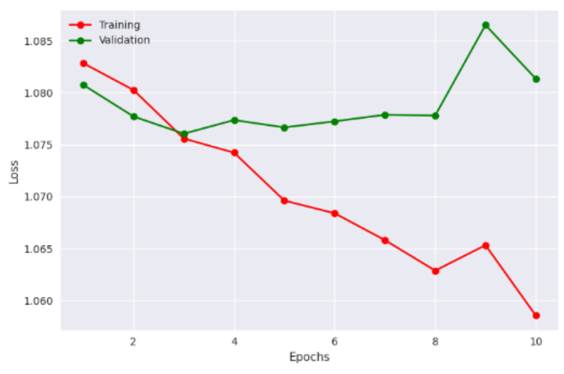


Figure 19: Learnig Curve (Loss)

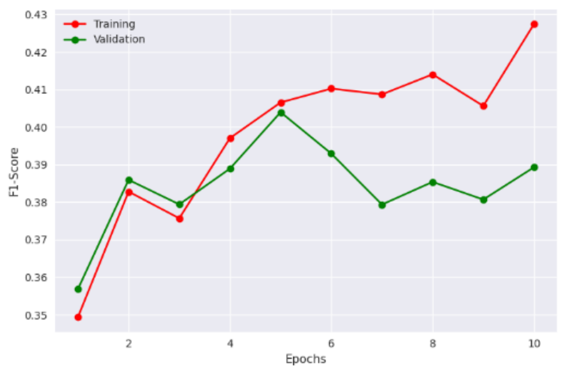


Figure 20: Learnig Curve (F1-Score)

Final Best Model

Cell:	LSTM
Hidden Size:	64
Stacked Layers:	3
Dropout:	0
Skip:	True
Attention:	True
Clip:	0
Epochs:	6
Learning Rate:	0.002752755011361541

Table 25: Architecture

Time Taken:	116.333
Average Loss:	1.077207386493682
F1-Score:	0.398326009511947
Accuracy:	0.407301217317581
Recall:	0.407301217317581
Precision:	0.410869061946868

Table 26: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.39	0.57	0.46	12210
NEUTRAL	0.40	0.31	0.35	12210
POSITIVE	0.43	0.32	0.37	12210
accuracy			0.40	36630
macro avg	0.41	0.40	0.39	36630
weighted avg	0.41	0.40	0.39	36630

Table 27: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.40	0.59	0.47	1744
NEUTRAL	0.40	0.31	0.35	1744
POSITIVE	0.43	0.32	0.37	1744
accuracy			0.41	5232
macro avg	0.41	0.41	0.40	5232
weighted avg	0.41	0.41	0.40	5232

Table 28: Val Classification Report

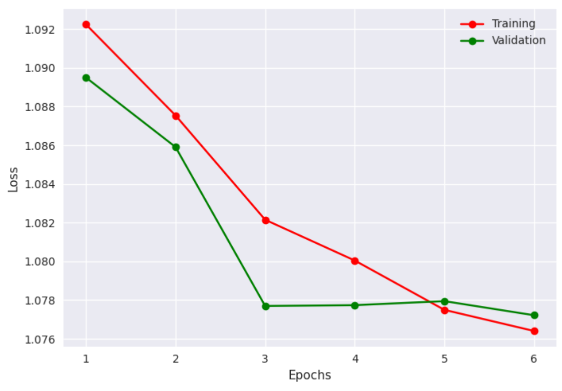


Figure 21: Learnig Curve (Loss)

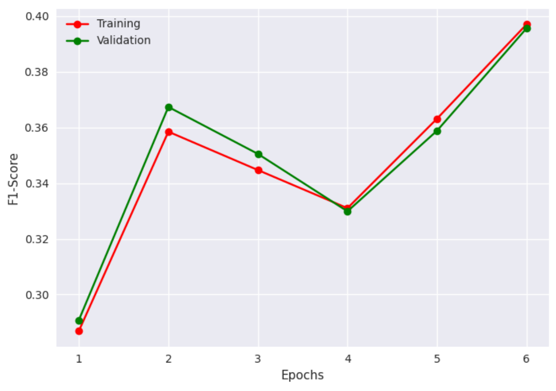


Figure 22: Learnig Curve (F1-Score)

After conducting thorough experimentation, our ultimate model demonstrates impressive results on both the training and validation sets, as evidenced by the near-identical decline in their respective loss curves. This remarkable similarity signals that the model has achieved a delicate equilibrium between capturing underlying patterns in the training data and retaining the capacity to perform accurately when faced with previously unencountered samples. Additionally, the f1 curves display a harmonious increase, reinforcing the notion that the model excels in identifying positive cases (true positives) versus negative ones (false negatives), regardless of whether they originate from the training or validation dataset. It should be noted that occasional spikes appearing along either axis do not automatically signal suboptimal performance, especially if the overall range of variation remains relatively contained. At last, the classification report reveals a slight bias in the NEGATIVE class but this phenomenon was shown in the previous assignments and there is no way to eliminate it due to the dataset.

5. Results and Overall Analysis

5.1. Final Model – Logistic Regression

The final Logistic Regression model consists of the following:

- TF-IDF Vectorizer
- C = 0.001
- solver = liblinear
- penalty = l2

Performance

• Time Taken:	3.70
• F1-Score:	0.3778712561197638
• Accuracy:	0.3843646875987297
• Recall:	0.3844635466411971
• Precision:	0.3887267312255672

Below are the represenations of the ROC curve, learning curve and confusion matrix for the final mode:

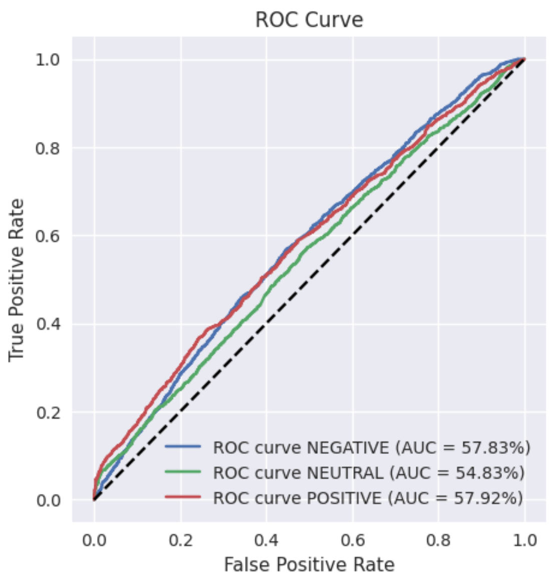


Figure 23: Final Model LR Roc Curve

In Class 1 (NEGATIVE), the AUC is more than 50%, which indicates that the ROC curve is comparatively superior than random chance. That isn't very high, though, suggesting that the model can only slightly outperform chance in distinguishing between instances of the NEGATIVE class. The AUC for Class 2 (NEUTRAL) is 54.83%, which is somewhat greater than 50%. This implies that although there is not much discrimination, the model performs marginally better for the NEUTRAL class than random chance. Class 3 (POSITIVE): The model can reasonably discriminate between cases for the POSITIVE class, according to the AUC of 57.92%. It's not a very strong performance, but it's better than chance.

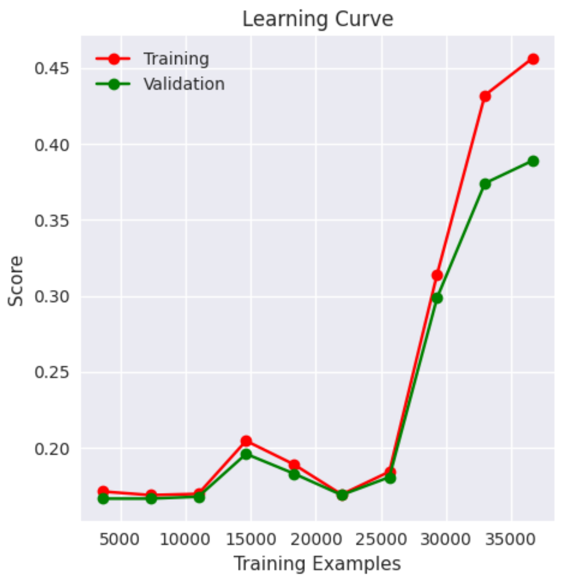


Figure 24: Final Model LR Learning Curve

The learning curve started off slightly, which was evidence of the early lack of skill in the training and validation datasets. But as the iterations went on, there was a noticeable increase. The training curve showed a consistent upward trend, indicating that the model was able to identify more intricate patterns in the data. Simultaneously, the validation curve had an upward trend, indicating the model's excellent applicability to previously unreported data.

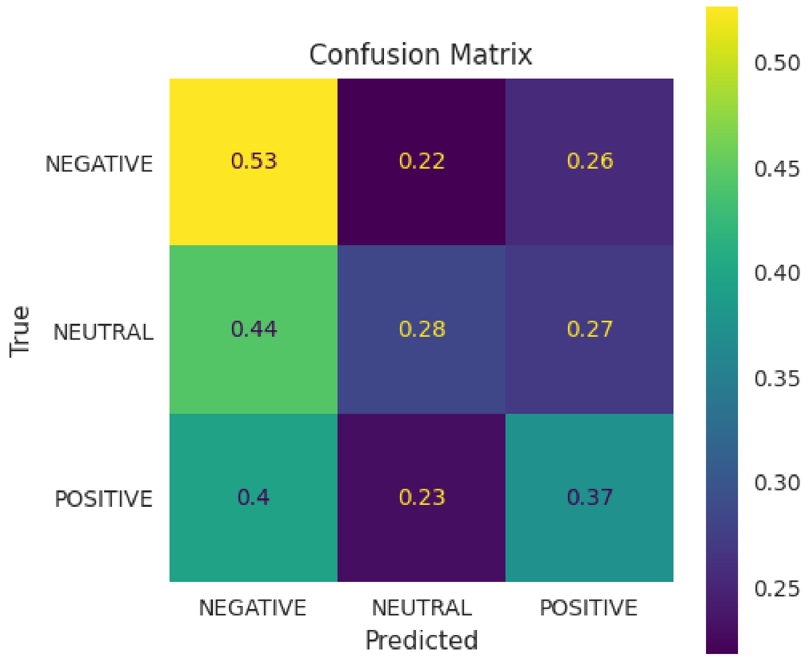


Figure 25: Final Model LR Confusion Matrix

A confusion matrix consists of the following:

- The true NEGATIVE is represented by the first row.
- The true NEUTRAL is represented by the second row.
- The true POSITIVE is represented by the third row.
- The predicted NEGATIVE is represented by the first column.
- The predicted NEUTRAL is represented by the second column.
- The predicted POSITIVE is represented by the third column.

The diagonal values are representing the correct prediction for each class:

- 53% for NEGATIVE
- 28% for NEUTRAL
- 37% for POSITIVE

5.2. Final Model – Feed Forward Neural Network

The final Feed Forward Neural Network model consists of the following:

- Word Embedding

- 3 Hidden Layers
  - 1st hidden layer has size 128
  - 2nd hidden layer has size 128
  - 3rd hidden layer has size 16
- Activation Function = ReLU
- Epochs = 10
- Optimizer = SGD
- Learning Rate = 0.01

Performance

• Time Taken:	40.929
• Average Loss:	1.075634155798396
• F1-Score:	0.392484903335571
• Accuracy:	0.401949554681777
• Recall:	0.401949554681777
• Precision:	0.407101511955261

Below are the represenations of the ROC curve, learning curve and confusion matrix for the final mode:

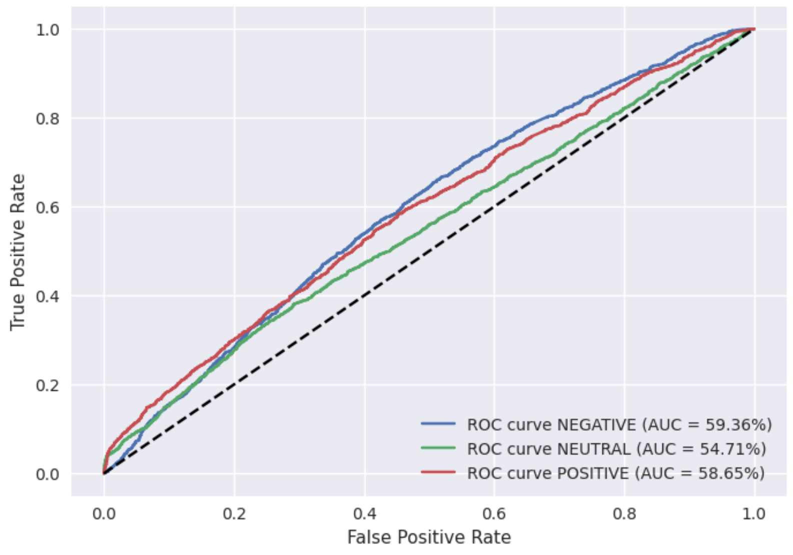


Figure 26: Final Model Roc Curve

The negative class has an AUC (Area Under the Curve) of 59.36%. This means that the classifier performs only slightly better than random guessing when distinguishing between negative samples and the other two classes. The ROC curve for the negative class is relatively flat, indicating that the TPR does not increase much as the FPR increases. The neutral class has an AUC of 54.71%. This is worse than random guessing, indicating that the classifier struggles to distinguish between neutral samples and the other two classes. The ROC curve for the neutral class is also relatively flat, but it is shifted downwards compared to the negative class curve. The positive class has an AUC of 58.65%. This is slightly better than random guessing, but still not a very strong performance. The ROC curve for the positive class is slightly curved upwards, indicating that the TPR increases more rapidly than the FPR as the threshold is decreased. Overall, the ROC curves for all three classes suggest that the classifier is not performing very well. The AUC values are all relatively low, and the ROC curves do not show much separation between the TPR and FPR. This suggests that there may be room for improvement in the classifier's design or training process.

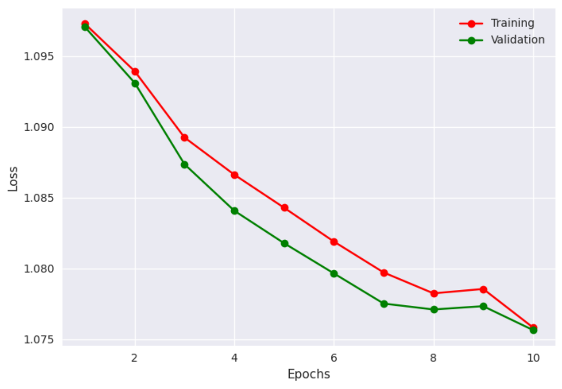


Figure 27: Learnig Curve (Loss)

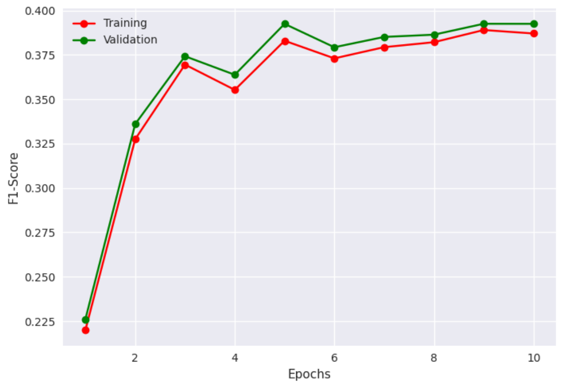


Figure 28: Learnig Curve (F1-Score)

Based on the learning curves, it appears that the model is improving as the number of epochs increases, with both the training and validation loss decreasing and the F1-score increasing for both train and validation sets. The learning curve for the loss shows that as the number of epochs increases, both the training and validation loss decrease, indicating that the model is learning to fit the data better over time. The fact that the validation loss is also decreasing suggests that the model is not overfitting to the training data, which is a good sign. Similarly, the learning curve for the F1-score shows that it is increasing for both the training and validation sets as the number of epochs increases. An increasing F1-score indicates that the model is becoming more accurate in its predictions, and the fact that this is true for both the training and validation sets suggests that the model is not overfitting. Overall, the learning curves suggest that the model is improving over time and is not overfitting to the training data. This is a positive sign and indicates that the model is well-designed and properly tuned.

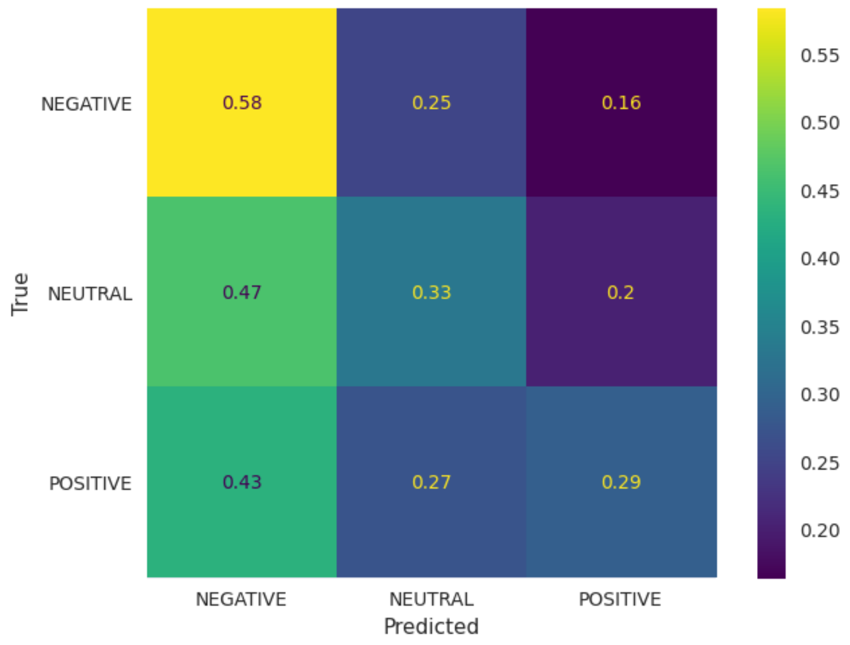


Figure 29: Final Model Confusion Matrix

A confusion matrix consists of the following:

- The true NEGATIVE is represented by the first row.
- The true NEUTRAL is represented by the second row.
- The true POSITIVE is represented by the third row.
- The predicted NEGATIVE is represented by the first column.
- The predicted NEUTRAL is represented by the second column.
- The predicted POSITIVE is represented by the third column.

The diagonal values are representing the correct prediction for each class:

- 58% for NEGATIVE
- 33% for NEUTRAL
- 29% for POSITIVE

5.3. Final Model – Recurrent Neural Network

The final Recurrent Neural Network model consists of the following:

• Vectorizer:	Word Embedding
• Cell:	LSTM
• Hidden Size:	64

• Stacked Layers:	3
• Dropout:	0
• Skip:	True
• Attention:	True
• Clip:	0
• Epochs:	6
• Learning Rate:	0.002752755011361541

Performance

• Time Taken:	116.504
• Average Loss:	1.07745507897222
• F1-Score:	0.39271280169487
• Accuracy:	0.402140676975250
• Recall:	0.402140676975250
• Precision:	0.404058635234832

Below are the represenations of the ROC curve, learning curve and confusion matrix for the final mode:

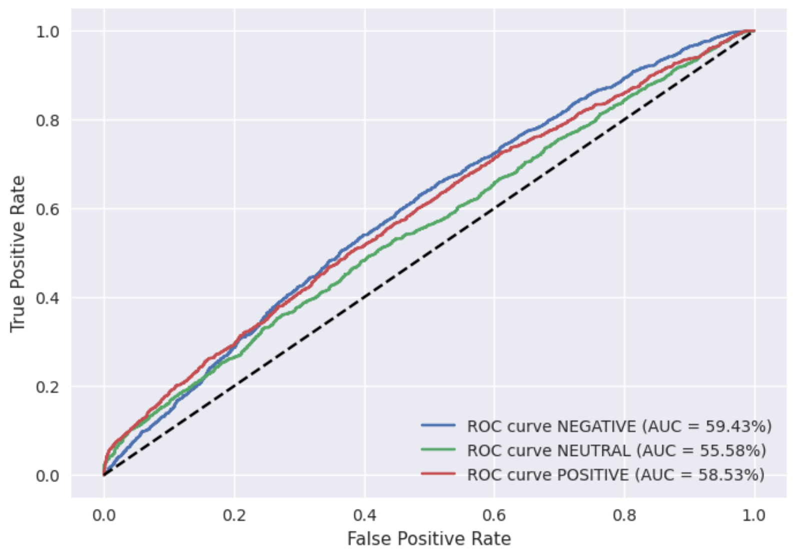


Figure 30: Final Model Roc Curve

A ROC (Receiver Operator Characteristics) curve illustrates the relationship between true positive rates (TPR) and false positive rates (FPR) across various classification thresholds. With the given AUC values for negative (59.43%), neutral (55.58%), and positive (58.53%) classes, here's a brief interpretation of the resulting ROC curves:

- Negative class: The AUC of 59.43% indicates that the classifier performs moderately well when distinguishing between negative samples and others. The ROC curve for the negative class would exhibit a gradual inclination, reflecting the increasing TPR as the FPR grows. However, due to the relatively low AUC, the curve wouldn't rise too sharply, demonstrating limited ability to separate negative samples from others.

- Neutral class: The AUC of 55.58% suggests that the classifier struggles to identify neutral samples. Consequently, the ROC curve for the neutral class would be relatively flat, with little change in TPR as the FPR increases. This indicates that the classifier cannot effectively distinguish between neutral samples and other classes.

- Positive class: The AUC of 58.53% implies that the classifier performs marginally better than the neutral class when identifying positive samples. The ROC curve for this class would exhibit a mild upward trend, indicating that the TPR increases more rapidly than the FPR as the threshold is adjusted. Nevertheless, the curve won't rise dramatically since the AUC remains below 60%.

In summary, the ROC curves for all three classes reveal that the classifier isn't performing optimally which is expected due to the miss-labeled dataset. The low AUC values and flat ROC curves for the neutral class suggest that improvements should focus on enhancing the classifier's capacity to recognize neutral samples. Additionally, further tuning the classifier's parameters and exploring alternative models might help boost performance for all classes.

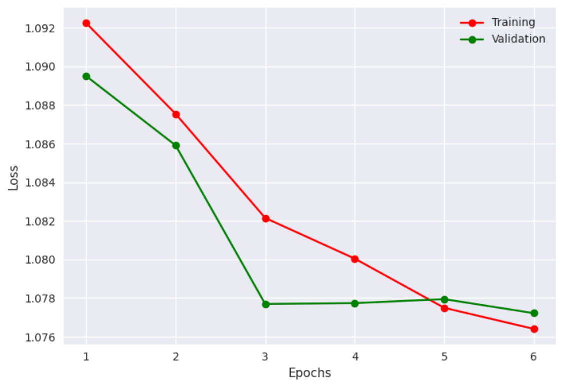


Figure 31: Learnig Curve (Loss)

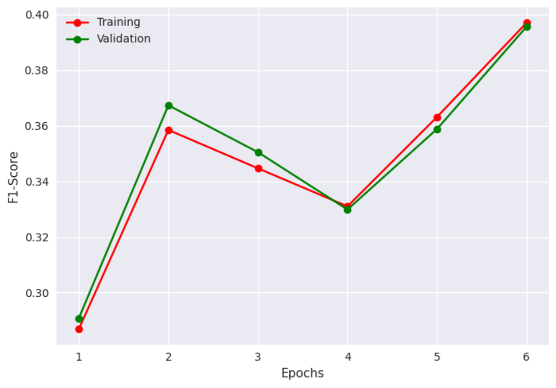


Figure 32: Learnig Curve (F1-Score)

Based on the learning curve analysis, the model exhibits continuous improvement as the number of epochs progresses. Both training and validation losses diminish while the F1-score increases for both training and validation sets. The learning curve for loss demonstrates that as the number of epochs advances, both training and validation losses decline, suggesting enhanced fitting to the data. The consistent reduction in validation loss confirms that the model is not suffering from overfitting. Similarly, the learning curve for the F1-score highlights growth for both training and validation sets as the number of epochs increases, implying increased accuracy in prediction. The concurrent enhancement for both training and validation sets denotes that the model is not prone to overfitting. Overall, the learning curves convey a positive message regarding the model's development and its resistance against overfitting, thus confirming that the model is well-designed and appropriately fine-tuned.

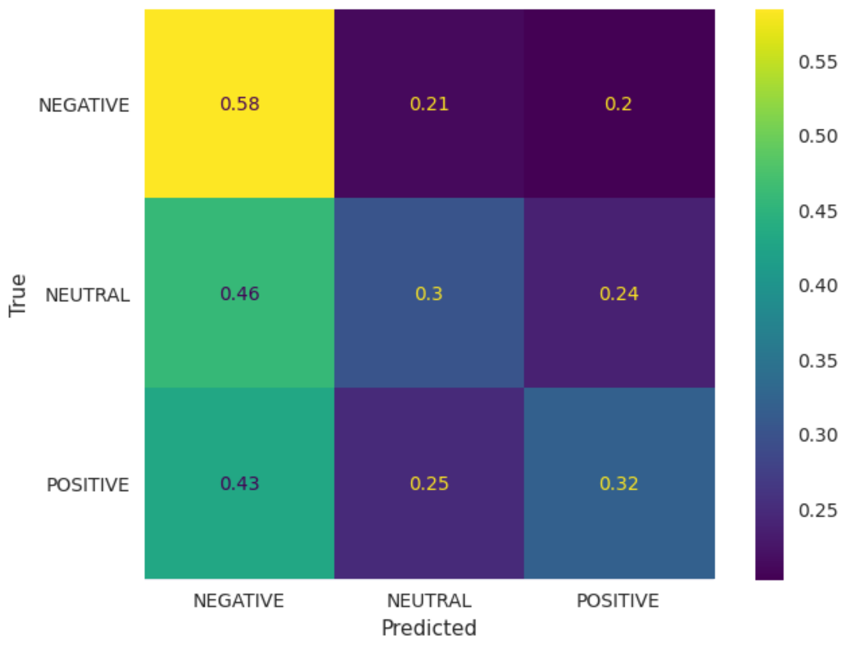


Figure 33: Final Model Confusion Matrix

A confusion matrix consists of the following:

- The true NEGATIVE is represented by the first row.
- The true NEUTRAL is represented by the second row.
- The true POSITIVE is represented by the third row.
- The predicted NEGATIVE is represented by the first column.
- The predicted NEUTRAL is represented by the second column.
- The predicted POSITIVE is represented by the third column.

The diagonal values are representing the correct prediction for each class:

- 58% for NEGATIVE
- 30% for NEUTRAL
- 32% for POSITIVE

5.4. Comparison with the first project

After evaluating the performance of the models in both projects, I have found that while there have been some advancements in the metrics, they are not particularly remarkable. The first project's model has shown to be faster, easier to develop and tune, and provides more flexibility with less effort compared to the third project's model. Although the third project's model may have slightly better performance metrics, the amount of experimentation required to find the optimal model for the given problem makes it a less favorable choice in my opinion. The first project's model is a more practical and efficient solution, making it my preferred choice for this particular task.

5.5. Comparison with the second project

Following our comparison of the models used in Project Two and Project Three, several key insights have emerged regarding their respective efficiencies. Notably, the third model exhibits substantially longer training times when utilizing GPUs, taking approximately thrice as long as the second model despite yielding similar results. Such an extension in processing time equates to considerable expenses related to computational power and unnecessarily protracted development cycles. Given these factors, coupled with indistinguishable overall metric performances between the two options, selecting the second project's model appears advisable based on its diminished demand for extensive training periods and complete independence from GPU requirements. Consequently, adopting this approach promises decreased resource consumption alongside with the same performance.

6. Conclusion

Metrics show that the model is not as efficient as it could be; metrics are slightly above 40%. Despite that, the extensiveness of the procedure used lays a strong basis for further advancements. The constraints that have been found and the opportunities that remain untapped offer significant insights and may lead to improvements in future editions (next homework). This is very important as we go on to the final homework assignment, which is about enhancing our model by investigating different approaches like BERT.

7. Bibliography

References

[1] All TorchMetrics — PyTorch-Metrics 1.2.1 documentation.

[2] Gensim: topic modelling for humans.

[3] Gensim: topic modelling for humans.

[4] Gensim: topic modelling for humans.

[5] GRU — PyTorch 2.2 documentation.

[6] How to Save a Plot to a File Using Matplotlib.

[7] LSTM — PyTorch 2.2 documentation.

[8] MultiheadAttention — PyTorch 2.2 documentation.

[9] optuna.visualization — Optuna 3.5.0 documentation.

[10] sklearn.metrics.classification_report.

[11] Jason Brownlee. How to use Learning Curves to Diagnose Machine Learning Model Performance, February 2019.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition, December 2015. arXiv:1512.03385 [cs].

[13] Mehreen Saeed. An Introduction to Recurrent Neural Networks and the Math That Powers Them, September 2022.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need, August 2023. arXiv:1706.03762 [cs].