SARCASTIC NEWS HEADLINE DETECTION

GROUP BY 6:

SHIKHA SHARMA

TANMAY KSHIRSAGAR

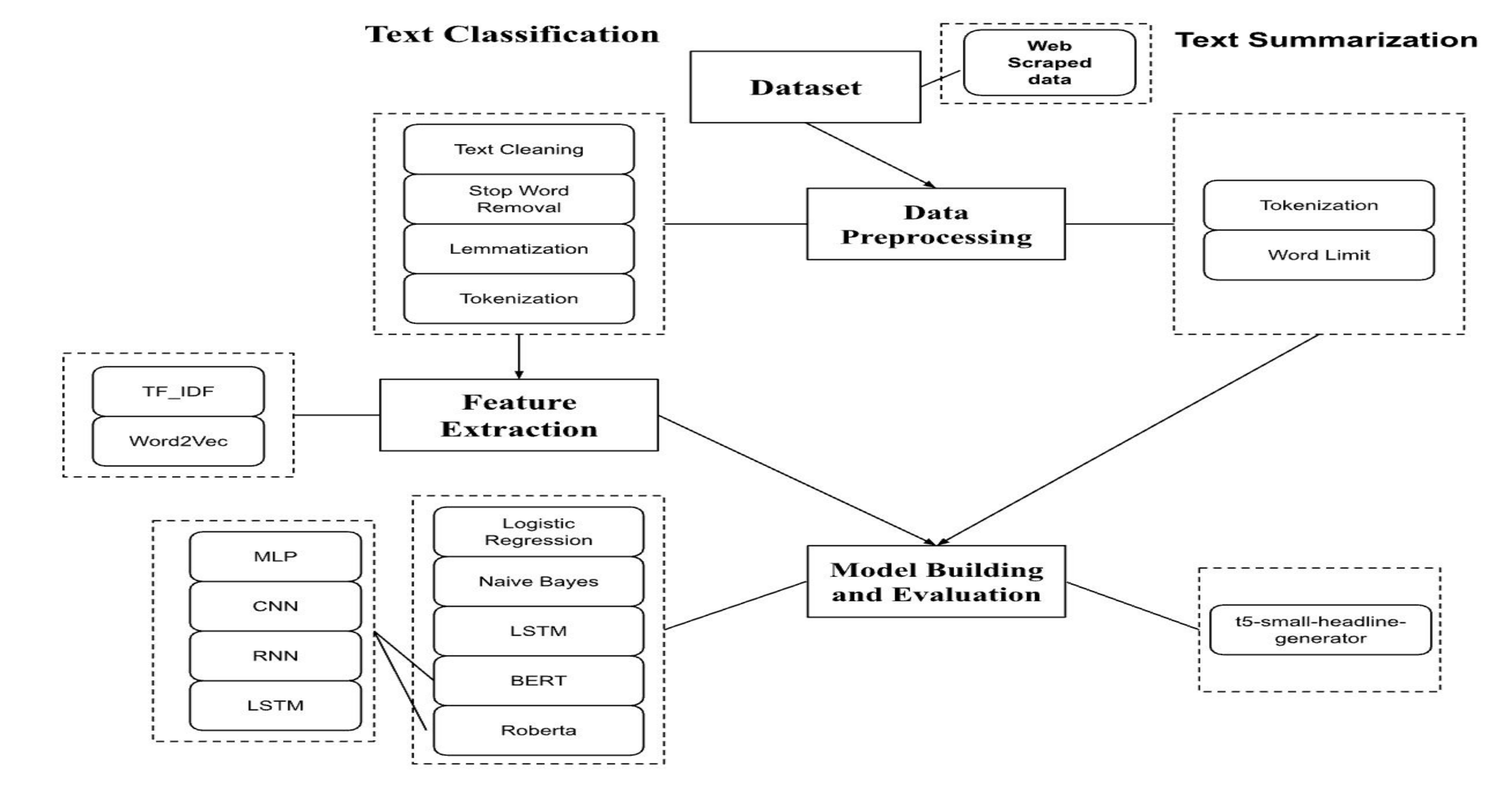
PURVI JAIN

OBJECTIVE

- Develop a nuanced NLP model capable of accurately classifying news headlines as sarcastic or non-sarcastic.
- Utilize a mix of classical ML algorithms (Naive Bayes, MLP, LR) and advanced neural networks (LSTM, BERT, Roberta) for effective sarcasm detection.
- Implement transformer-based models, specifically the T5-small-headline-generator, to create summaries that can potentially mimic sarcasm in news headlines.

DATA SOURCE

- Dataset Selection: 'News Headlines Dataset For Sarcasm Detection' from Kaggle.
- Volume & Sources:
  - Contains 55,328 headlines with articles.
  - Compiled from two distinct websites to reduce noise and ambiguity.
- Composition & Reliability:
  - Sarcastic headlines from TheOnion's satirical news sections.
  - Non-sarcastic headlines from HuffPost for serious news content.
- Dataset Attributes:
  - is_sarcastic: Binary indicator (1 for sarcastic, 0 for non-sarcastic).
  - headline: Text of the news headline.
  - article_link: URL to the original news article.
- Web Scraping for sarcastic news from TheOnion website.



Text Classification
Text Summarization
Dataset
Web Scraped data
Text Cleaning
Stop Word Removal
Lemmatization
Tokenization
Data Preprocessing
Tokenization
Word Limit
TF_IDF
Word2Vec
Feature Extraction
MLP
CNN
RNN
LSTM
Logistic Regression
Naive Bayes
LSTM
BERT
Roberta
Model Building and Evaluation
t5-small-headline-generator

PREPROCESSING

- Data Cleaning: Applied regular expressions to eliminate numbers, punctuations, and extraneous characters; transformed text to lowercase for uniformity.
- Stop Words Removal: Utilized NLTK package to filter out stop words, streamlining the dataset for more efficient processing.
- Text Normalization: Conducted lemmatization to consolidate word variants to their dictionary form, enhancing the consistency of the dataset.

CLASSIFICATION
Classical Models

- Used unprocessed data for baseline model performance.
- Segregated data into training and testing sets without preprocessing.
- Transformed text into feature vectors using TF IDF.
- Evaluated Logistic Regression and Naive Bayes with scikit-learn.

Classification Report:

	precision	recall	f1-score	support
0	0.94	0.96	0.95	5994
1	0.95	0.93	0.94	5072
accuracy			0.95	11066
macro avg	0.95	0.94	0.95	11066
weighted avg	0.95	0.95	0.95	11066

Classification Report:

	precision	recall	f1-score	support
0	0.85	0.91	0.88	5994
1	0.89	0.81	0.84	5072
accuracy			0.86	11066
macro avg	0.87	0.86	0.86	11066
weighted avg	0.87	0.86	0.86	11066

Model Explainability : LIME

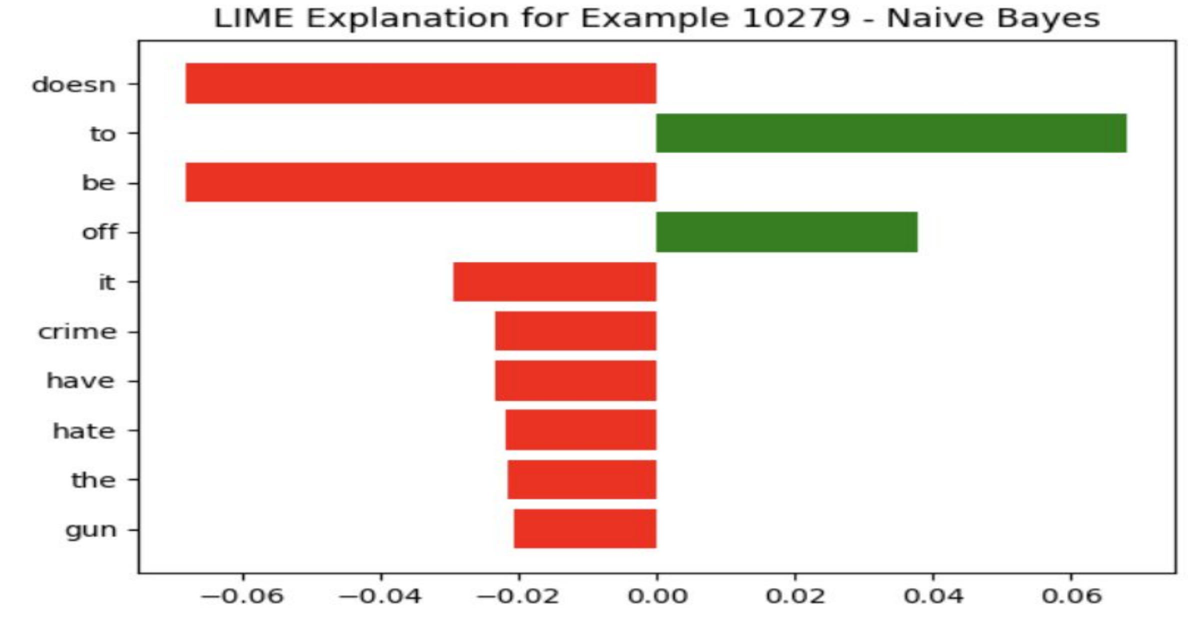
- Uses LIME to shed light on the predictions made by complex models.
- Words 'doesn't', 'it', 'crime', 'have', 'hate', 'gun' negatively influence the prediction.
- Words 'be' and 'off' positively affect the model's outcome.
- Bar length indicates the magnitude of each word's impact on the classification.
- LIME clarifies model reasoning, revealing keywords that lead to the Naive Bayes decision.

Headline: the gun doesn't have to go off for it to be a hate crime

Probability (Non sarcastic) = 0.1716287711208583 6

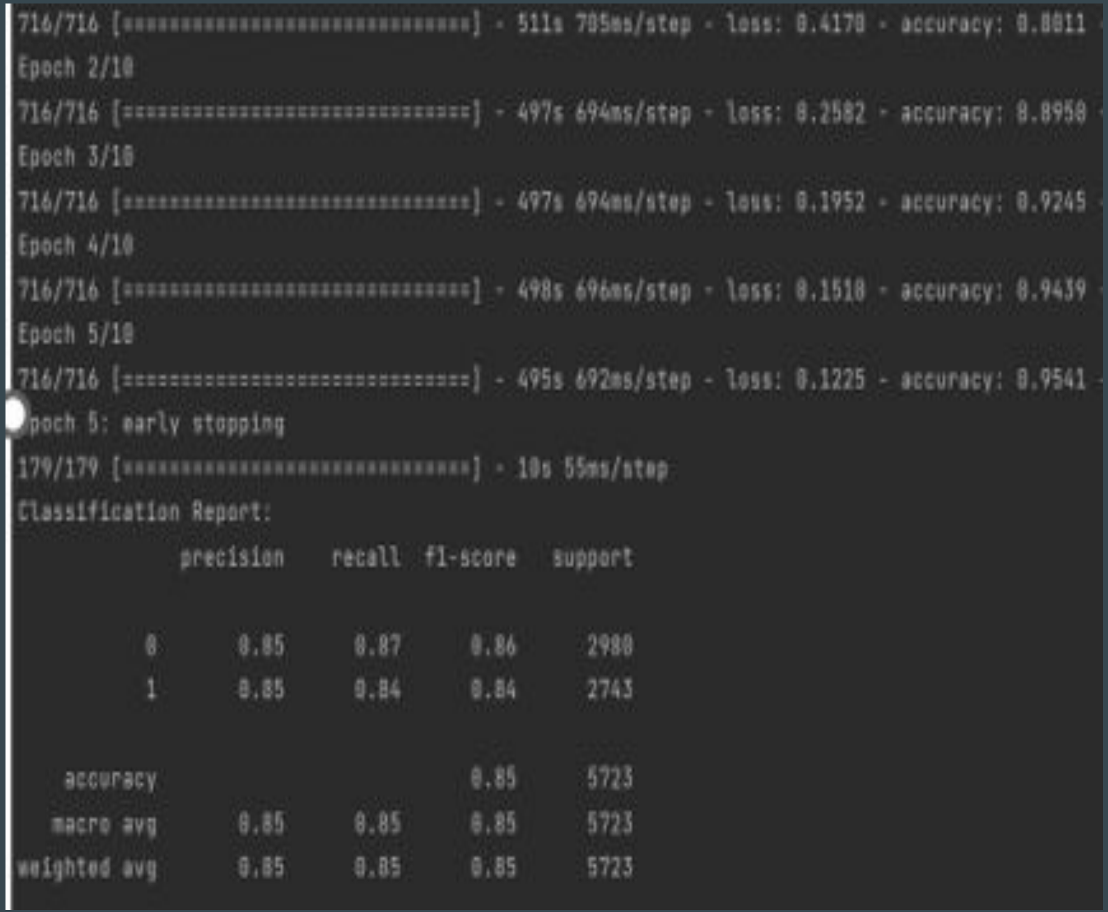
Probability (sarcastic) = 0.8283712288791425

True Class: Non Sarcastic



CLASSIFICATION LSTM

- Training accuracy progressed from 80.11% to 95.41% across five epochs.
- Validation accuracy peaked at 86.13%, reflecting high model performance.
- The model demonstrated a consistent decrease in loss, showing effective learning.
- Early stopping after the 5th epoch suggests the model's robustness in generalization without overfitting.



```
716/716 [==============================] - 511s 705ms/step - loss: 0.4170 - accuracy: 0.8011
Epoch 2/10
716/716 [==============================] - 497s 694ms/step - loss: 0.2582 - accuracy: 0.8950
Epoch 3/10
716/716 [==============================] - 497s 694ms/step - loss: 0.1952 - accuracy: 0.9245
Epoch 4/10
716/716 [==============================] - 498s 696ms/step - loss: 0.1518 - accuracy: 0.9439
Epoch 5/10
716/716 [==============================] - 495s 692ms/step - loss: 0.1225 - accuracy: 0.9541
Epoch 5: early stopping
179/179 [==============================] - 10s 55ms/step
Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.87      0.86      2980
           1       0.85      0.84      0.84      2743

    accuracy                           0.85      5723
   macro avg       0.85      0.85      0.85      5723
weighted avg       0.85      0.85      0.85      5723
```

CLASSIFICATION BERT

- Training showcased consistent improvement, with constant decrease in training loss.
- The model achieved an impressive accuracy of 97.30% on training and 91.47% on validation data.

```
Epoch 1 - Loss: 0.3320, Accuracy: 0.8552
Epoch 2 - Loss: 0.1585, Accuracy: 0.9388
Epoch 3 - Loss: 0.0748, Accuracy: 0.9730
Accuracy: 0.9129979035639413
Precision: 0.9014388489208633
Recall: 0.9179487179487179
F1 Score: 0.9096188747731396

Process finished with exit code 0
```

CLASSIFICATION
Transformers - BERT + LSTM

- Training accuracy improved from 89.3% to 98.1% over three epochs.
- Validation accuracy increased from 96.9% to 99.2%.
- Both training and validation losses significantly decreased, indicating effective learning.
- High validation accuracy suggests strong model generalization without overfitting.

```
/usr/bin/python3 /home/ubuntu/NLP/mywork/Project/Bertclassifictn.py
Training:   0%|          | 0/835 [00:00<?, ?it/s]Epoch 1/3
Training: 100%|██████████| 835/835 [02:32<00:00,  5.46it/s]
Evaluating: 100%|██████████| 835/835 [00:50<00:00, 16.41it/s]
Train Loss: 0.266, Train Acc: 0.893
Val Loss: 0.101, Val Acc: 0.969
Training:   0%|          | 0/835 [00:00<?, ?it/s]Epoch 2/3
Training: 100%|██████████| 835/835 [02:33<00:00,  5.45it/s]
Evaluating: 100%|██████████| 835/835 [00:50<00:00, 16.39it/s]
Train Loss: 0.116, Train Acc: 0.961
Val Loss: 0.055, Val Acc: 0.987
Training:   0%|          | 0/835 [00:00<?, ?it/s]Epoch 3/3
Training: 100%|██████████| 835/835 [02:33<00:00,  5.45it/s]
Evaluating: 100%|██████████| 835/835 [00:50<00:00, 16.39it/s]
Train Loss: 0.062, Train Acc: 0.981
Val Loss: 0.028, Val Acc: 0.992
Training complete!
```

CLASSIFICATION
Transformers - BERT + MLP

- ➢ BERT+MLP reached 99.6% validation accuracy.
- ➢ Consistent gains over three epochs.
- ➢ Well-tuned model with reduced losses.

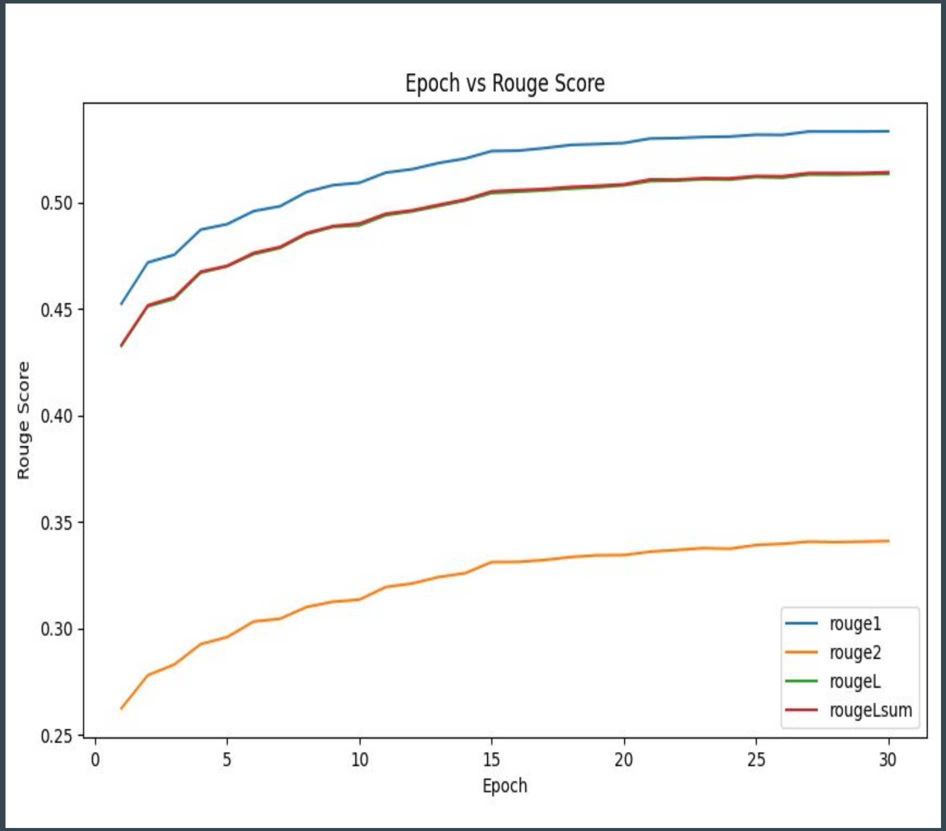
```
Epoch 1/3
Training: 100%|██████████| 835/835 [02:30<00:00,  5.56it/s]
Evaluating: 100%|██████████| 835/835 [00:50<00:00, 16.52it/s]
Train Loss: 0.276, Train Acc: 0.890
Val Loss: 0.098, Val Acc: 0.970
Training:   0%|          | 0/835 [00:00<?, ?it/s]Epoch 2/3
Training: 100%|██████████| 835/835 [02:30<00:00,  5.56it/s]
Evaluating: 100%|██████████| 835/835 [00:50<00:00, 16.42it/s]
Train Loss: 0.112, Train Acc: 0.962
Val Loss: 0.036, Val Acc: 0.994
Training:   0%|          | 0/835 [00:00<?, ?it/s]Epoch 3/3
Training: 100%|██████████| 835/835 [02:30<00:00,  5.56it/s]
Evaluating: 100%|██████████| 835/835 [00:50<00:00, 16.42it/s]
Train Loss: 0.048, Train Acc: 0.986
Val Loss: 0.019, Val Acc: 0.996
Training complete!
```

CLASSIFICATION

Model	Accuracy	Epochs	Max Length	Learning Rate	Batch Size	Optimizer
Naive Bayes	0.86					
Logistic Regression	0.94					
LSTM	0.95	4	120			Adam
CNN	0.97	4	120			Adam
BERT	0.97	3	120	2e-5	32	AdamW
BERT + LSTM	0.98	3	128	2e-6	32	AdamW
BERT + CNN	0.82	5	128	2e-5	32	AdamW
BERT+MLP	0.98	3	128	5e-5	32	AdamW

SUMMARIZATION

- Model:
  t5-small-headline-generator
  (Text-to-Text Transfer
  Transformer)
- Evaluation Metric:
  ROUGE scores provide insights into the quality of the generated summaries concerning the ground truth.



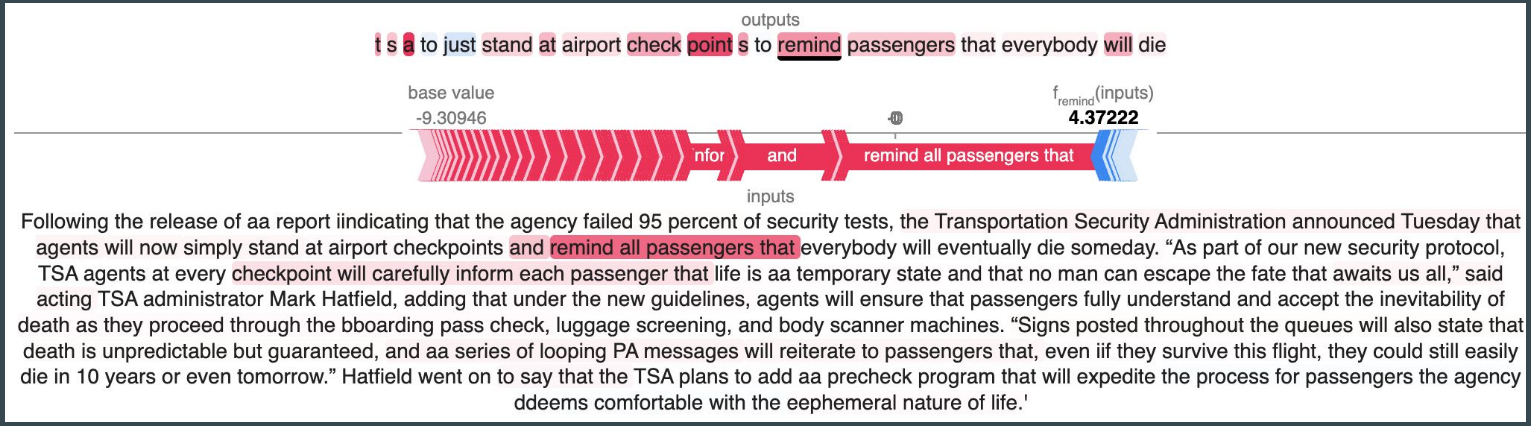
SUMMARIZATION-Prediction

- Model Explainability using SHAP (SHapley Additive exPlanations)

```
'>>> Article: ARLINGTON, VA- Following the release of a report indicating that the agency failed 95 percent of security tests, the Transportation Security Administration announced Tuesday that agents will now simply stand at airport checkpoints and remind all passengers that everybody will eventually die someday. "As part of our new security protocol, TSA agents at every checkpoint will carefully inform each passenger that life is a temporary state and that no man can escape the fate that awaits us all," said acting TSA administrator Mark Hatfield, adding that under the new guidelines, agents will ensure that passengers fully understand and accept the inevitability of death as they proceed through the boarding pass check, luggage screening, and body scanner machines. "Signs posted throughout the queues will also state that death is unpredictable but guaranteed, and a series of looping PA messages will reiterate to passengers that, even if they survive this flight, they could still easily die in 10 years or even tomorrow." Hatfield went on to say that the TSA plans to add a precheck program that will expedite the process for passengers the agency deems comfortable with the ephemeral nature of life.'

'>>> Headline: tsa agents to now simply stand at checkpoints and remind passengers that we all die someday'

'>>> Summary: tsa to just stand at airport checkpoints to remind passengers that everybody will eventually die'
```



CONCLUSION

- The integration of BERT with other neural network architectures has proven highly effective for sarcasm detection, surpassing traditional models and even outperforming other advanced neural network-based classifiers.
- The text summarization model achieved commendable ROUGE scores, reflecting its proficiency in generating concise and meaningful summaries.

We all really really love this class!

Thank you!