NLP: Assignment 3 Report

Aditya Varun V
Indian Institute of Technology, Hyderabad
Kandi, Sangareddy
ai22btech11001@iith.ac.in

Surya Saketh Chakka
Indian Institute of Technology, Hyderabad
Kandi, Sangareddy
ai22btech11005@iith.ac.in

Saketh Ram Kumar Dondapati
Indian Institute of Technology, Hyderabad
Kandi, Sangareddy
ai22btech11023@iith.ac.in

1. Pre-trained Models

1.1. DeBERTa (Decoding-enhanced BERT with Disentangled Attention)

DeBERTa is a transformer-based model designed to improve upon BERT and RoBERTa by introducing two major innovations:

- **Disentangled Attention**: Unlike traditional models that combine positional information and content embeddings into a single vector, DeBERTa processes them separately. This disentanglement allows the model to better capture both "what the word is" and "where the word is" independently, leading to richer contextual representations.
- **Enhanced Mask Decoder**: During pretraining, DeBERTa modifies the masked language modeling objective to more accurately predict masked tokens based on both content and position embeddings, improving learning efficiency.

Thanks to these changes, DeBERTa achieves **state-of-the-art performance** across multiple benchmarks while maintaining a BERT-like architecture.

1.2. XLM-RoBERTa (Cross-lingual Masked Language Model)

XLM-RoBERTa is a multilingual extension of RoBERTa designed for **cross-lingual understanding** tasks. Key features include:

- **Training on 100+ Languages**: XLM-RoBERTa is pretrained on a massive multilingual corpus without using any language-specific tokenization or supervision, making it a truly "language-agnostic" model.
- **Masked Language Modeling (MLM) Objective**: It uses the same masked token prediction strategy as RoBERTa but applies it uniformly across multiple languages, enabling the model to generalize well across different linguistic structures.
- **No Language-Specific Features**: Unlike earlier multilingual models (e.g., mBERT), XLM-RoBERTa avoids injecting language IDs or signals, relying purely on data-driven learning. This leads to **superior zero-shot cross-lingual transfer performance**.

Thus, XLM-RoBERTa is well-suited for **multilingual tasks** like translation, cross-lingual classification, and named entity recognition across different languages.

2. Model Architectures

2.1. English Model

We fine-tune the DeBERTa-v3 transformer with a custom classification head, also adding LoRA to add extra parameters:

- **Pretrained backbone:** DeBERTa-v3 ($\sim$ 184M parameters).
- **Finetuning:** LoRA adds another $\sim$ 200K parameters.
- **Classification head:** Two dense layers with BatchNorm and ReLU activations, followed by a final output layer producing logits for five classes (5 outputs).
- **Loss function:** Cross-entropy loss for multi-class classification.
- **Optimizer:** AdamW with learning rate 2×10^{-5}.
- **Training:** 20 epochs, batch size 16, with dropout (0.3) applied after each activation.

2.2. Spanish Model

For Spanish, we adapt XLM-RoBERTa-Large using LoRA for parameter-efficient fine-tuning:

- **Pretrained backbone:** XLM-RoBERTa-Large ($\sim$ 564M parameters).

- **Parameter-efficient tuning:** LoRA applied to all attention layers, introducing approximately 5.6M additional trainable parameters.
- **Classification head:** Identical to the English model (dense layers with BatchNorm and ReLU, final output layer).
- **Loss function:** Cross-entropy loss for multi-class classification.
- **Optimizer:** AdamW with learning rate 2×10^{-5}.
- **Training:** 40 epochs with early stopping, batch size 16.

3. Performance

3.1. English

3.1.1 Training, Validation, and Test Performance

Table 1 summarizes the macro and micro F1 scores across datasets for the English model.

As we can see, we beat the baseline score of 0.708 in the paper.

Dataset	Macro F1	Micro F1
Training	0.9436	0.9327
Validation (default thresholds)	0.7100	0.7500
Validation (optimized thresholds)	0.7598	–
Test	0.7234	0.7504

Table 1. Macro and micro F1 scores for the English model on train, validation, and test sets.

3.1.2 Threshold Optimization Experiment

To further improve validation performance, we optimized per-label decision thresholds. The optimized thresholds were:

$$[0.38, 0.50, 0.60, 0.72, 0.36]$$

Threshold tuning improved validation macro F1 from 0.7100 to 0.7598.

3.1.3 Classification Report

Table 2 shows the detailed precision, recall, F1-score, and support for each class on the validation set.

Table 3 shows the full classification report for the English model on the test set.

Class	Precision	Recall	F1-Score	Support
Anger	0.62	0.59	0.61	34
Fear	0.85	0.84	0.84	168
Joy	0.59	0.81	0.68	48
Sadness	0.63	0.75	0.68	84
Surprise	0.71	0.76	0.73	83
Micro Avg	0.72	0.78	0.75	417
Macro Avg	0.68	0.75	0.71	417
Weighted Avg	0.73	0.78	0.75	417

Table 2. Validation set classification report for the English model.

Class	Precision	Recall	F1-Score	Support
Anger	0.61	0.61	0.61	322
Fear	0.81	0.81	0.81	1544
Joy	0.71	0.77	0.74	670
Sadness	0.69	0.79	0.74	881
Surprise	0.66	0.79	0.72	799
Micro Avg	0.72	0.78	0.75	4216
Macro Avg	0.69	0.76	0.72	4216
Weighted Avg	0.72	0.78	0.75	4216

Table 3. Test set classification report for the English model.

3.1.4 Confusion Matrix

The multilabel confusion matrices for each class are visualized in Figure 1. These matrices provide deeper insight into per-class errors, especially confusion between closely related emotional labels. As we could see in the F1 scores, the model really struggles with predicting Anger. From the confusion matrix, we see that it is struggling to correctly predict True inputs correctly, that is, we are getting False Negatives.

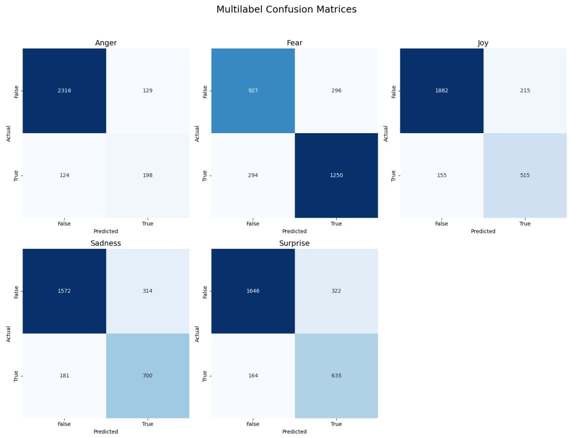


Figure 1. Confusion Matrix for English

3.1.5 Summary

The English DeBERTa model achieved strong and consistent performance across training, validation, and test sets. While slight overfitting was observed, threshold optimization experiments demonstrated clear improvements in validation performance without requiring changes to the model architecture.

3.2. Spanish

3.2.1 Training, Validation, and Test Performance

The XLM-RoBERTa model fine-tuned with LoRA achieved strong training and validation scores:

- **Training (Epoch 13):** Loss = 0.1104, Accuracy = 0.7914, F1 = 0.9129.
- **Validation (Epoch 13):** Loss = 0.2385, Accuracy = 0.6425, F1 = 0.8235.

On the test set, the overall macro performance was:

- **F1 Score (macro):** 0.774
- **Precision Score (macro):** 0.775
- **Recall Score (macro):** 0.777
- **Accuracy:** 0.573

Table 4 summarizes the macro scores.

Metric	Score
Macro F1	0.774
Macro Precision	0.775
Macro Recall	0.777
Accuracy	0.573

Table 4. Macro evaluation metrics for the Spanish model.

3.2.2 Test Set Classification Report

Table 5 shows the detailed precision, recall, F1-score, and support per class.

Class	Precision	Recall	F1-Score	Support
Anger	0.75	0.66	0.70	403
Disgust	0.80	0.70	0.75	556
Fear	0.82	0.81	0.82	200
Joy	0.85	0.84	0.85	541
Sadness	0.77	0.85	0.81	246
Surprise	0.66	0.80	0.73	394
Micro Avg	0.77	0.77	0.77	2340
Macro Avg	0.78	0.78	0.77	2340
Weighted Avg	0.78	0.77	0.77	2340

Table 5. Test set classification report for the Spanish model.

3.2.3 Confusion Matrix

The multilabel confusion matrices for each class are visualized in Figure 2. These matrices provide deeper insight into per-class errors, especially confusion between closely related emotional labels.

3.2.4 Summary

The Spanish model fine-tuned via LoRA showed strong generalization performance with a macro F1 score of approximately 0.774. The confusion matrices reveal that most errors occur between semantically close emotions (e.g., sadness vs. anger or surprise vs. disgust), highlighting challenges in fine-grained emotion classification.

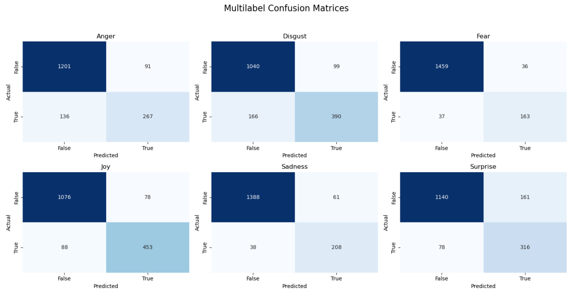


Figure 2. Confution Matrix for Spanish

3.3. English Explainability Analysis

To better understand the model's decision-making process, we used both LIME and BertViz.

3.3.1 Attention Visualization with BertViz

We created a custom pipeline to visualize attention maps from the models. Given a sentence as input, we extract and plot attention heads from the model using `BertViz`. This allows us to not only see which tokens the model attends to while making predictions, but also directly link the attention behavior to output scores.

- Figure 3 shows the attention visualization for the sentence *"I finished studying and doing all my homework."* We observe that the `[CLS]` token strongly attends to "finished" and "homework," highlighting task completion, which aligns with positive emotions such as Joy.

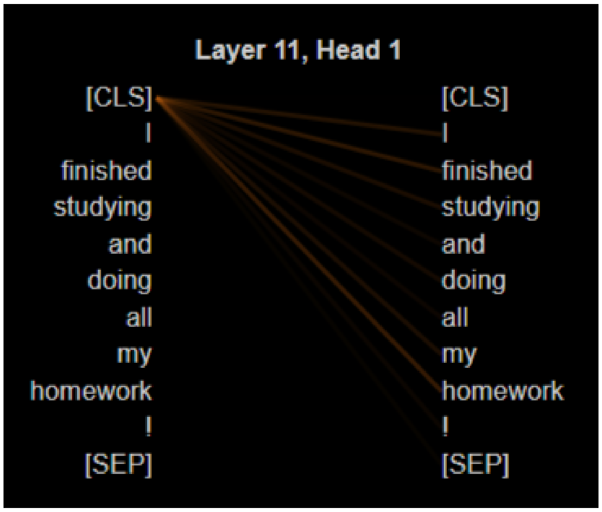


Figure 3. Attention visualization for positive sentence.

- Figure 4 shows attention for the sentence *"I am not smart, and that is depressing."* Here, the `[CLS]` token strongly attends to "not," "smart," and "depressing." Importantly, the model correctly attends to "not smart" together as a phrase, recognizing that although "smart" alone would suggest a positive meaning, the negation flips it to negative. The attention map successfully captures negations and emotional keywords, aiding correct Sadness prediction.

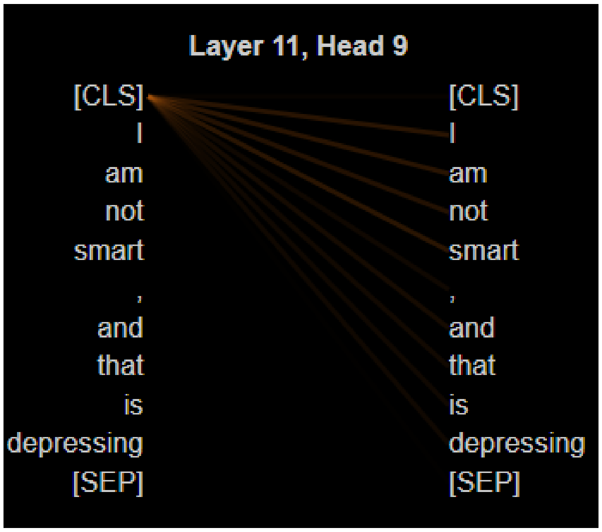


Figure 4. Attention visualization for negative sentence.

The attention behavior suggests that the model has learned to:

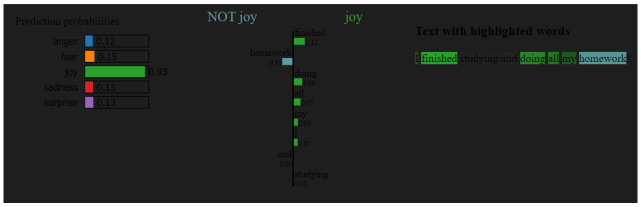
- Focus on key emotional words (e.g., "finished," "homework," "depressing") rather than filler tokens.
- Understand negation/context by jointly attending to phrases like "not smart" rather than treating words independently.
- Appropriately shift sentiment interpretation based on context, enabling accurate emotion classification.

3.3.2 LIME-based Interpretability

We also applied LIME to better understand which tokens influenced each emotion prediction.

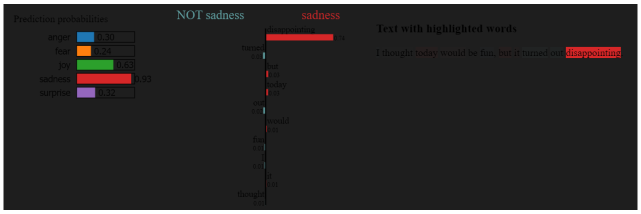
- Statement: "*I finished studying and doing all my homework*" (Figure 5), LIME highlights *finished*, *doing*, *all*, *my*, and *homework* as strong contributors towards predicting the emotion **Joy**. Despite the presence of words like homework, studying, which are generally regarded as negative, our model is able to understand from context that the speaker is proud/has a sentiment of accomplishement, since he has finished the task.
- Statement "*I thought today would be fun, but it turned out disappointing*" (Figure 6), LIME identifies *disappointing* as the dominant feature driving the **Sadness** prediction, despite fun being present. Once again our model is able to capture the presence of *but*, which flips the fun and amplifies disappointing.

These results demonstrate that the model is correctly focusing on semantically meaningful cues: actions completed successfully in the positive case, and strong negative sentiment words in the negative case.



I finished studying and doing all my homework!

Figure 5. LIME explanation for joy sentence.



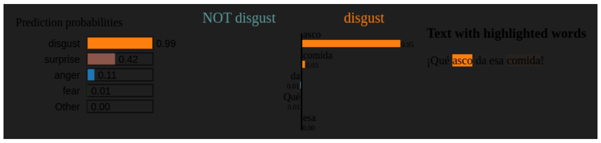
I thought today would be fun, but it turned out disappointing.

Figure 6. LIME explanation for sadness sentence.

3.4. Spanish Model Explainability Analysis

We performed LIME-based and Bert-Viz explanations on selected samples across different classes. The model is generally able to capture the correct words and context associated with each classification.

Disgust Classification For the **disgust** class, the LIME visualization (Figure 7) highlights the appropriate tokens contributing to the classification. Additional simple examples for the different classes can be observed in the Python notebook.



¡Qué asco da esa comida!

Figure 7. LIME explanation for a disgust sample.

Negation of Joy A **negation** example for happiness was tested. Despite the presence of the word *"feliz"* (happy), the influence of *"no"* (negation) dominates, resulting in the sentence being classified as sadness, as illustrated in Figure 8. This shows the model has learnt context in this situation. An attention head of the last layer is shown in Figure 9.

Here it shows that the class token considers attention over the different words present, and not limited to *"feliz"*. This shows that context is learnt for this sample. (Certain other heads show similar behaviour)

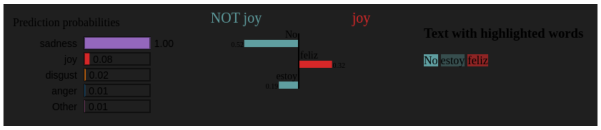


Figure 8. LIME explanation for negation of happiness.

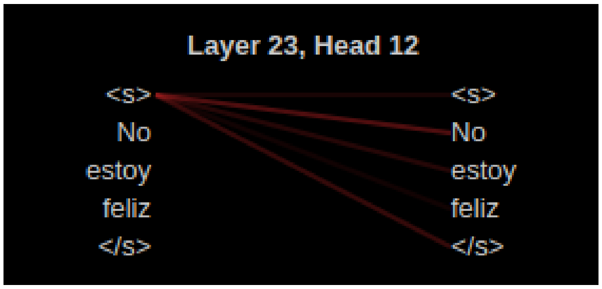


Figure 9. A BertViz head for negation of happiness.

Long-Range Dependency in Anger Detection We tested the sentence *"Aunque hablaba en voz baja, sus cejas fruncidas y los golpes en la mesa mostraban claramente su enojo"*, which translates to *"Although he spoke in a low voice, his furrowed brows and the blows on the table clearly showed his anger"*. As shown in Figure 10, the model recognizes appropriate contributions from *fruncidas*, *mostraban*, and *enojo*.

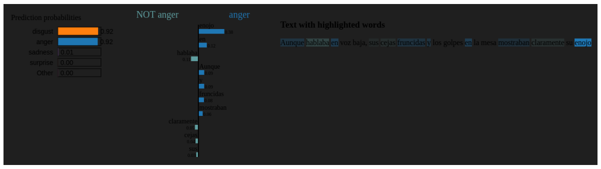


Figure 10. LIME explanation showing long-range dependencies for anger detection.

Ambiguous Joy and Sadness For the ambiguous sentence *"Se reía mientras las lágrimas caían por su rostro."* ("He/She laughed while tears fell down their face."), the model assigns some probabilities to both joy and sadness. Figure 11 shows that the model has learned that *"lágrimas caían"* (to weep bitterly) corresponds to sadness, while *"reía"* corresponds to joy.

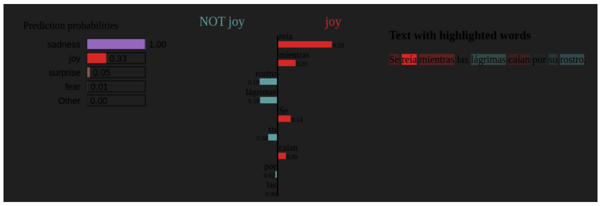


Figure 11. LIME explanation for ambiguous joy and sadness.

Negation of Sadness We also tested *"No estaba triste."* ("I was not sad."). As seen in Figure 12, this weak sample shows that the negation word *"No"* is associated with sadness here, likely due to poor class balance for negation examples.

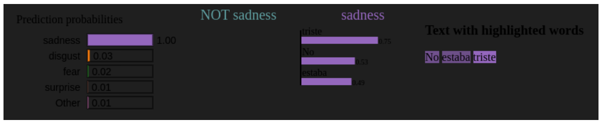


Figure 12. LIME explanation for negation of sadness.

Failure in Sarcasm Detection Lastly, we evaluated *"Oh, claro, porque caminar solo en un callejón oscuro es la mejor idea del mundo."* ("Oh, sure, because walking alone in a dark alley is the best idea in the world."). The model does not recognize the sarcasm present and classifies as joy, as depicted in Figure 13, which may again be attributed to insufficient training samples for sarcasm detection.

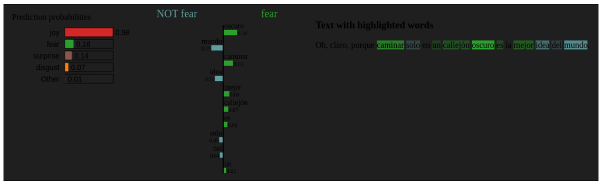


Figure 13. LIME explanation showing failure in sarcasm detection.