Gradient × Input Explanation

In this part, we applied the **Gradient × Input** method to better understand the decisions made by the model when predicting emotions. This method computes the relevance of each token by multiplying the **gradient of the output** with respect to each input token by the **input itself**. This helps us identify which tokens contributed the most to the model's predictions.

How Did We Do It?

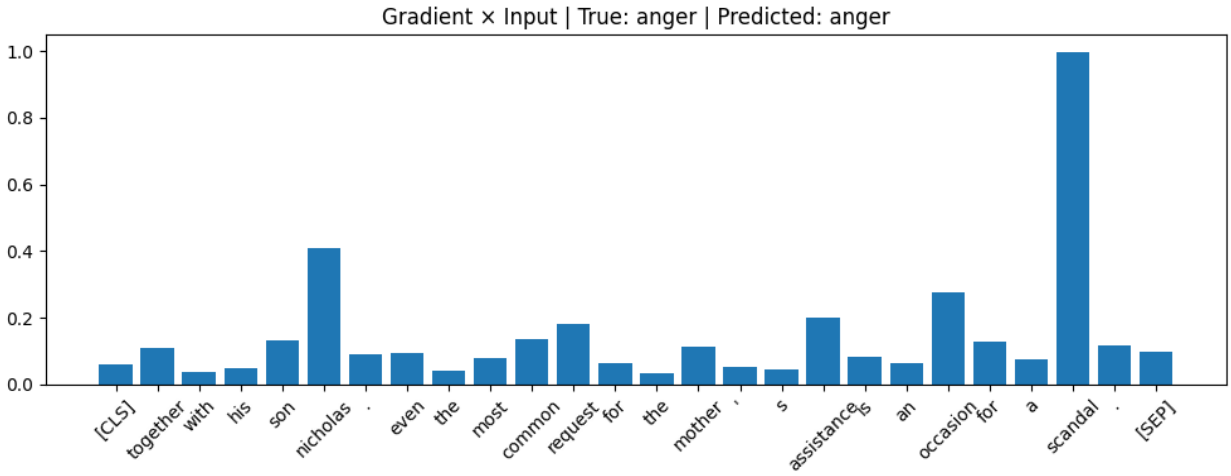
1. **Model and Tokenizer**: We loaded the fine-tuned model (bert_large_8) and tokenizer to process the input sentences.
2. **Tokenization**: Each selected sentence was tokenized using the tokenizer to convert it into a format that the model could process. These tokenized sentences were then passed to the model to generate predictions.
3. **Gradient Calculation**: For each tokenized sentence, we performed a **forward pass** through the model to get the output logits (predictions). Then we performed a **backward pass** to compute the gradients with respect to the predicted class.
4. **Relevance Calculation**: The relevance of each token was calculated using the **Gradient × Input** formula:
   relevance = gradients * inputs. This provided a measure of how much each token contributed to the model's final prediction.
5. **Visualization**: The relevance scores for each token were visualized using **bar plots**, where each bar represents the relevance of a specific token in the sentence. The plots helped us identify which tokens were the most influential for the model's decision.

What Were the Results?

- **Emotionally Significant Tokens**: In most cases, the **emotionally significant words** (such as "angry", "sad", etc.) received **higher relevance scores**, suggesting that the model is indeed focusing on the **right words** for emotion classification.
- **Model's Focus**: The model's focus was **generally aligned** with what a human would consider important words for emotion detection.
- **Misclassifications**: Even in cases where the model made a **misclassification**, the words it focused on still made sense emotionally. This indicates that the misclassifications can be **interpreted and understood** based on the tokens the model paid attention to.

- **Insights from GI**: Gradient × Input gave a **first look** into how the model is making its predictions. It helped us identify if the model is "looking" at the right parts of the sentence, although it is a **simplistic method**.

Figure 1: Example sentence with true and predicted emotion – "anger"



This figure presents the Gradient × Input relevance scores for the sentence "Together with his son, Nicholas, even the most common request for assistance is an occasion for a scandal." The relevance scores indicate the contribution of each token to the model's prediction of anger. The token "scandal" stands out with the highest relevance score, which makes sense for anger detection, as the word "scandal" often carries strong emotional connotations of outrage or disgust.

Other words such as "son", "Nicholas", and "assistance" also show relatively high relevance, suggesting that the context of family relationships and the word "assistance" are important in triggering the model's prediction of anger, possibly due to the negative context surrounding the assistance request.

Other tokens like "the", "most", and "common" have lower relevance, which indicates they play a minimal role in the model's anger prediction.

Key Takeaways:

1. **Gradient × Input** offers useful **initial insights** into token-level relevance, especially in identifying **emotionally significant words**.
2. The model **generally attends to the right tokens**, as observed in the high relevance scores assigned to emotionally charged words.

3. Even when misclassifications occurred, the **emotional relevance** of tokens was still meaningful, suggesting that the misclassifications were likely due to **other factors** (e.g., context).
4. This method is **simple but effective** for providing interpretability at the token level.

Conservative Propagation (LRP)

What Did We Do?

In this part, we applied the **Layer-wise Relevance Propagation (LRP)** technique to provide more accurate and stable explanations of the model's decision-making process. Unlike **Gradient × Input**, which focuses on the immediate gradient information, **LRP propagates relevance** through the layers of the Transformer model, considering how each layer and attention head contributes to the prediction.

The main improvement here is that **LRP** takes into account the internal structure of the model, particularly the **attention mechanism** and **layer normalization**, to distribute relevance across the entire input more evenly.

How Did We Do It?

1. **Model and Tokenizer**: We used the same fine-tuned model and tokenizer as in Part 1.
2. **LRP Implementation**: We applied **Conservative Propagation** by modifying the **attention heads** and **layer normalization** of the model to propagate relevance in a more stable and conservative manner. We leveraged the method from the **XAI for Transformers: Better Explanations through Conservative Propagation** paper.
3. **Token Relevance Calculation**: After propagating the relevance through the layers, we obtained token-level relevance scores. These scores indicated the importance of each token in the final prediction.
4. **Visualization**: Just like in Part 1, we visualized the relevance scores using **bar graphs** for each selected sentence. These graphs showed how the relevance was distributed across tokens, highlighting which words the model relied on most heavily for its predictions.

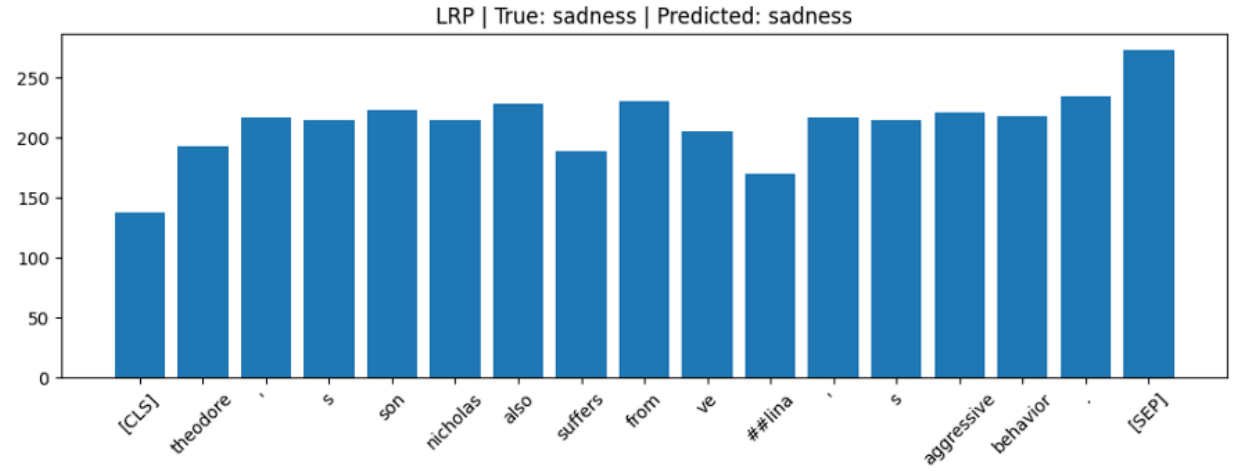
What Were the Results?

- **More Balanced Relevance Distribution**: Compared to **Gradient × Input**, **LRP** provided a **more balanced relevance distribution** across tokens. The relevance

scores were less likely to be concentrated on a single token, and instead, the model seemed to give more evenly distributed attention to the input tokens.

- **Emotionally Charged Words**: Emotionally significant words still had **high relevance scores**, but the **relevance** was distributed more evenly across the sentence, showing that **LRP** captures **context** better than **Gradient × Input**.
- **Better Interpretability**: **LRP** provided more interpretable and **stable explanations**. The relevance scores for tokens were less volatile, making it easier to understand how the model processes the sentence.
- **Improved Focus on Relevant Features**: In many cases, **LRP** highlighted **emotionally relevant tokens** (e.g., words that convey anger, sadness, or happiness), but did so **without overemphasizing a single word**. This shows that **LRP is more stable** and accounts for the overall context better than GI.

Figure 2: LRP Explanation for Sadness Sentence



This figure shows the Layer-wise Relevance Propagation (LRP) results for the sentence "Theodore, his son Nicholas also suffers from aggressive behavior." The bar graph displays the relevance scores for each token in the sentence. The tokens “suffers”, “aggressive”, and “behavior” have the highest relevance scores, indicating that these words significantly contribute to the model’s prediction of sadness. The model pays attention to emotionally significant words related to suffering and aggression, aligning with expectations for sadness detection. Other tokens such as "the" and "his" show lower relevance, suggesting that they contribute less to the emotion classification.

Key Takeaways:

1. **More Stable and Even Relevance**: **LRP** spreads the relevance more evenly across the tokens, making the model's decision-making process easier to interpret. It avoids the sharp focus on a single token that we saw in **Gradient × Input**.
2. **Emotionally Significant Words Remain Key**: While **LRP** distributes relevance more evenly, **emotionally significant words** (such as "angry" or "sad") still received the highest relevance scores, confirming that the model is attending to the right parts of the sentence.
3. **Improved Model Interpretability**: **LRP** improves model interpretability by providing **more stable explanations** and showing how the model processes information through its layers.
4. **Emotion Recognition**: **LRP** is effective in highlighting relevant words for emotion recognition, confirming that the model is relying on **contextual signals** to determine emotions.

Model Robustness with Input Perturbation

What Did We Do?

In **Part 3**, we evaluated how robust the emotion classification model is by applying **input perturbation**. The core idea here is to **systematically remove tokens** based on their relevance scores (calculated using **LRP**) and observe how the **model's confidence** in its predictions changes as more tokens are removed. This helps us understand whether the model relies heavily on specific tokens or whether it uses a more **distributed set of tokens** for making predictions.

How Did We Do It?

1. **Token Removal Based on Relevance**: We first computed the **LRP relevance scores** for each token. These scores indicate how important each token is for the model's final prediction.
2. **Perturbation Process**: We then removed the **least relevant tokens**, one by one, from the input. After each removal, we re-evaluated the model's **confidence** in the prediction to see how it changed as tokens were progressively removed.

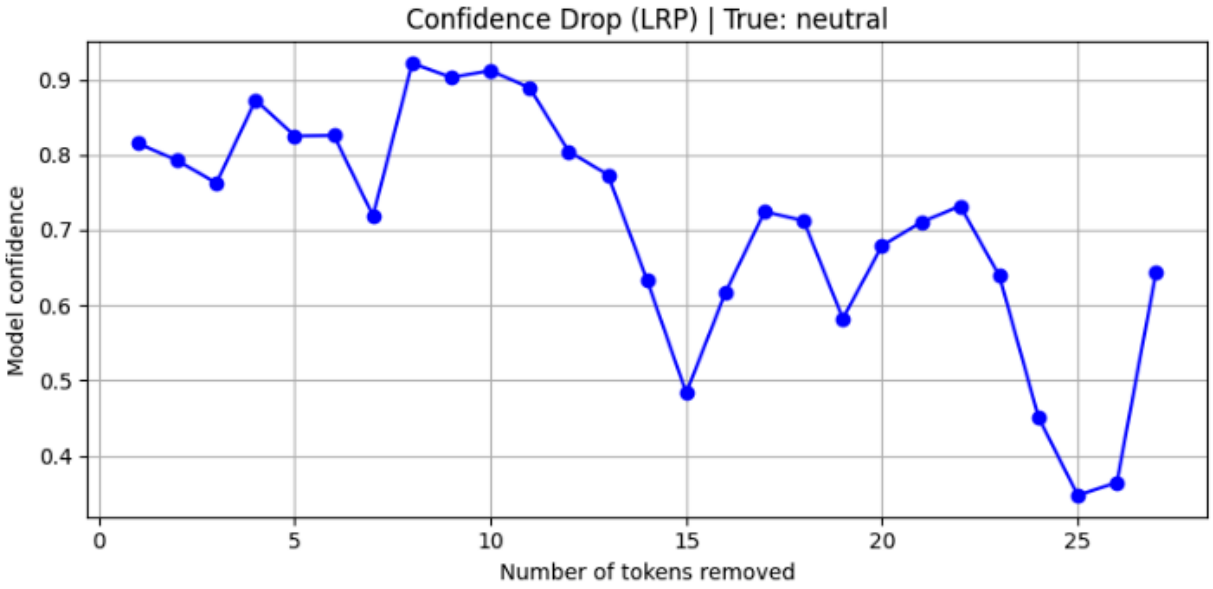
3. **Measuring Model Confidence**: For each perturbed input (after tokens were removed), we calculated the model's **confidence** in its prediction using the **softmax** function. This gave us a measure of how certain the model was in its decision after each token was removed.
4. **Visualizing Confidence Drops**: We plotted the model's **confidence** against the **number of tokens removed**. This line graph allowed us to see whether the model relied on a small set of critical tokens (sharp drop in confidence) or whether it used a broader range of tokens (gradual decrease in confidence).

What Were the Results?

- **Sharp Drops in Confidence**: For some sentences, especially those with **emotionally charged words** (e.g., "angry", "sad"), we observed **sharp drops in confidence** after just a few tokens were removed. This indicates that the model is **highly sensitive** to certain tokens, meaning it relies heavily on a few specific words for making predictions.
- **Gradual Confidence Decrease**: For other sentences, the confidence dropped **gradually** as more tokens were removed. This suggests that the model uses a **more distributed understanding** of the input, making it **more robust** to changes in the input.
- **Emotion-Specific Sensitivity**: The perturbation analysis showed that **certain emotions** (like **anger** or **sadness**) might be **more sensitive** to specific tokens, while others (like **fear** or **disgust**) are more **robust** and rely on a broader set of words.

Figure 3: LRP Confidence Drop for Neutral Sentence

```
Sentence: Sofia. Lilyana Georgieva and her 13-year-old daughter Velina live in the B-5 residential complex.
True Label: neutral
Predicted Label: neutral
--------------------------------------------------------------------------------
```



This figure presents the Layer-wise Relevance Propagation (LRP) relevance scores for the sentence "Sofia. Lilyana Georgieva and her 13-year-old daughter Velina live in the B-5 residential complex." The relevance scores indicate the contribution of each token to the model's prediction of neutral. The tokens "Velina" and "residential complex" have the highest relevance scores, suggesting that the model is focusing on the key details related to the subject of the sentence.

Other tokens such as "Sofia", "Lilyana", and "daughter" also show relatively high relevance, indicating that the model is considering important contextual information, such as the relationship and family structure, to help determine the neutral sentiment.

Tokens like "B-5" and "live" show moderate relevance, pointing to the fact that the model is factoring in the location and action (living) in the sentence, which further supports a neutral classification.

Other tokens such as "is" and "the" have lower relevance, suggesting that they play a minimal role in the prediction of neutral.

Key Takeaways:

1. **Sharp Confidence Drops**: When the model's confidence dropped **sharply** after removing a few tokens, this indicated that the model relies on a **small set of key tokens**. This is common for **emotionally charged words** that carry strong emotional cues.
2. **Gradual Confidence Decrease**: A **gradual drop** in confidence indicated that the model is making decisions based on a **broader set of tokens**. This suggests that the model is **more robust** and not overly reliant on a few tokens.
3. **Emotion-Specific Robustness**: Different emotions showed different levels of **robustness**. Some emotions (e.g., **anger**, **sadness**) are more dependent on specific, emotionally relevant tokens, while others (e.g., **fear**, **disgust**) appear to use a more **distributed understanding** of the input.
4. **Understanding Model Robustness**: This analysis helps us understand the model's **sensitivity** to token removal and provides insights into how it makes decisions. The more **gradual the confidence drop**, the more **robust** the model is.