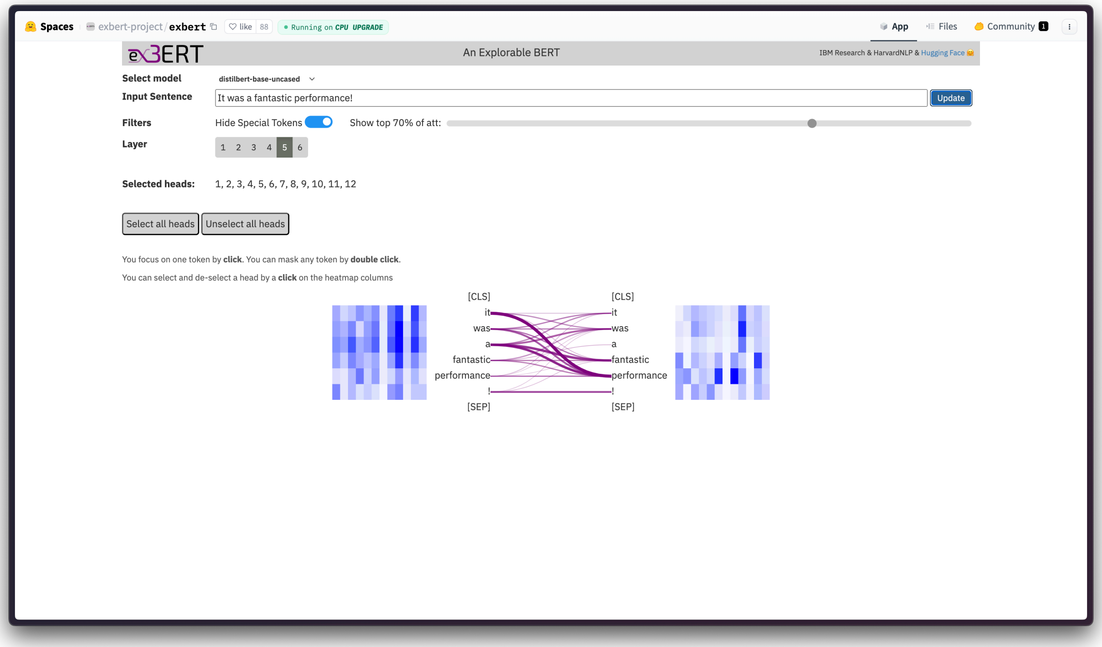
1. Describe your understanding and findings about the attention mechanism by exBERT.
   (a) Understanding the Attention Mechanism
   - Overview of Attention Mechanism
     The attention mechanism in transformer models like DistilBERT allows the model to weigh the importance of different words in a sentence when making predictions. This mechanism helps the model focus on relevant parts of the input text, leading to more accurate and context-aware predictions.
   - Multi-Head Attention
     Transformers use multi-head attention, which means they have several attention heads that operate in parallel. Each head can focus on different parts of the input sequence, capturing various aspects of the context. The outputs of these heads are then combined and processed further.
   - Visualization with exBERT
     exBERT is a tool that visualizes the attention patterns of transformer models. It shows how each token (word) in the input sequence attends to other tokens across different layers and heads. This visualization helps us understand which parts of the input text the model considers important for making predictions.

   (b) Findings from exBERT Visualization
   Using exBERT, I visualized the attention patterns of the `distilbert-base-uncased` model for the sentences ”It was a fantastic performance!” and ”That is a terrible movie.” Here are some key observations:
   - Focused Attention: For the sentence ”It was a fantastic performance!”, the model's attention is heavily focused on the words ”fantastic” and ”performance”. This indicates that the model correctly identifies these words as crucial for determining the positive sentiment of the sentence.



Select model: distilbert-base-uncased

Input Sentence: It was a fantastic performance!

Filters Hide Special Tokens ● Show top 70% of att:

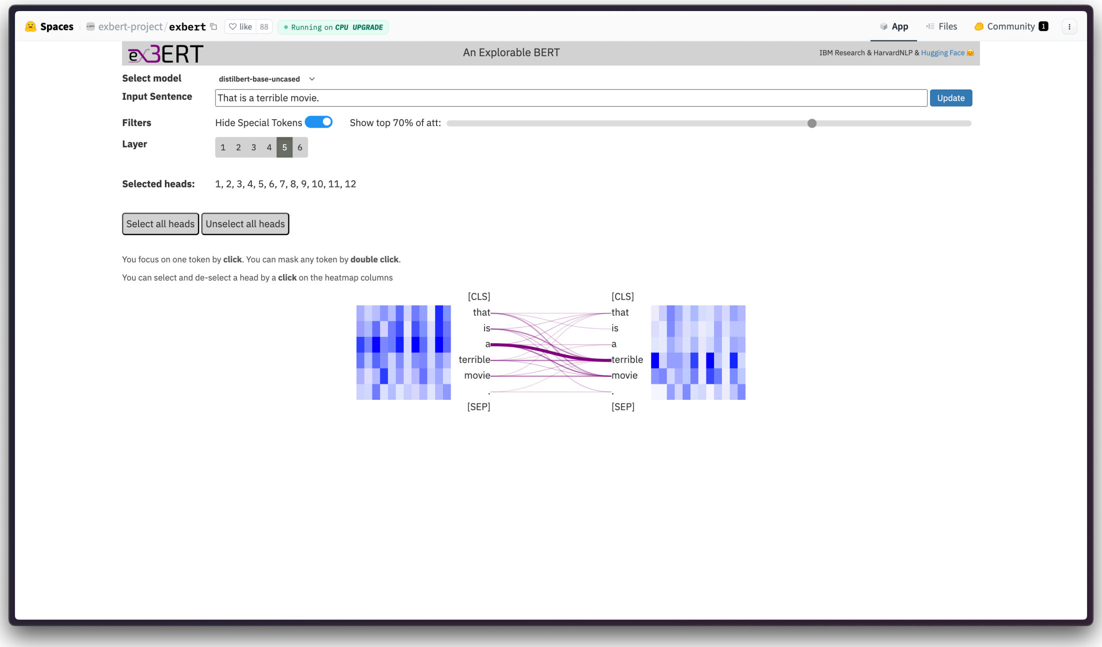
Layer: 1 2 3 4 5 6

Selected heads: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

You focus on one token by **click**. You can mask any token by **double click**.

You can select and de-select a head by a **click** on the heatmap columns

- Distributed Attention: For the sentence ”That is a terrible movie.”, the attention is more distributed but still shows a strong focus on the word ”terrible”. This suggests that the model recognizes the negative sentiment associated with this word.



An Explorable BERT

IBM Research & HarvardNLP & Hugging Face

Select model distilbert-base-uncased

Input Sentence That is a terrible movie.

Filters Hide Special Tokens ● Show top 70% of att:

Layer 1 2 3 4 5 6

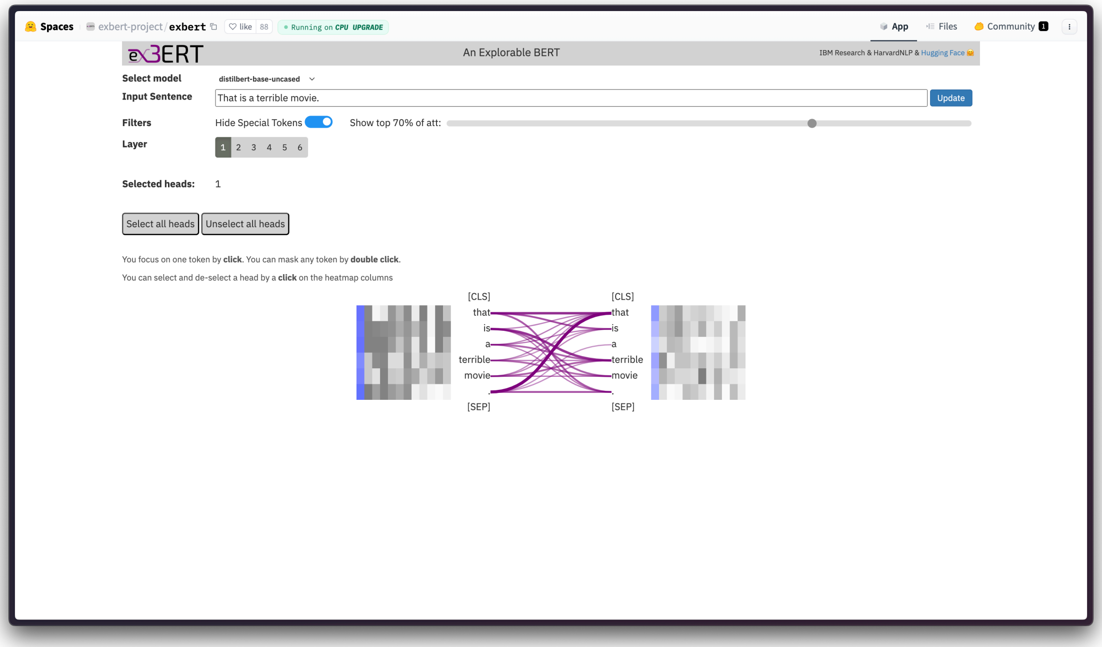
Selected heads: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

You focus on one token by **click**. You can mask any token by **double click**.

You can select and de-select a head by a **click** on the heatmap columns

The attention heads in different layers show varying patterns. Some heads focus on specific words, while others capture broader contextual information. For example:

- Layer 1, Head 1: Might focus on the immediate neighbors of each word, capturing local context.



An Explorable BERT

IBM Research & HarvardNLP & Hugging Face

Select model distilbert-base-uncased

Input Sentence That is a terrible movie.

Filters Hide Special Tokens ● Show top 70% of att:

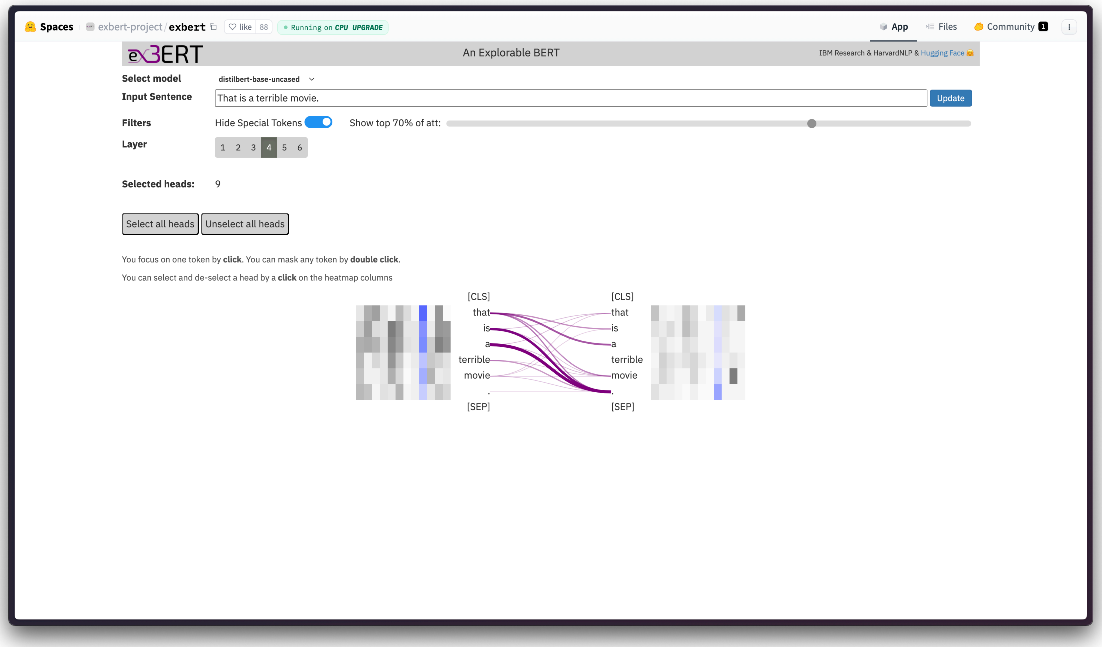
Layer 1 2 3 4 5 6

Selected heads: 1

You focus on one token by **click**. You can mask any token by **double click**.

You can select and de-select a head by a **click** on the heatmap columns

- Layer 4, Head 9: Could capture broader sentence-level information, identifying relationships between distant words.



Select model distilbert-base-uncased

Input Sentence That is a terrible movie.

Filters Hide Special Tokens Show top 70% of att:

Layer 1 2 3 4 5 6

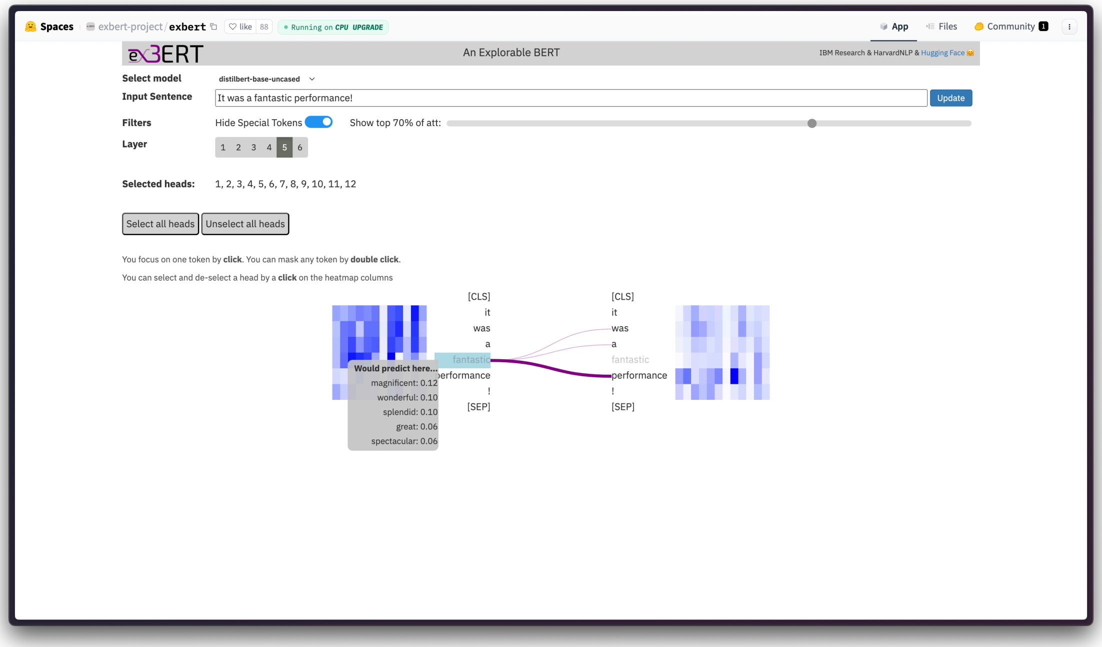
Selected heads: 9

You focus on one token by **click**. You can mask any token by **double click**.

You can select and de-select a head by a **click** on the heatmap columns

The attention heatmaps show high attention scores on words that provide context for the masked word. Here are the findings for the sentence ”It was a [MASK] performance!”:

- Focused Attention
  The thicker purple lines in the visualization indicate stronger attention from tokens like ”was” and ”performance” towards ”fantastic”. This demonstrates that the model identifies ”fantastic” as a key word for predicting sentiment.
- Predictions Based on Attention
  The prediction box in the visualization lists possible words for the masked token. The top predictions (”magnificent”, ”wonderful”, ”splendid”, ”great”, ”spectacular”) all reflect positive sentiments, showing the model’s understanding of the context.



Select model distilbert-base-uncased

Input Sentence It was a fantastic performance!

Filters Hide Special Tokens Show top 70% of att:

Layer 1 2 3 4 5 6

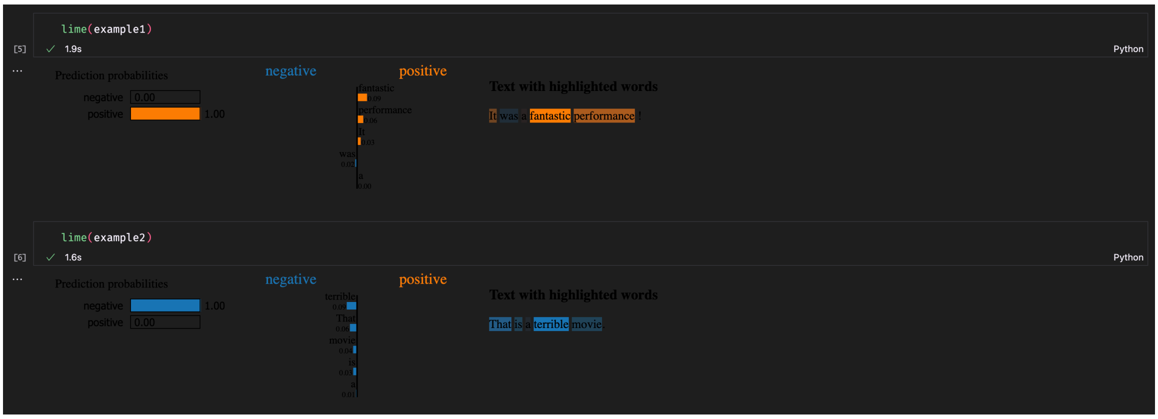
Selected heads: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

You focus on one token by **click**. You can mask any token by **double click**.

You can select and de-select a head by a **click** on the heatmap columns

**Would predict here...**
magnificent: 0.12
wonderful: 0.10
splendid: 0.10
great: 0.06
spectacular: 0.06

2. Compare at least 2 sentiment classification models.
   - Performance
     `TA_model_1.pt` has a slightly higher F1 score compared to `TA_model_2.pt`. This indicates that performs slightly better in terms of classification accuracy.
   - Model Complexity
     `tdistilbert-base-uncased` has a higher dimension compared to `prajjwal1/bert-small`. This means `TA_model_1.pt` is a more complex model with a larger number of parameters.
   - Attention Mechanisms
     - `distilbert-base-uncased` shows more nuanced and distributed attention patterns. It captures more detailed contextual information, which is beneficial for understanding and classifying sentiment.
     - `prajjwal1/bert-small`, being a smaller model, has less complex attention patterns. It might focus on key sentiment words but lacks the depth of contextual understanding compared to `distilbert-base-uncased`.
   - Predictions
     Both models correctly identify the sentiment of the sample sentences, but `TA_model_1.pt` provides more confident and accurate predictions due to its higher complexity and better performance metrics.
3. Compare the explanation of LIME and SHAP.
   - Overview of LIME and SHAP
     - LIME (Local Interpretable Model-agnostic Explanations)
       LIME explains the predictions of any machine learning model by approximating it locally with an interpretable model.
     - SHAP (SHapley Additive exPlanations)
       SHAP assigns each feature an importance value for a particular prediction by computing Shapley values from cooperative game theory.
   - Comparison Using Sentences
     - LIME Explanation
       For the sentences ”It was a fantastic performance!” and ”That is a terrible movie.”, LIME provides the following explanations:
       * Positive Sentiment (”It was a fantastic performance!”)
         LIME highlights words like ”fantastic” and ”performance” as contributing positively to the sentiment prediction. The prediction probabilities indicate a strong positive sentiment with a probability of 1.00. The bar chart shows the importance of each word in determining the model’s prediction.
       * Negative Sentiment (”That is a terrible movie.”)
         LIME highlights words like ”terrible” and ”movie” as contributing negatively to the sentiment prediction. The prediction probabilities indicate a strong negative sentiment with a probability of 1.00. The bar chart also shows the importance of each word in determining the model’s prediction.



```
[5] lime(example1)
    ✓ 1.9s                                                     Python
```

```
[6] lime(example2)
    ✓ 1.6s                                                     Python
```

– SHAP Explanation

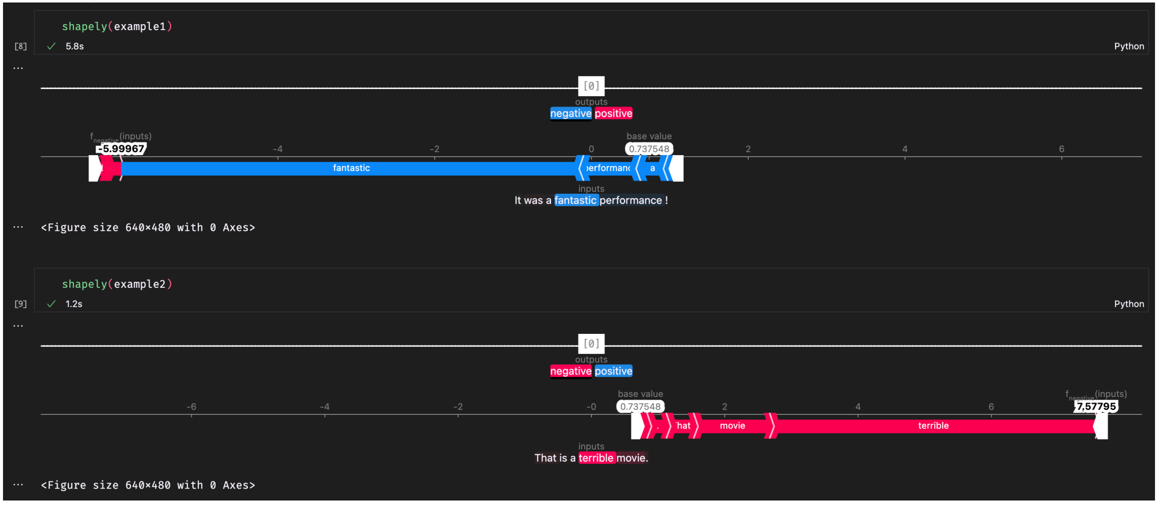
For the same sentences, SHAP provides the following explanations:

* Positive Sentiment (”It was a fantastic performance!”)

SHAP highlights ”fantastic” and ”performance” as the most influential words contributing to the positive sentiment. The force plot shows how each word contributes to pushing the prediction towards the positive class. The base value (0.737548) represents the model’s expected output before seeing any features. ”Fantastic” and ”performance” are shown to have the largest positive impact, significantly moving the prediction towards the positive side.

* Negative Sentiment (”That is a terrible movie.”)

SHAP highlights ”terrible” and ”movie” as the most influential words contributing to the negative sentiment. The force plot shows how each word contributes to pushing the prediction towards the negative class. The base value (0.737548) represents the model’s expected output before seeing any features. ”Terrible” and ”movie” are shown to have the largest negative impact, significantly moving the prediction towards the negative side.



```
[8] shapely(example1)
    ✓ 5.8s                                                     Python
```

```
<Figure size 640×480 with 0 Axes>
```

```
[9] shapely(example2)
    ✓ 1.2s                                                     Python
```

```
<Figure size 640×480 with 0 Axes>
```

Both LIME and SHAP are powerful tools for explaining model predictions, each with its own strengths. LIME is more straightforward and faster, making it suitable for quick, local explanations. SHAP, while more computationally intensive, provides a theoretically sound and detailed explanation of feature contributions, making it ideal for in-depth analysis. The choice between LIME and SHAP depends on the specific requirements of interpretability, computational resources, and the depth of explanation needed.

4. Try 3 different input sentences for attacks. Also, describe your findings and how to prevent the attack if you retrain the model in the future.

   In this part, we will try different input sentences to attack the sentiment classification model.

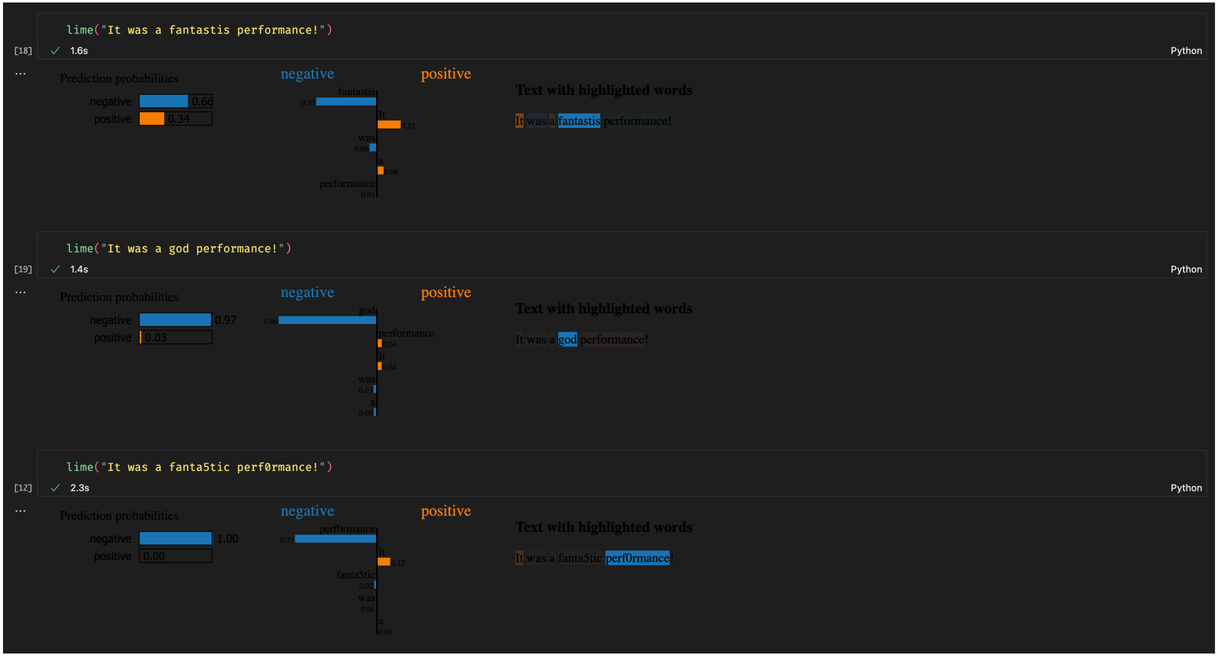
   - Attack Examples

     Original Sentence: ”It was a fantastic performance!”

     (a) Altered Sentence: ”It was a fantastis performance!”
        - LIME Explanation: The model’s prediction changes significantly. The word ”fantastis” is highlighted, and the sentiment shifts from positive to neutral/negative.
        - Prediction Probabilities: Negative: 0.66, Positive: 0.34.

     (b) Altered Sentence: ”It was a god performance!”
        - LIME Explanation: Substituting ”fantastic” with ”god” changes the sentiment drastically. The word ”god” is highlighted, leading to a strong negative prediction.
        - Prediction Probabilities: Negative: 0.97, Positive: 0.03.

     (c) Altered Sentence: ”It was a fanta5tic perf0rmance!”
        - LIME Explanation: Character substitution leads to a significant change in sentiment. The model misinterprets ”fanta5tic”, ”perf0rmance” and shifts the sentiment to strongly negative.
        - Prediction Probabilities: Negative: 1.00, Positive: 0.00.



```
[18] lime("It was a fantastis performance!")
     ✓ 1.6s
```

Text with highlighted words

It was a fantastis performance!

```
[19] lime("It was a god performance!")
     ✓ 1.4s
```

Text with highlighted words

It was a god performance!

```
[12] lime("It was a fanta5tic perf0rmance!")
     ✓ 2.3s
```

Text with highlighted words

It was a fanta5tic perf0rmance!

   - Findings
     - Sensitivity to Key Words: The model’s predictions are highly sensitive to key sentiment words.
     - Vulnerability to Character-Level Attacks: The model is vulnerable to character-level transformations, which can confuse it and lead to incorrect predictions.
     - Contextual Understanding: Although the model captures context, altering key words can easily shift the sentiment, indicating that the model relies heavily on certain words for its predictions.

- Prevention Strategies
  - Data Augmentation: Include augmented data with synonyms to help the model learn that different words can have similar meanings.
  - Adversarial Training: Train the model with adversarial examples created by perturbing the input sentences.
  - Advanced Models: Use ensemble models to combine predictions from multiple models, reducing the impact of any single perturbation.

5. Describe problems you meet and how you solve them.

   (a) `DistilBertModel` object has no attribute `_use_flash_attention_2`

```
"name": "AttributeError",
    "message": "'DistilBertModel' object has no attribute '_use_flash_attention_2'",
    "stack": "---------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[10], line 1
----> 1 lime(example1)

Cell In[9], line 17, in lime(text)
     14 explainer = LimeTextExplainer(class_names=['negative', 'positive'])
     16 ## Generate explanations for a prediction
---> 17 exp = explainer.explain_instance(text, predict, num_features=10, num_samples=500)
     19 ## Visualize explanations
     20 exp.show_in_notebook()
```

Solution: Reinstall `transformers` and its dependencies:

```
pip uninstall transformers
pip install transformers==4.30.0
```