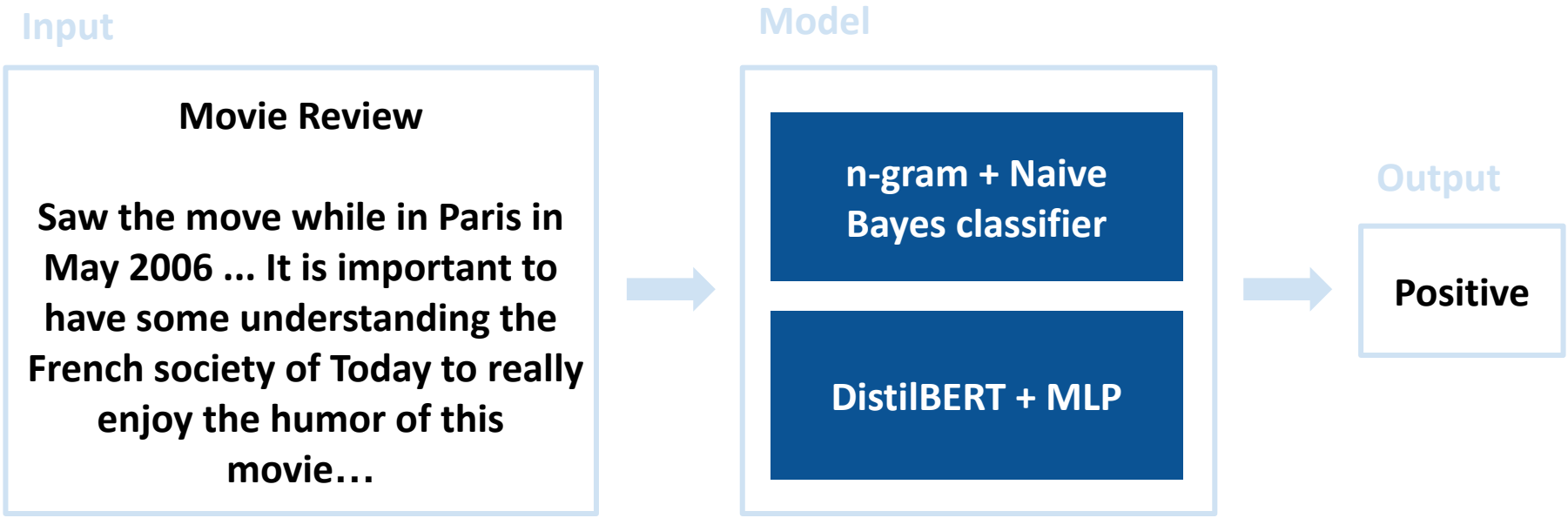
HW4

Introduction to AI

May 10, 2022

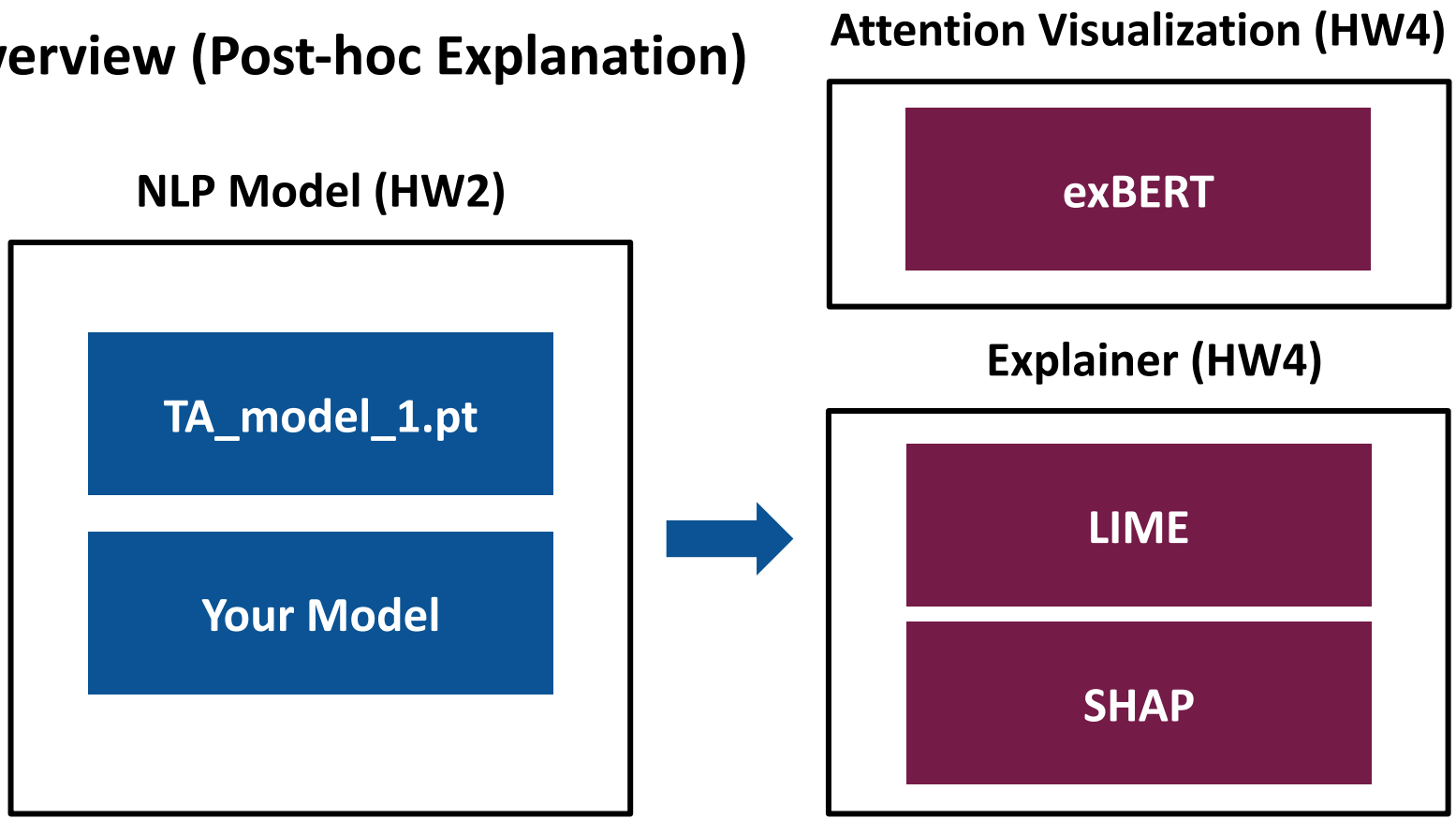
Recap: HW2


Input
Movie Review
Saw the move while in Paris in May 2006 ... It is important to have some understanding the French society of Today to really enjoy the humor of this movie…
Model
n-gram + Naive Bayes classifier
DistilBERT + MLP
Output
Positive

Guide - How to Explain a Model?

Step	Definition	In HW4
1	Decide **which kind of explanation**, i.e., global explanation or local explanation	Show users which words were the most influential in compelling the model to label a sentence as positive or negative sentiment
2	Decide **the form of explanation**, i.e., visual, textual, tabular, graphical	
3	Select the **explainability techniques**	LIME, SHAP
4	Optimise the technique	Explainability helped us build a better model, e.g., find causes in False Positive and biases in the data

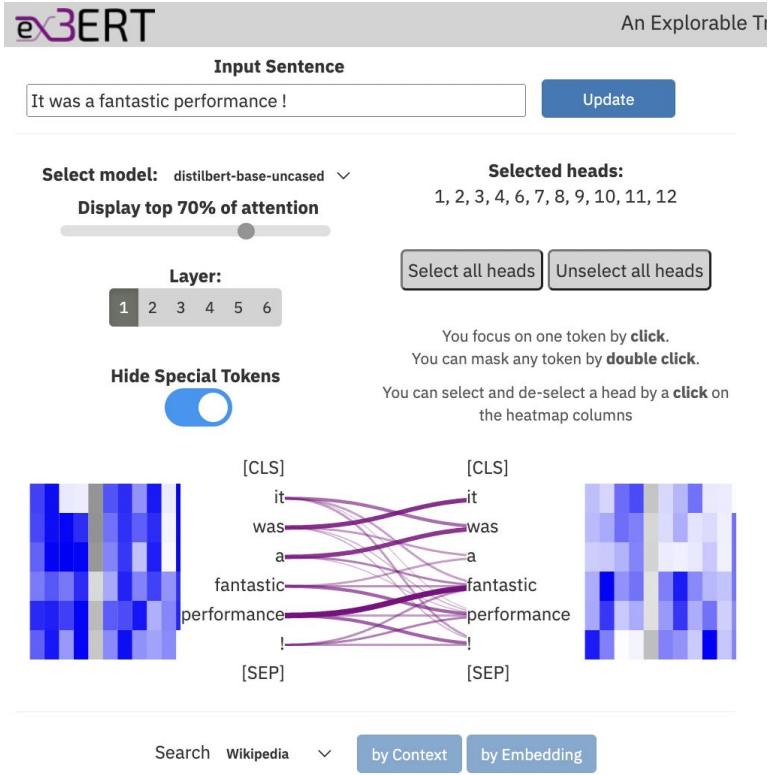
Overview (Post-hoc Explanation)



Attention Visualization - exBERT

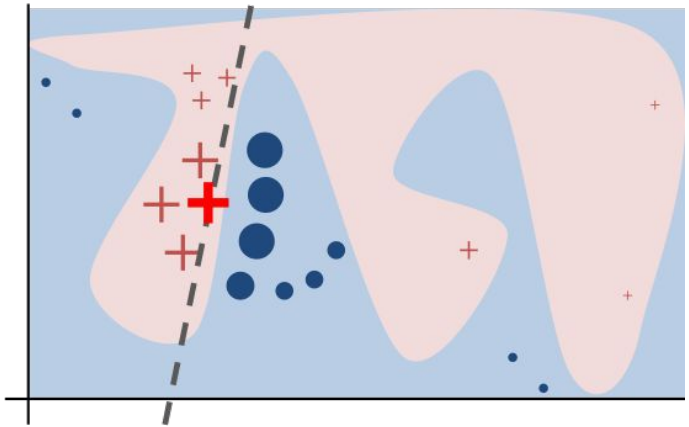
- Website: https://exbert.net/exBERT.html
- Alternative link: https://huggingface.co/exbert/
- Paper: https://arxiv.org/pdf/1910.05276.pdf
- Tutorial: https://youtu.be/e31oyfo_thY

- Select model: distilbert-base-uncased (used in HW2)



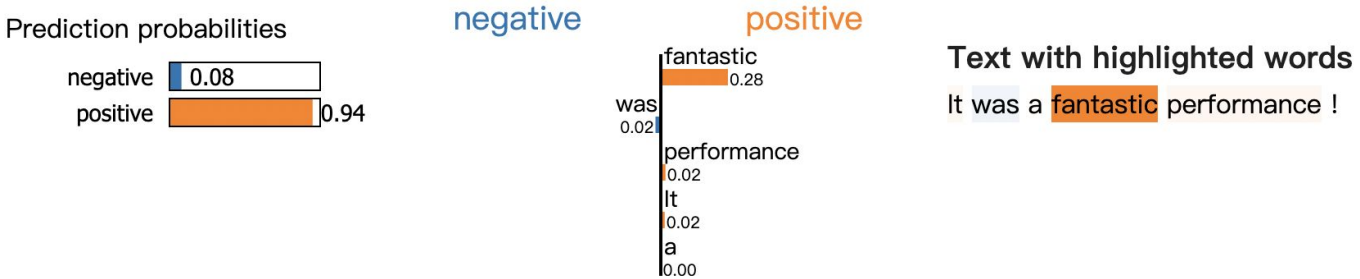
LIME

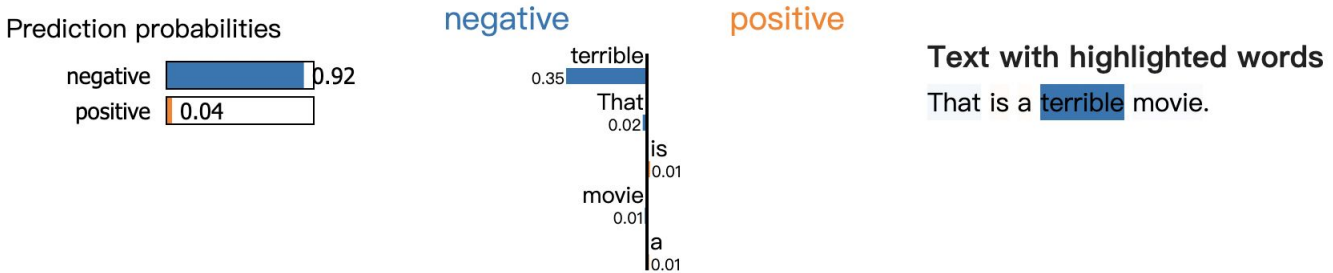
- An explanation of LIME is a **local linear approximation of the model's behaviour.** While the model may be very complex globally, it is easier to approximate it around the vicinity of a particular instance.
- While treating the model as a black box, we **perturb the instance** we want to explain and **learn a sparse linear model** around it, as an explanation.



LIME

- Those messy numbers in the picture define **how a certain word influenced each of the classes**. If a number is >0 then it increased the chance for the class. On the contrary, if it is <0 it decreased it.





LIME

class **lime.lime_text.LimeTextExplainer**(*kernel_width=25, kernel=None, verbose=False, class_names=None, feature_selection='auto', split_expression='\W+', bow=True, mask_string=None, random_state=None, char_level=False*)

Bases: `object`

Explains text classifiers. Currently, we are using an exponential kernel on cosine distance, and restricting explanations to words that are present in documents.

Init function.

explain_instance(*text_instance, classifier_fn, labels=(1,), top_labels=None, num_features=10, num_samples=5000, distance_metric='cosine', model_regressor=None*)

Generates explanations for a prediction.

First, we generate neighborhood data by randomly hiding features from the instance (see __data_labels_distance_mapping). We then learn locally weighted linear models on this neighborhood data to explain each of the classes in an interpretable way (see lime_base.py).

SHAP

- **SH**apley **A**dditive ex**P**lanations
- Shapley Value
  - Map an input to a game where players are the individual features and the payout is the model behavior
  - The contribution of each feature is quantified by the difference between including and not including itself, and average over all subsets

Different expectation of running the model on a modified version of the input

$$\phi(x_i) = \sum_{S \subseteq N \setminus \{i\}} \frac{1}{n\binom{n-1}{|S|}} \left(\nu(S \cup \{i\}) - \nu(S)\right)$$

Subsets except the quantified feature

All possible permutations

Shapley Value - An Example

$$\phi(x_i) = \sum_{S \subseteq N \setminus \{i\}} \frac{1}{n\binom{n-1}{|S|}} \left(\nu(S \cup \{i\}) - \nu(S)\right)$$

- Assume 3 engineers need to do a project with 100 lines of codes, what is the Shapley values of the first engineer (x1) ?

Engineer (S)	Coding ability (val(S))
x_1	10
x_2	30
x_3	5
x_1, x_2	50
x_2, x_3	35
x_1, x_3	40
x_1, x_2, x_3	100

Order	x_1 Contribution	value
x_1, x_2, x_3	$val(x_1)$	10
x_1, x_3, x_2	$val(x_1)$	10
x_2, x_1, x_3	$val(x_2, x_1) - val(x_2)$	$50 - 30 = 20$
x_2, x_3, x_1	$val(x_2, x_3, x_1) - val(x_2, x_3)$	$100 - 35 = 65$
x_3, x_1, x_2	$val(x_3, x_1) - val(x_3)$	$40 - 5 = 35$
x_3, x_2, x_1	$val(x_3, x_2, x_1) - val(x_3, x_2)$	$100 - 35 = 65$
	$\frac{1}{6}(10 + 10 + 20 + 65 + 35 + 65) = 34.17$	

Shapley Value - An Example

$$\phi(x_i) = \sum_{S \subseteq N \setminus \{i\}} \frac{1}{n\binom{n-1}{|S|}} \left(\nu(S \cup \{i\}) - \nu(S)\right)$$

- Assume 3 engineers need to do a project with 100 lines of codes, what is the Shapley values of the first engineer (x1) ?

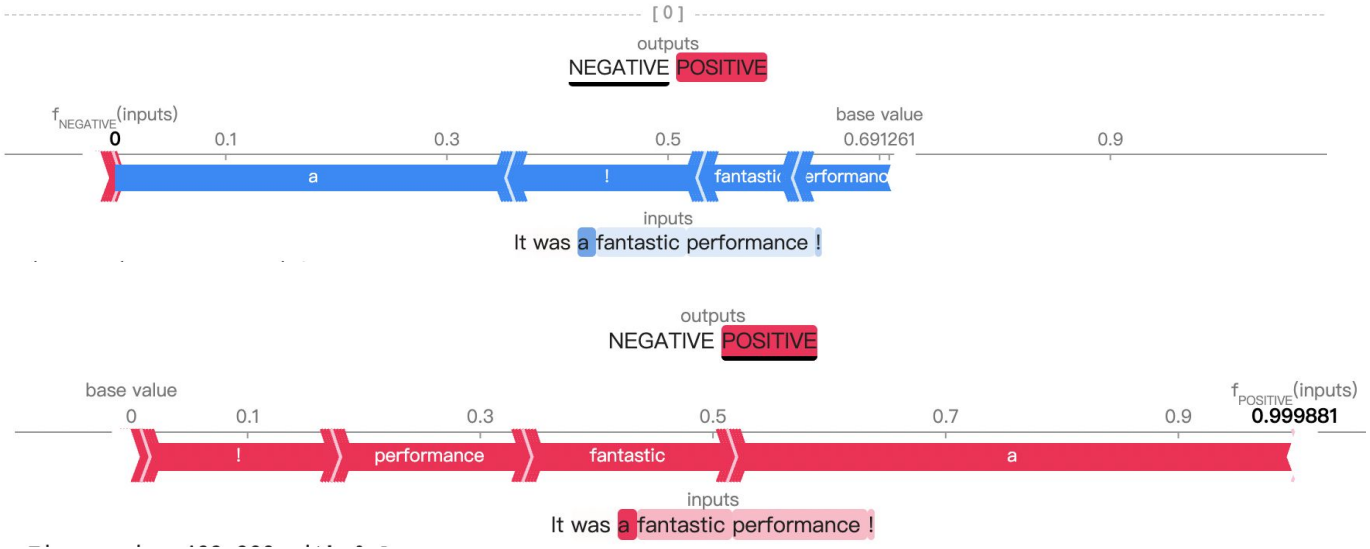
Engineer (S)	Coding ability (val(S))
x_1	10
x_2	30
x_3	5
x_1, x_2	50
x_2, x_3	35
x_1, x_3	40
x_1, x_2, x_3	100

Order	x_1 Contribution	x_2 Contribution	x_3 Contribution
x_1, x_2, x_3	$val(x_1) = 10$	$val(x_1, x_2) - val(x_1) = 50 - 10 = 40$	$val(x_1, x_2, x_3) - val(x_1, x_2) = 100 - 50 = 50$
x_1, x_3, x_2	$val(x_1) = 10$	$val(x_1, x_3, x_2) - val(x_1, x_3) = 100 - 40 = 60$	$val(x_1, x_3) - val(x_1) = 40 - 10 = 30$
x_2, x_1, x_3	$val(x_2, x_1) - val(x_2) = 50 - 30 = 20$	$val(x_2) = 30$	$val(x_2, x_1, x_3) - val(x_2, x_1) = 100 - 50 = 50$
x_2, x_3, x_1	$val(x_2, x_3, x_1) - val(x_2, x_3) = 100 - 35 = 65$	$val(x_2) = 30$	$val(x_2, x_3) - val(x_2) = 35 - 30 = 5$
x_3, x_1, x_2	$val(x_3, x_1) - val(x_3) = 40 - 5 = 35$	$val(x_3, x_1, x_2) - val(x_3, x_1) = 100 - 40 = 60$	$val(x_3) = 5$
x_3, x_2, x_1	$val(x_3, x_2, x_1) - val(x_3, x_2) = 100 - 35 = 65$	$val(x_3, x_2) - val(x_3) = 35 - 5 = 30$	$val(x_3) = 5$
	1/6(10+10+20+65+35+65) =34.17	1/6(40+60+30+30+60+30) =41.7	1/6(50+30+50+5+5+5) =24.17

Sum: 100 lines of codes

SHAP

- SHAP values offer a way of **measuring the relative contribution of each feature** to the output produced by the model



SHAP

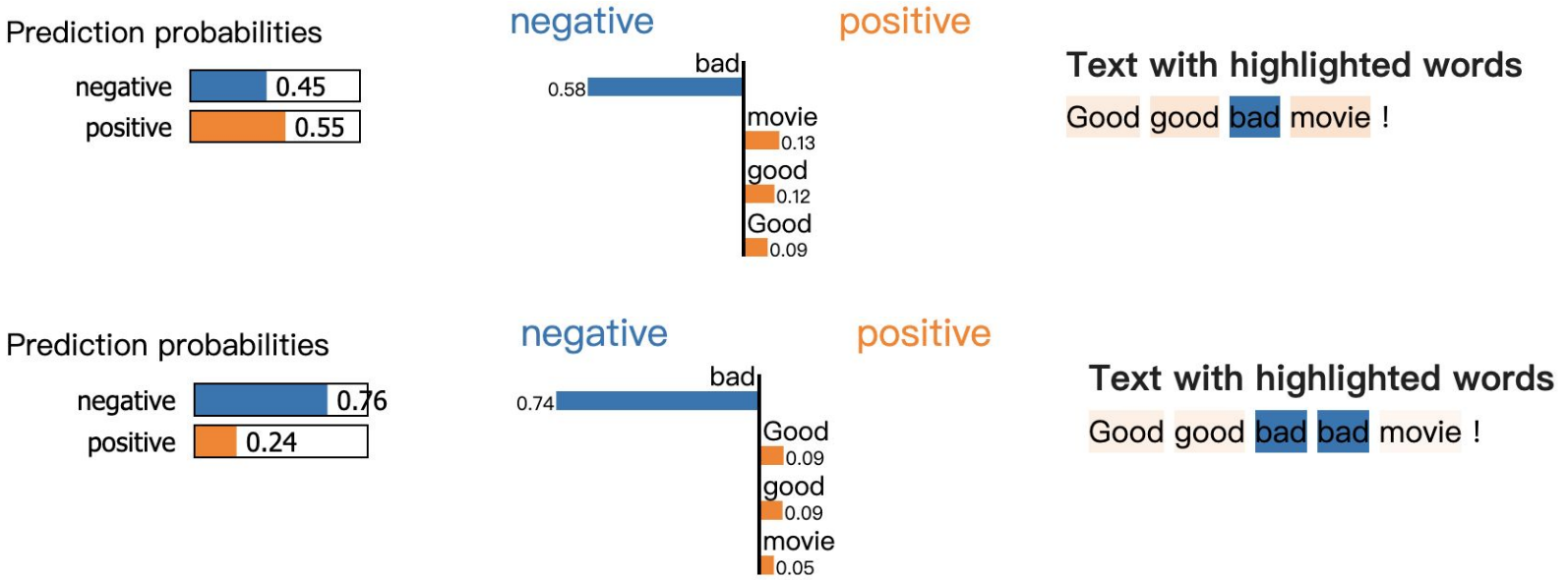
class **shap.Explainer**(*model, masker=None, link=CPUDispatcher(<function identity>), algorithm='auto', output_names=None, feature_names=None, linearize_link=True, **kwargs*)

shap.plots.text(*shap_values, num_starting_labels=0, grouping_threshold=0.01, separator='', xmin=None, xmax=None, cmax=None, display=True*)

Plots an explanation of a string of text using coloring and interactive labels.

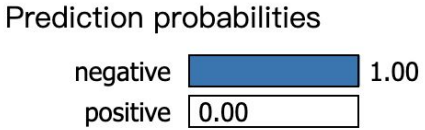
The output is interactive HTML and you can click on any token to toggle the display of the SHAP value assigned to that token.

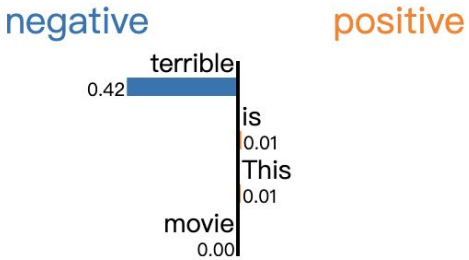
Interesting Example

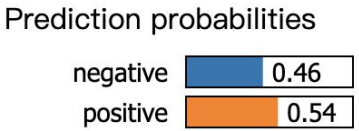


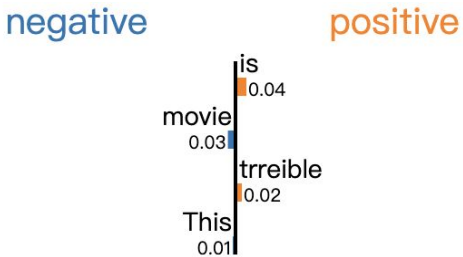
Attacks in NLP

- Misspelling Noise









Reference

- LIME: https://github.com/marcotcr/lime
- SHAP: https://github.com/slundberg/shap
- exBERT: https://exbert.net/
- Some useful XAI Python Libraries: https://github.com/wangyongjie-ntu/Awesome-explainable-AI#python-librariessort-in-alphabeta-order