Using Captum to Explain Generative Language Models

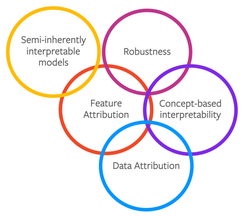
Vivek Miglani*, Aobo Yang*, Aram H. Markosyan, Diego Garcia-Olano, Narine Kokhlikyan
Meta AI

WHAT IS CAPTUM?

Captum is a unified and generic model interpretability library

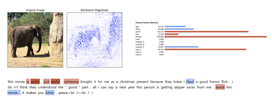
MULTIMODAL

UNIFIED

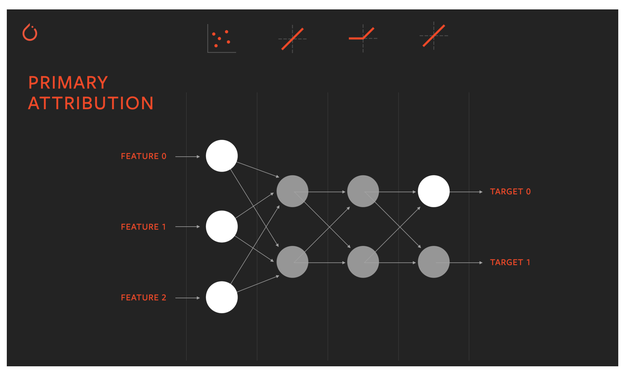


EASY TO USE

```
visualize_image_attr(attr_algo.attribute(input), ...)
```



ATTRIBUTION



Attribute model output (or internal neurons) to input features

- SHAP Methods
  - GradientSHAP
  - DeepLiftSHAP
- DeepLift
- Input * Gradient
- GuidedGradCam
- Integrated Gradients
- Saliency
- Occlusion
- Shapely Value Sampling
- FeatureAblation / FeaturePermutation
- GuidedBackprop / Deconvolution

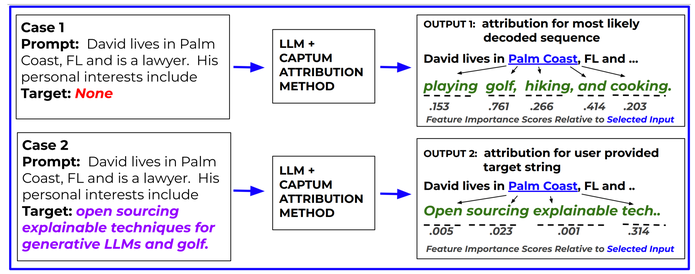
Gradient
Perturbation

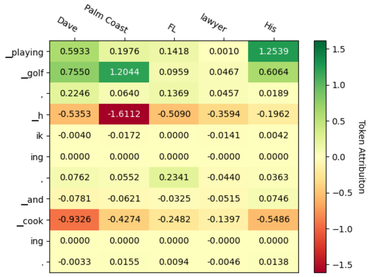
LANGUAGE MODEL ATTRIBUTION

Consider the following example language model prompt and completion
Prompt: David lives in Palm Coast, FL and is a lawyer. His personal interests include ...
Completion: playing golf, hiking, and cooking,

Why did the model predict playing golf?
Is it because of the city Palm Beach, FL?
Is it because of the lawyer occupation?

Captum allow us to define each of these as features and quantify their contribution to the corresponding output.





EXAMPLE USAGE

```python
from captum.attr import ShapleyValueSampling, LLMAttribution,
TextTemplateFeature

shapley_values = ShapleyValueSampling(model)
llm_attr = LLMAttribution(shapley_values, tokenizer)

inp = TextTemplateFeature(
    # the text template
    "{} lives in {}, {} and is a {}. {} personal interests include",
    # the values of the features
    ["Dave", "Palm Coast", "FL", “lawyer", "His"],
    # the reference baseline values of the features
    baselines=["Sarah", "Seattle", "WA", "doctor", "Her"],
)
res = llm_attr.attribute(inp, target=“playing golf”)
```

TRY IT NOW

Install Captum through either pip or conda

```
pip install captum
```

```
conda install captum
```

Learn more about Captum through tutorials and documentation at our website captum.ai

Captum