README

This is the README file for code inside the “Artifact(Feature Attributions)” folder.

================

## 1. Compilation Instructions

The primary setup involves creating a Conda environment and installing the necessary Python packages.

1. Create Environment & Install Dependencies: Run the setup script. This will create a Conda environment named hallucination and install required libraries like PyTorch, Hugging Face Transformers, Captum, and others.

// filepath: d:\Git\BDT\llm-hallucinations-factual-qa\setup.sh

bash setup.sh

1. Activate Environment: Activate the created Conda environment before running any notebooks or scripts.

conda activate hallucination

## 2. Execution Instructions

The Jupyter notebooks (IG\_classifier\_train.ipynb and IG\_collection.ipynb) are designed to be run within a Jupyter environment (like VS Code's notebook editor, Jupyter Lab, or Jupyter Notebook) after activating the hallucination Conda environment and completing the necessary setup steps (especially data collection via Feature\_Attributions\_collection.ipynb for IG\_classifier\_train.ipynb).

* Open the desired notebook (.ipynb file) in your Jupyter environment.
* Execute the cells sequentially.

## 3. Source File Descriptions

**Feature\_Attributions\_collection.ipynb:** This notebook focuses on collecting and comparing different attribution methods (Integrated Gradients via get\_ig, Saliency via get\_saliency, and InputXGradient via get\_input\_x\_gradient) for the TriviaQA dataset using a specified model (e.g., open\_llama\_7b). It runs these methods on dataset examples, saves the collected attributions along with questions, answers, and correctness labels into new .pickle files (prefixed with NEW\_) in the results directory, and includes cells for basic timing comparisons between the attribution methods.

**IG\_classifier\_train.ipynb:** This notebook loads pre-computed attribution data (Integrated Gradients, Saliency, InputXGradient) and correctness labels from .pickle files generated by result\_collector.py and potentially IG\_collection.ipynb. It defines and trains an RNNHallucinationClassifier model using these attributions to predict whether a model's response is correct or a hallucination. The trained Integrated Gradients model is saved to IG\_RNN\_classifier.pth. It evaluates the classifier using metrics like ROC AUC and accuracy.

## 4. Example Usage

1. Run bash setup.sh to set up the environment.
2. Activate the environment: conda activate hallucination.
3. Configure and run **Feature\_Attributions\_collection.ipynb** to generate base artifact data (this takes a long time).
4. Open IG\_classifier\_train.ipynb in VS Code.
5. Run the cells sequentially to load the collected data, train the RNN classifier on IG attributions, save the model, and evaluate its performance. Repeat for Saliency and InputXGradient sections if desired, using data potentially generated by IG\_collection.ipynb.

## 5. Tested Operating Systems

Tested both Feature\_Attributions\_collection.ipynb and IG\_classifier\_train.ipynb in **Windows 10**.