Documentation on Dependencies

This project relies on several Python libraries and frameworks to preprocess data, train the model, and evaluate its performance. Below is a detailed documentation of the dependencies used, their purpose, and installation instructions.

1. Core Dependencies

These are the primary libraries required to run the code:

1. PyTorch (`torch`):

- Purpose: PyTorch is the deep learning framework used to build, train, and evaluate the model.

- Installation:

Command:

pip install torch

- Version: Ensure compatibility with your hardware (CPU or GPU). For GPU support, install the CUDA-enabled version.

2. Transformers (`transformers`):

- Purpose: The Hugging Face `transformers` library provides pre-trained models (e.g., DistilBERT) and tools for natural language processing tasks.

- Installation:

Command:

pip install transformers

- Version: Use the latest stable version for access to the most recent models and features.

3. Datasets (`datasets`):

- Purpose: The Hugging Face `datasets` library provides easy-to-use tools for loading and preprocessing datasets.

- Installation:

Command:

pip install datasets

- Version: Ensure compatibility with the `transformers` library.

4. Pandas (`pandas`):

- Purpose: Pandas is used for data manipulation and preprocessing, such as loading CSV files, handling missing values, and splitting data.

- Installation:

Command:

pip install pandas

5. Scikit-learn (`scikit-learn`):

- Purpose: Scikit-learn is used for splitting the dataset into training and validation sets and computing evaluation metrics.

- Installation:

Command:

pip install scikit-learn

6. NumPy (`numpy`):

- Purpose: NumPy is used for numerical computations, such as calculating evaluation metrics.

- Installation:

Command:

pip install numpy

2. Optional Dependencies

These dependencies are not strictly required but are useful for additional functionality or debugging:

1. tqdm:

- Purpose: Provides progress bars for loops and iterative processes, making it easier to track training progress.

- Installation:

Command:

pip install tqdm

2. Matplotlib/Seaborn:

- Purpose: Used for visualizing data distributions, training curves, or evaluation metrics.

- Installation:

Command:

pip install matplotlib seaborn

3. Jupyter Notebook:

- Purpose: Useful for interactive development and debugging.

- Installation:

Command:

pip install notebook

3. Dependency Management

To ensure reproducibility and avoid version conflicts, it is recommended to use a virtual environment and a dependency management tool like `pip` or `conda`.

Using `pip`:

1. Create a virtual environment:

Command:

python -m venv myenv

2. Activate the virtual environment:

- On Windows:

Command:

myenv\Scripts\activate

3. Install dependencies:

Command:

pip install torch transformers datasets pandas scikit-learn numpy

Using `conda`:

1. Create a conda environment:

Command:

conda create -n myenv python=3.9

2. Activate the environment:

Command:

conda activate myenv

3. Install dependencies:

Command:

conda install pytorch transformers datasets pandas scikit-learn numpy -c pytorch -c conda-forge

4. Dependency List

Below is a consolidated list of all dependencies and their typical versions:

| Dependency | Purpose | Installation Command |

|------------------|----------------------------------------|------------------------------------------|

| `torch` | Deep learning framework | `pip install torch` |

| `transformers` | Pre-trained NLP models and tools | `pip install transformers` |

| `datasets` | Dataset loading and preprocessing | `pip install datasets` |

| `pandas` | Data manipulation and preprocessing | `pip install pandas` |

| `scikit-learn` | Data splitting and evaluation metrics | `pip install scikit-learn` |

| `numpy` | Numerical computations | `pip install numpy` |

| `tqdm` | Progress bars | `pip install tqdm` |

| `matplotlib` | Data visualization | `pip install matplotlib` |

| `seaborn` | Enhanced data visualization | `pip install seaborn` |

| `notebook` | Interactive development | `pip install notebook` |

5. Compatibility Notes

- Ensure that the versions of `torch` and `transformers` are compatible. For example, newer versions of `transformers` may require specific versions of `torch`.

- If using GPU acceleration, ensure that the installed version of `torch` supports your CUDA version.

- The `datasets` library should be compatible with the `transformers` library to avoid issues during data loading and preprocessing.

6. Example `requirements.txt’

For easy installation of all dependencies, you can create a `requirements.txt` file:

plaintext

torch>=2.0.0

transformers>=4.30.0

datasets>=2.12.0

pandas>=1.5.0

scikit-learn>=1.2.0

numpy>=1.23.0

tqdm>=4.65.0

matplotlib>=3.7.0

seaborn>=0.12.0

notebook>=6.5.0

Install all dependencies using:

Command:

pip install -r requirements.txt