Please python scripts after reading the following README and Assignment in detail.

environment.yaml

```yaml
name: msmarco-analysis
channels:
  - conda-forge
  - defaults
dependencies:
  - python=3.9
  - pandas>=1.3.0
  - numpy>=1.21.0
  - matplotlib>=3.4.0
  - seaborn>=0.11.0
  - tqdm>=4.65.0
  - requests>=2.26.0
  - datasets
  - pip
  - pip:
    - ipykernel  # Optional: for Jupyter notebook support
```

MSMARCO Relevance Ranking Analysis

In MSMARCO, there are a few files as following,

```
        # MSMARCO URLs
        self.urls = {
            'collection':
'https://msmarco.blob.core.windows.net/msmarcoranking/collection.tar.gz',
            'queries':
'https://msmarco.blob.core.windows.net/msmarcoranking/queries.tar.gz',
            'qrels':
'https://msmarco.blob.core.windows.net/msmarcoranking/qrels.train.tsv',
            'qrels_dev':
'https://msmarco.blob.core.windows.net/msmarcoranking/qrels.dev.tsv'
        }
```

```
Dataset({
    features: ['query_id', 'doc_id_a', 'doc_id_b'],
    num_rows: 39780811
})
```

MSMARCO Relevance Ranking Analysis

This project provides tools for analyzing the MSMARCO dataset for relevance ranking tasks. It includes comprehensive data analysis, statistics, and visualizations, with integration to Weights & Biases (wandb) for experiment tracking and visualization.

Features

- **Data Analysis**:
  - Query analysis (length, distribution)
  - Document analysis (length, distribution)
  - Relevance analysis (scores, distribution)
  - Query-document relationship analysis

- **Visualization**:
  - Interactive plots through wandb
  - Local PNG file generation
  - Statistical summaries and distributions

- **Experiment Tracking**:

- Track all analysis runs in wandb
- Compare different runs
- Monitor dataset statistics
- Share results with team members

Setup

Option 1: Using Conda (Recommended)

1. Install Miniconda or Anaconda if you haven't already:
   - [Miniconda](https://docs.conda.io/en/latest/miniconda.html)
   - [Anaconda](https://www.anaconda.com/products/distribution)

2. Create and activate the conda environment:

```bash
# Create the environment from the environment.yml file
conda env create -f environment.yml

# Activate the environment
conda activate msmarco-analysis
```

```

### Option 2: Using pip

1. Install the required dependencies:

```bash
pip install -r requirements.txt
```

Downloading the Dataset

The MSMARCO dataset can be downloaded automatically using the provided script:

```bash
python download_msmarco.py
```

This script will:

1. Create a `data` directory
```

2. Download the following files:
   - Collection (documents)
   - Queries
   - Relevance judgments (qrels)
3. Extract and prepare the files
4. Verify the download

Alternatively, you can manually download and prepare the files:

1. Create a `data` directory in the project root
2. Download and place the following files in the `data` directory:
   - `collection.tsv`: MSMARCO document collection
   - `queries.tsv`: MSMARCO queries
   - `qrels.tsv`: MSMARCO relevance judgments

Usage

Running the Analysis

1. Log in to wandb (if you haven't already):

```bash
```

wandb login

```

2. Run the analysis script:

```bash

python msmarco_analysis.py

```

The script will:

1. Load and process the MSMARCO data

2. Generate various statistics about queries, documents, and relevance judgments

3. Create visualizations saved as PNG files

4. Upload all analysis results to wandb

Viewing Results

1. **Local Output**:

   - Statistical summaries in the console

   - Visualization plots saved as PNG files:
```

- Query length distribution

- Documents per query distribution

- Document length distribution

- Document length by relevance score

2. **wandb Dashboard**:

- Access your results at https://wandb.ai/your-username/msmarco-analysis

- View interactive plots and statistics

- Compare different analysis runs

- Track dataset metrics over time

Analysis Features

Query Analysis

- Total number of queries

- Query length statistics

- Query length distribution visualization

- Interactive wandb plots

Relevance Analysis

- Total number of relevance judgments
- Relevance score distribution
- Documents per query statistics
- Documents per query distribution visualization
- Interactive wandb tables and plots

Document Analysis

- Total number of documents
- Document length statistics
- Document length distribution visualization
- Interactive wandb plots

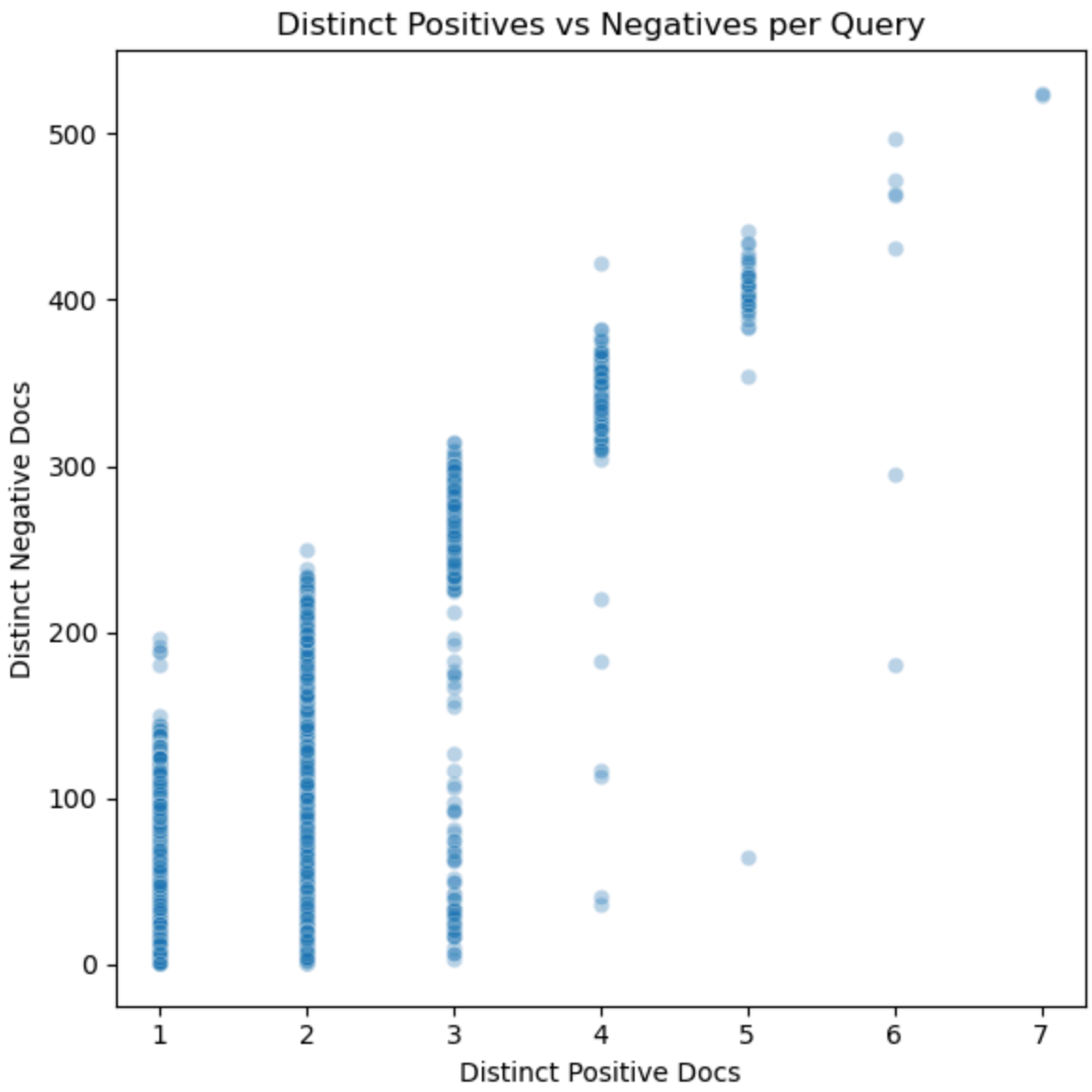
Query-Document Relationship Analysis

- Document length statistics by relevance score
- Visualization of document length distribution by relevance
- Interactive wandb plots and comparisons

Environment Management

To manage the conda environment:

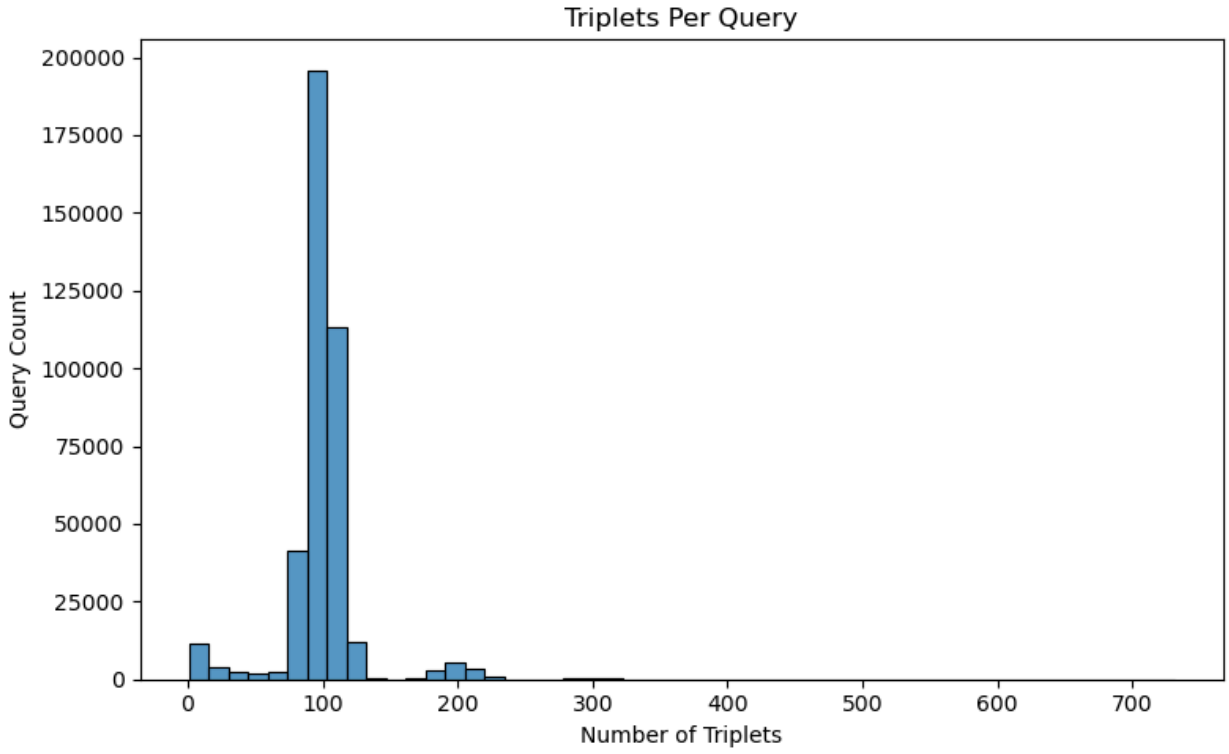
```bash
# Activate the environment
conda activate msmarco-analysis

# Deactivate the environment
conda deactivate
# Remove the environment
conda env remove -n msmarco-analysis
# Update the environment (if environment.yml changes)
conda env update -f environment.yml
```



Positives vs. Negatives Scatterplot

- **What it plots**
  Each point represents one query.
    - The *x-coordinate* is the number of **distinct positive** passages for that query.
    - The *y-coordinate* is the number of **distinct negative** passages for that query.

- **Why it matters**
    - Ideally, you want a good range of both positives and negatives per query so the model can learn to discriminate.
    - If nearly every query has exactly one positive (x=1) but dozens of negatives (y»1), that's typical MS MARCO, but you might consider mining additional positives.
    - Queries appearing near the diagonal (x≈y) have balanced supervision; points far from the diagonal show imbalance.
- **How to interpret**
    - A vertical line at x=1 shows "one relevant passage" per query.
    - Spread along y shows negative sampling richness.
    - Clusters of points with very low y (few negatives) may indicate queries with weak or missing negative supervision.



Triplets Per Query Histogram

- **What it plots**
  On the *x-axis* is the number of triplets associated with a single query; on the *y-axis* is the count of queries that have exactly that many triplets.

- **Why it matters**

    - If most queries have only a handful of triplets, the model sees fewer examples to learn per query.

    - A long right-hand tail (some queries with hundreds or thousands of triplets) indicates a few “heavy” queries that could dominate training and skew the model.

    - A tight cluster around a single value means uniform coverage.

- **How to interpret**

    - A sharp peak at 1 or 2 suggests most queries have only one or two positive–negative pairs.

    - A wide spread indicates diversity: some queries give you many negatives to sample from, which can help robustness.