Prepare Project Environment

STEP-1: Creating and Activating a Virtual Environment on macOS

A virtual environment is an isolated environment that allows you to manage packages specific to a project without affecting the global Python installation.

Create a Virtual Environment

1. **Open your terminal** and navigate to your project directory:

```
cd /path/to/your/project
```

2. **Create the virtual environment** using the `venv` module:

```
python -m venv .venv
```

Activate the Virtual Environment

You need to activate the virtual environment each time you open a new terminal. To activate the virtual environment, use the following command:

```
source .venv/bin/activate
```

STEP-2: Installing Project Dependencies

A `requirements.txt` file is included with this project. Ensure that it is located in the same directory as the project.

1. Ensure your virtual environment is activated.
2. Run the following command:

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

3. Verify Installation (Optional)

```
pip list
```

Project Execution

Running the Project

1. Ensure your virtual environment is activated.
2. First generate keywords from entries.
   To generate keywords, execute `generate_keywords.py` file by following command:

   ```
   python generate_keywords.py
   ```

   On the first execution, the necessary model will be downloaded, and keywords will be generated from the `oecd_findings_recommendations.pkl` file. The generated keywords will then be stored in a new pickle file named `oecd_keywords_df.pkl` for future use.

Generating Word Clouds

1. Ensure your virtual environment is activated.
2. Ensure keywords are generated.
3. To generate Word Clouds, execute `generate_wordclouds.py` file by following command:

   ```
   python generate_wordclouds.py
   ```

   This will generate several word clouds and save them in the `output/word_clouds/` directory. You can modify the output path and word cloud variations in the generate_wordclouds.py file. The word clouds are created using the create_word_cloud method, which you can find defined in the helper.py file.

   Here's an example:

   ```
   create_word_cloud(
       df=df_collection,
       start_year=2000,
       end_year=2010,
       column="recommendations",
       color="blue",
       title="Recommendations Word Cloud (2000-2010)",
       output_dir=output_dir
   )
   ```

   In this example, **`df`** is a pandas DataFrame containing extracted keywords from the `oecd_keywords_df.pkl` file. **`start_year`** and **`end_year`** define the year range. column specifies the keyword variations, with available

options being 'findings', 'recommendations', and 'all'. **`color`** refers to the predefined color set for the word clouds, with available options of 'blue' and 'orange'. **`title`** is the title of the word cloud, and **`output_dir`** is the path where the generated word cloud will be saved.

If you want to generate word clouds for specific country, multiple countries, or regions, you can use an additional parameter called **`countries`.** This parameter accepts either a string or a list. You can pass a single country or region, or multiple countries or regions, such as 'Argentina, Australia, Japan', or just 'Japan', or a region like 'America', or multiple regions like 'America, Asia'. You can also mix regions and countries, such as 'America, Australia, Japan'. The same variations can be provided as a list, like ['Argentina', 'Australia', 'Japan'], or a single country ['Japan'], or a single region ['America'], multiple regions ['America', 'Asia'], or a mix of regions and countries ['America', 'Australia', 'Japan'].

Available regions are 'America', 'Asia', and 'Europe'. You can modify the regions and the countries within them in the `extract_countries` method located in the `helper.py` file.

Here is an example of country-based generation:

```
create_word_cloud(
    df=df_collection,
    start_year=2011,
    end_year=2023,
    countries="Japan",
    column="all",
    color="orange",
    title="Word Cloud (2011-2023)",
    output_dir=output_dir
)
```

Note: Country names must match exactly with those in the extracted data. You can copy the country names from the summary PDF I previously provided.

Generating Time Series Graph

1. Ensure your virtual environment is activated.
2. Ensure keywords are generated.
3. To generate a time series graph, run the `generate_climate_ts.py` file for climate-related graphs or the `generate_gender_ts.py` file for gender-related graphs using the following commands:
   Climate TS Graph:

   ```
   python generate_climate_ts.py
   ```

   Gender TS Graph:

   ```
   python generate_gender_ts.py
   ```

   This will generate time series graphs for the relevant terms and save them in the `output/time_series/` directory. You can adjust the output path and time series variations in the `generate_climate_ts.py` or `generate_gender_ts.py` file as needed.

   For both types of time series graphs, the graphs are created using the `generate_ts` method defined in `helper.py`.

   Here is an example of how to use the `generate_ts` method:

   ```
   generate_ts(
       df=df_collection,
       related_terms=gender_related_terms,
       start_year=2000,
       end_year=2023,
       relevancy="gender",
       color="#4392CC",
       title="Gender-Related Terms Occurrences Over Time (2000-
   2023)",
       x_label="Count of Gender-Related Terms",
       y_label="Year",
       output_dir=output_dir
   )
   ```

   Consider the following: **`df`** is a pandas DataFrame containing extracted keywords from the `oecd_keywords_df.pkl` file. **`related_terms`** are the keywords for which frequency is calculated; you can find and modify this **list in the `terms.py`** file. **`start_year`** and **`end_year`** define the period for the analysis. **`relevancy`** specifies the type of related terms, with options such as 'climate' and 'gender'. **`color`** sets the color of the graph line. **`title`** is the graph's title, while **`x_label`** and **`y_label`** are the labels for the x-axis and y-axis, respectively.

`output_dir` is the directory where the generated time series graph will be saved.

If you want to generate TS graph for specific country, multiple countries, or regions, you can use an additional parameter called **`countries`.** The format of this parameter is the same as Word Cloud's `countries` parameter.

Here is an Example:

```
generate_ts(
    df=df_collection,
    related_terms=gender_related_terms,
    start_year=2000,
    end_year=2023,
    countries="Japan",
    relevancy="gender",
    color="#4392CC",
    title="Japan Gender-Related Terms Occurrences Over Time
(2000-2023)",
    x_label="Count of Gender-Related Terms",
    y_label="Year",
    output_dir=output_dir
)
```

Generating Recommendations Category Charts

1. Ensure your virtual environment is activated.
2. Ensure keywords are generated.
3. To generate charts, run the `generate_categories_graph.py` file using the following commands:

```
python generate_categories_graph.py
```

This will generate category bar charts and save them in the `/output/categories_charts` directory.

The charts are created using the `generate_category_chart` method defined in `helper.py`.

Here is an example:

```
generate_category_chart(
    df=df_collection,
    start_year=2000,
    end_year=2010,
    countries="Europe",
    output_dir=output_dir
)
```

Consider the following: **`df`** is a pandas DataFrame containing extracted keywords from the `oecd_keywords_df.pkl` file. **`start_year`** and **`end_year`** define the period for the analysis. **`countries`** specifies the countries or regions for analysis. The format for countries should be consistent with the other countries parameters. **`output_dir`** is the directory where the generated charts will be saved.

To generate charts for all countries, either remove the **`countries`** parameter or set it to `None`.