Digging Deeper Reaching Further

Libraries Empowering Users to Mine the HathiTrust Digital Library Resources

DDRF Code Reuse Guide

This guide provides directions for how to reuse the Python scripts used DDRF workshop. For each script, we share what it did, the command we ran, and tips for how to reuse it. When additional Python libraries need to be installed, we note that under “Dependencies,” as the script is dependent on those libraries to run.

To follow this guide:

- Find all scripts in the activity_files zip file (download from curriculum page).
- Choose the script you want to run.
- Check that your input data is in the proper format and directory (i.e. folder).
- Install the required dependencies if needed.
- Run the command, replacing the text that appears in ALL CAPS with the input you would like to use.

Module 3: remove_tag.py

We used this script in Module 3 to remove the HTML tags from the file we scraped from Wikisource, which contained the raw HTML.

Command we ran:

```
python remove_tag.py washington_4.txt
```

For reuse:

- This script should be re-usable with other text files that contain HTML tags. Generally, the file should end with .txt or .html.
- Make sure your Python script file (remove_tag.py) is in the same directory (i.e. folder) as the text file from which you would like to remove tags.
- To run, replace “washington_4.txt” with name of the file to which you’d like to apply the remove tags script.

```
python remove_tag.py YOURFILENAME.txt
```

Module 3: remove_stopwords.py

We used this script in Module 3 to remove stopwords from the George Washington speech file from which we had already removed the tags.

Command we ran:

```
python remove_stopwords.py tagless_file.txt stopwords.txt
cleanfile.txt
```

For reuse:

- This script should be re-usable with other text files from which you would like to remove stopwords.
- Make sure your Python script file (remove_stopwords.py) is in the same directory (i.e. folder) as the text file from which you would like to remove tags and the file containing the list of stopwords.
- To run, replace “tagless_file.txt” with the name of the file you would like to clean, and (if you want) stopwords.txt with the name of another replacement file. If using, this other list of stopwords should be a text file, formatted with a single column containing one stopword per row. Finally, include the name of your new output file.

```
python remove_stopwords.py FILE_TO_REMOVE_FROM.txt
LIST_OF_STOPWORDS.txt NEW_OUTPUT_FILE.txt
```

Module 4.2: top_adjectives.py

We used this script in Module 4.2 to create a Pandas dataframe (i.e. a tabular view) of the top adjectives that occured in our set of HTRC Extracted Features file

Dependencies that must be installed:

- Feature Reader python library
- Pandas (note: standard in PythonAnywhere)

Commands we ran:

```
python top_adjectives.py 1970
python top_adjectives.py 1930
```

For reuse:

- The “1930” and the “1970” are arguments required for the script to run. They represent directory names, and the script expects that inside the directories (i.e. folders) there will be at least one HTRC Extracted Features file.
- To run, replace the directory name with the name of the directory containing Extracted Features files that you would like to investigate.

```
python top_adjectives.py YOURDIRECTORYNAME
```

Module 4.2: word_count.py

We used this script in Module 4.2 to create a visualization of the word count per page in a single volume

Dependencies that must be installed:

- ○ Feature Reader python library
- ○ Pandas (note: standard in PythonAnywhere)
- ○ Matplotlib (note: standard in PythonAnywhere)

Command we ran:

```
python word_count.py
```

For reuse:

- Open file and replace the directory (i.e. folder) and file name with the directory location and the name of the Extracted Features file you would like to visualize. Make sure to keep the quotes!
- This file has been optimized to run in PythonAnywhere, and may need slight tweaks to successfully run outside of that environment.

```
python word_count.py
```