CS 5293, Spring 2020 Project 2

Due 4/24 end of day

The Summarizer

Introduction

Millions of documents are push onto the Internet every year. Hundreds of thousands of those documents are academic documents that may be meaningful. Unfortunately, no human is able to wade through the gruff to reach the meaningful documents. Most scientist only look at documents that are published at “prestigious” journals and conferences. In this project, you are going to exercise what you have learned so far in class to help scientists and interested citizes get a lay of the land when it comes to publlished academic work.

In this project, we will cooperate with classmates to summarize as much of the corona virus literature as possible. You will create a package called “The Summarizer”. The package will do that following:

1. Select a subset of academic documents.
2. Tokenize and cluster the documents.
3. Summarize each cluster.

We will use a data set available from the Allen Institute for AI through Kaggle. https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge . You may need to register and get a Kaggle account to access the data.

You will need to show that you have done the following six steps:

1. Explore the data set and look at the format of the files.
2. Write a function to choose a set of documents in the data set, randomly choose 10 percent of the total files.
3. Write a file reader a file and tokenize the data.
4. Write a function to take tokenized data and add it to the clustering method of your choice.
5. Write a function to take clusters of documents, and summarize the documents.
6. Write the summarized clusters to a file.

Below we discuss each of the six steps.

1.Look at the format of the file

The data contains a file called `json_schema.txt` that describes how each json file is set up. Identify the fields that will be important to cluster and summarize the documents. There are several folders full of text and amout 45000 documents. Identify the folders and locations and prepare about 5000 documents. You do not need to store the files in your github version control (they are likely too large).

2. Choose documents

Write a function that takes a glob (representing the data files) and randomly obtains 10% of the files. It would help to make the 10% number a parameter to this function.

3. Write a files reader

Write a function that takes a list of files and Tokenize the data in the given files. Tokenization at this step should be document wide so it may be appropriate to merge all the sentences and paragraphs from each file into a a single large document. You should decide how to develop the tokenizer. Take into consideration the issues of data size, computation time.

4. Cluster documents

Cluster documents given by the previous step. You should experiment to identify the best clustering function and parameters. Use the Silhouette Coefficient to measure the quality of your cluster. Record the documents that are a part of each cluster.

5. Summarize document clusters

Write a function that takes documents in a cluster and in the data set and summarize each document . Note, this task with be very time consuming you may want to **temporarily** increase the size and speed of your instance. To perform this step, use the TextRank algorithm over the *sentences* in each document.

6. Write Summarized clusters to a file

Write your best summarized result in a single file called “SUMMARY.md”. In the file, add a header at the top explaining how the summary was generated.

Submission

Save your code and the result of a successful run to a private github repository. Please also submit your code to Gradescope. In this project, we've relaxed the Python code formating expectations.

Create a private repository GitHub called `cs5293sp20-project2`

Add collaborators `cegme`, `kbanweer`, and `mghirsch42` by going to `Settings > Collaborators`.

COLLABORATORS file

This file should contain a comma separated list describing who you worked with and a small text description describing the nature of the collaboration. This information should be listed in three fields as in the example is below:

```
Katherine Johnson, kj@nasa.gov, Helped me understand calculations
Dorothy Vaughan, doro@dod.gov, Helped me with multiplexed time mar
```

README.md

The README file name should be *uppercase* with an `.md` extension. You should write your name in it, and example of how to run it, and a list of any web or external resources that you used for help. The README file should also contain a list of any bugs or assumptions made while writing the program. Note that you should not be copying code from any website not provided by the instructor. You should include directions on how to install and use the code. You should describe any known bugs and cite any sources or people you used for help. **Be sure to include any assumptions you make for your solution.**

Be prepared to discuss each part of your solution.

When ready to submit, create a tag on your repository using git tag on the latest commit:

```
git tag v1.0
git push origin v1.0
```

Deadline

Your submission/release is due on Friday, April 24th at 11:45pm. Submissions arriving between 11:45:01pm and 11:45pm on the following day will receive a 10% penalty; meaning these scores will only receive 90% of its value. Any later submissions will not receive credits for this project.

Grading

Grades will be assessed according to the following distribution:

- 60%: Correctness.
  - We will look for the presence of all 6 functions and ensure they are written properly.

- 40%: Documentation.
  - Your README file should fully explain your process for developing your code.
  - All other code and results should be well-documented.

- 10%: Bonus
  - Testing and comparing the result of one or more additional sumarization methods. More methods compared and the better quality comparison the more points obtained.

Addendum
