CS235 - Data Mining Techniques Fall 2017 - Assignment

Instructor: Vagelis Papalexakis, University of California Riverside

Introduction

You are thinking about starting research in data mining and related fields. Your friend who is already working on these fields has told you that conferences usually happen in nice and exotic locations and even though this should not affect your decision, knowing that you will get to travel to an exotic location to present your work is always some extra motivation. In this assignment you are going to (empirically) verify your friend's statement by mining WikiCFP http://www.wikicfp.com/cfp/ a website that contains calls for papers for a wide variety of conferences for multiple fields. You will have to 1) crawl the data, 2) clean the data, 3) use Hadoop to compute various statistics of the data, and 4) visualize the results.

Phase 1 [45/100]

Data Crawling [35/100]: Crawl WikiCFP for data mining, machine learning, database, and AI conferences and their location every year. **You have to use the provided Java code and build your crawler based on that**. When searching per category, WikiCFP allows navigation until page 20, so we are going to crawl all 20 pages for data mining, databases, artificial intelligence, and machine learning. The output of the crawling should be in the tab-separated format:

conference_acronym conference_name conference_location

You may stack the results for all four categories in the same file, or have separate files per category.

Hint: You may want to check

http://www.wikicfp.com/cfp/call?conference=data%20mining&page=1

http://www.wikicfp.com/cfp/call?conference=databases&page=1
http://www.wikicfp.com/cfp/call?conference=artificial%20intelligence&page=1
http://www.wikicfp.com/cfp/call?conference=machine%20learning&page=1
Observe the structure of the URL; this will tell you how to navigate through all the results. By selecting “view source” in your web browser, you can see the HTML code that you will eventually need to parse.

IMPORTANT: Make sure you are limiting your queries per WikiCFP policy: http://www.wikicfp.com/cfp/data.jsp . The policy states at most 1 query per 5 seconds, so please set the limiter to 9 or 10 seconds (it might take a bit longer to run so you might want to let your crawler run while you are doing something else). ***Don’t run any code that has no query limiter!*** WikiCFP is letting us crawl their website, so we must respect their rules. In general, whenever you set out to collect data this way, please check for similar policies!! The Java code provided to you has a query limiter. If for some reason you decide (subject to instructor’s approval) to write your own crawler in a different language, make sure you include a query limiter as in the Java code, and in any case **don’t change the code that implements the query limiter!**

Data Cleaning [10/100]: WikiCFP is a relatively well curated site but you will notice that some of the crawled data might have inconsistencies and various other imperfections. Use OpenRefine http://openrefine.org/ to clean the data you just crawled. Describe your process in detail and include screenshots of your data cleaning in the assignment report.

Phase 2 [55/100]

Hadoop [40/100]: In this part you will use Hadoop on the data you just crawled to compute various statistics. The size of the data you have crawled is actually pretty small and is not what Hadoop is made for, but the purpose of this assignment is to get you acquainted with Hadoop nevertheless. We have prepared stand-alone single-node Hadoop installations in CSE’s lab machines. Read this guide carefully to see how you can access to one of those machines and start using Hadoop:
https://sites.google.com/a/ucr.edu/cse-instructional-support/home/hadoop .
We are going to assign to each of you a lab machine and you will be able to retrieve it if you log in iLearn and go to your grades, where we have created a special column with the lab machine.
In order to get started with this part of the assignment, after you get Hadoop up and running according to the instructions above, the next step is to read the excellent tutorial provided by Apache on how to start HDFS (the Hadoop File System) and YARN (which manages Map/Reduce). In particular, please read carefully Sections “Execution” and “YARN on a Single Node” in
https://hadoop.apache.org/docs/r2.7.0/hadoop-project-dist/hadoop-common/SingleCluster.html
After you get HDFS and YARN running, you can go ahead and start writing Hadoop programs. The best way to do that is to refer to the Apache tutorial here:
https://hadoop.apache.org/docs/r2.7.0/hadoop-mapreduce-client/hadoop-mapreduce-client-core

/MapReduceTutorial.html
and more specifically the “WordCount” example that is the most typical Hadoop/MapReduce example. In fact, most of what you have to do in this assignment will be a modification of the WordCount example, so make sure you understand how it works, and make sure you can compile it and run it successfully before you start implementing what is needed for this part of the assignment.
Finally, after you have successfully made “WordCount” work, you will have to do the following computations:

1. Compute and output the number of conferences per city. Which are the top 10 locations? [10/40]
2. Output the list of conferences per city [10/40]
3. For each conference regardless of the year (e.g., KDD), output the list of cities. [10/40]
4. For each city compute and output a time series of number of conferences per year [10/40]

You may have a separate Hadoop program per computation.
Note: If you can demonstrate to the instructor that you have *extensive* prior experience with Hadoop, *only with instructor’s permission*, you may use Spark instead of Hadoop for this part of the assignment.

Result Visualization [15/100]: As we discussed in class, visualizing the results in an intuitive way is extremely important, especially when external input from domain experts is needed. Here, you have to visualize the output of question #4 of the Hadoop section (i.e., the number of conferences per year, for each city); the visualization must be done on a map and the number of conferences can be represented as a heatmap (“red”/”hot” means a lot of conferences, “blue”/”cold” means few conferences) with an associated assignment of colors to numbers, or with variable size dots (large dot means a lot of conferences, small dots means a few). You are free to use any map interface you wish, however, you must clearly describe your approach (providing screenshots or code, whenever applicable) and provide image files with the visualizations.

Deliverables

Each phase consists of the following two deliverables:

1. A report where you document your solution for each part (including the screenshots of the data cleaning). The report should also include the results of the third part, as well as the plots required.
2. A copy of all the code you wrote for the crawler and the Hadoop programs. Please make sure you properly document your code with comments.

Each deliverable should be uploaded as an archive on iLearn.

Grading scheme

The respective points for each part of the assignment are shown in the description. For each part we are going to evaluate both how well you describe your approach (and this is based on the report you will submit) as well as whether you actually implemented everything successfully and generated the desired results. In particular, the breakdown (per each section of each phase) is:

1. 10% for describing your approach in detail.
2. 90% for implementation and results

Academic Integrity

If you use any external packages or help from the web (e.g., StackOverflow) please cite your sources! Same goes if one of your colleagues helped you with some part. You will not lose points using such help as long as you cite your sources.