COSC-589  
Web Search and Sense-making  
*Information Retrieval In the Big Data Era*

Spring 2016

Instructor: Grace Hui Yang

The Web provides abundant information which allows us to live more conveniently and make quicker decisions. At the same time, the growth of the Web and the improvements in data creation, collection, and use have lead to tremendous increase in the amount and complexity of the data that a search engine needs to handle. The increase of the magnitude and complexity of the data has become a major drive for new data analytics algorithms and technologies that are scalable, highly interactive, and able to handle complex and dynamic information seeking tasks in the big data era. How to effectively and efficiently search for the documents relevant to our information needs and how to extract the valuable information and make sense out from "big data" are the subjects of this course.

The course will cover Web search theory and techniques, including basic probabilistic theory, representations of documents and information needs, various retrieval models, link analysis, classification and recommender systems. The course will also cover programming models that allow us to easily distribute computations across large computer clusters. In particular, we will teach Apache Spark, which is an open-source cluster computing framework that has soon become the state-of-the-art for big data programming. The course is featured in step-by-step weekly/bi-weekly small assignments which compose a large big data project, such as building Google’s PageRank on the entire Wikipedia. Students will be provided knowledge to Spark, Scala, Web search engines, and Web recommender systems with a focus on search engine design and "thinking at scale".

Prerequisites: None.

Course Website: https://piazza.com/georgetown/spring2016/cosc589/home

Class Time:  Monday and Wednesday 1100-1215  
Venue: Intercultural Center 205  
Blackboard: Homework submission

Textbooks:

*Information Retrieval: Implementing and Evaluating Search Engines*  
By Stefan Büttcher, Charles L. A. Clarke and Gordon V. Cormack  
Publisher: The MIT Press  
July 2010

*Learning Spark: Lightning-Fast Big Data Analysis*  
By Holden Karau, Andy Konwinski, Patrick Wendell, Matei Zaharia  
Publisher: O'Reilly Media
January 2015

By Martin Odersky, Lex Spoon, and Bill Venners
Publisher: Artima
January 2011

Lecturer:

Prof. Grace Hui Yang
Email: gh243@georgetown.edu
Office Phone: 202 687 6355
Website: http://www.cs.georgetown.edu/~huiyang

Teaching Assistants:

Jiyun Luo (jl1749@georgetown.edu)
Website: http://cs-class.uis.georgetown.edu/~jl1749/homepage/

Office Hours:
  Prof. Grace Hui Yang
  St Mary’s Hall 338
  Tuesday 1-2pm

  TA hours
  St Mary’s Hall TA room 330
  Wednesday evening 6-7:30pm
Grading:

- Midterm Exam 15%
- Assignments 80%
- Pop Quizzes 5%
- Bonus - Optional Assignment Problems 20%

Homework policy:

All homework should be submitted through Blackboard. Homework is due by 11:59pm on the due date. Three late days in total are allowed without penalty for the entire semester. For instance, you may be late by 1 day for homework 1 and be late by 2 days for homework 2. Once the three-late-dates are used, you will be penalized according to the policy below:

- a penalty of 50% will be applied for homework submitted within the next 24 hours.
- zero credit will be assigned after that.

All homework (even with zero credit) must be turned in order to pass the course.

Integrity policy:

All experimental results turned in must be true. No copying/cheating is allowed. Please check Georgetown's Honor system.
<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/13/16</td>
<td>Lecture: Introduction, Spark Installation</td>
</tr>
<tr>
<td></td>
<td>Readings: Chapter 1, 2</td>
</tr>
<tr>
<td></td>
<td>Assignment 1: Due 1/20/16 Install Spark</td>
</tr>
<tr>
<td>1/18/16</td>
<td>Break: Martin Luther King Day</td>
</tr>
<tr>
<td>1/20/16</td>
<td>Lecture: Scala Crash Course</td>
</tr>
<tr>
<td></td>
<td>Readings: Scala Book Ch 2,3,7,8</td>
</tr>
<tr>
<td></td>
<td>Assignment 2: Due 1/27/16 Scala practice</td>
</tr>
<tr>
<td></td>
<td>Due: Assignment 1</td>
</tr>
<tr>
<td>1/25/16</td>
<td>No class</td>
</tr>
<tr>
<td>1/27/16</td>
<td>Lecture: Scala Crash Course, SBT, Passing functions</td>
</tr>
<tr>
<td></td>
<td>Readings: Scala Book Ch 2,3,7,8, Chapter 2</td>
</tr>
<tr>
<td></td>
<td>Assignment 3: Due 2/3/16 Scala practice</td>
</tr>
<tr>
<td></td>
<td>Due: Assignment 2</td>
</tr>
</tbody>
</table>
2/1/16
Lecture:
  RDD
  File Operations

Readings: Chapter 3
          Chapter 5

2/2/16
Lecture:
  RDD
  Transformation
  Actions
  Spark UI

Readings: Chapter 3
          Chapter 7

Assignment 4: Due 2/10
  Word Count

Due: Assignment 3

2/3/16
Lecture:
  RDD
  Transformation
  Actions
  Spark UI

Readings: Chapter 3
          Chapter 7

Assignment 4: Due 2/10
  Word Count

Due: Assignment 3

2/8/16
Lecture:
  RDD
  Transformation
  Actions
  Write SBT Build File

Readings:
  Chapter 3
  Chapter 2

2/10/16
Lecture:
  RDD
  Persistence
  Regular Expression

Readings:
  Chapter 3
  Scala Book Section 26.7

Assignment 5: Due 2/17
  Extract Patterns

Due: Assignment 4
<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>2/15/16</td>
<td>Break: President Day</td>
<td>2/17/16</td>
<td>Lecture:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Text Processing</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Bag of Words</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Query and Documents</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Vector Space Model</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Readings:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IR Book Chapter 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IR Book Chapter 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Assignment 6: Due 3/2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Download Wikipedia Dump</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Clean the dataset</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Due: Assignment 5</td>
</tr>
<tr>
<td>2/17/16</td>
<td></td>
<td>2/22/16</td>
<td>Break: WSDM Week</td>
</tr>
<tr>
<td>2/22/16</td>
<td></td>
<td>2/24/16</td>
<td>Break: WSDM Week</td>
</tr>
<tr>
<td>2/29/16</td>
<td>Lecture:</td>
<td>3/2/16</td>
<td>Lecture:</td>
</tr>
<tr>
<td></td>
<td>Accumulators</td>
<td></td>
<td>XML Parser</td>
</tr>
<tr>
<td></td>
<td>Broadcast Variables</td>
<td></td>
<td>Readings:</td>
</tr>
<tr>
<td></td>
<td>Readings:</td>
<td></td>
<td>Scala Book Chapter 28</td>
</tr>
<tr>
<td></td>
<td>Chapter 6</td>
<td></td>
<td>Assignment 7: Due 3/14</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Parse the Wikipedia data using Scala.xml</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Word Count</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Due: Assignment 6</td>
</tr>
<tr>
<td>3/2/16</td>
<td></td>
<td>3/7/16</td>
<td>Spring Break</td>
</tr>
<tr>
<td>3/7/16</td>
<td></td>
<td>3/9/16</td>
<td>Spring Break</td>
</tr>
<tr>
<td>3/14/16</td>
<td>Lecture: Midterm Review</td>
<td>3/16/16</td>
<td>Midterm Exam (in class, closed book)</td>
</tr>
<tr>
<td></td>
<td>Due: Assignment 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3/16/16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>Events</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3/21/16</td>
<td>Break: ECIR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3/23/16</td>
<td>Easter Break</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3/28/16</td>
<td>Easter Break</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| 3/30/16 | Lecture: Web Search, Link Graph  
Readings: IR Book Chapter 15  
Assignment 8: Due 4/6  
Build Wikipedia Link graph  
Write to a file |
| 4/4/16  | Lecture: PageRank, Key/Value Pairs  
Readings: Chapter 4, IR Book Chapter 15 |
| 4/6/16  | Lecture: PageRank  
Readings: IR Book Chapter 15, PageRank paper  
Assignment 9: Due 4/20  
Compute PageRank for Wikipedia pages  
Due: Assignment 8 |
<p>| 4/11/16 | Break: WWW Conference |
| 4/13/16 | Break: WWW Conference |</p>
<table>
<thead>
<tr>
<th>Date</th>
<th>Lecture</th>
<th>Readings</th>
<th>Date</th>
<th>Lecture</th>
<th>Readings</th>
<th>Assignments</th>
</tr>
</thead>
<tbody>
<tr>
<td>4/18/16</td>
<td>Web Knowledge Base</td>
<td>IR Book Chapter 15</td>
<td>4/20/16</td>
<td>Recommender Systems</td>
<td>Chapter 11</td>
<td>Assignment 10: Due 5/2 Compute Topic-oriented PageRank for Wikipedia</td>
</tr>
<tr>
<td></td>
<td>Topic-Oriented PageRank</td>
<td>Papers</td>
<td></td>
<td></td>
<td>IR Book Section 10.1</td>
<td>Due: Assignment 9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4/25/16</td>
<td>Regression &amp; Classification</td>
<td>Chapter 11</td>
<td>4/27/16</td>
<td>Advanced Search</td>
<td>Papers</td>
<td>Assignment 11 (optional): Due 5/14 Secret</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Papers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5/2/16</td>
<td>(Class End)</td>
<td></td>
<td>5/7/16</td>
<td>Study Week</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Conclusion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Due: Assignment 10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5/9/16</td>
<td>Exam Week</td>
<td></td>
<td>5/14/16</td>
<td>Exam Week</td>
<td></td>
<td>Due: Assignment 11</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

See Sample Lecture Notes on the Next Page.

It is also available at http://people.cs.georgetown.edu/~huiyang/cosc-589/intro.pdf
IF YOU ARE THE GOD OF DATA
- What are the challenges/problems when your data is big?
- What is your solution?
  - Parallel/Concurrent
  - Compression

MAP REDUCE
- Parallel/Concurrent
- Cloud
- In-memory

MAPREDUCE IS A LITTLE BIT OUTDATED
- Map goes to the data and does its thing
- Reduce is not efficient enough for multiple map algorithms
- Reduce does not scale relatively (looking at other algorithms)
- Same goes for the system
  - AWS/EC2
  - HEC

SPARK
- New Java
- Large in-memory
- Resilient Distributed Dataset (RDDs)
- Cache MR in Java, Scala, Python, R

COURSE PLAN
- MapReduce
- Spark
- Programming language
  - Java
  - Scala
  - Python

ASSIGNMENTS
- Build Distributed pagerank algorithm over Wikipedia
- Aim to produce the final system
- Final report ( NW of 30%)
- Final report is due on Wednesday, 11:00 PM

HIGHLIGHTS OF TODAY
- Spark
- Install Spark

STEP 3: RUN SPARK SHELL
- We’ll run Spark’s interactive shell
- Write the “spark” directory name
- Spark shell
- Then run the “test” script
- Let’s create some data...

STEP 4: CREATE AN RDD
- Create an RDD based on the data...
- val data = sc.parquetFile("...
- Then use filter to select values less than 20...
- val data1 = data.filter(20)

STEP 5: INSTALL SPARK
- Note: Please download Spark using...
- browser or Mac/OS
- Cygwin on Windows

STEP 2: GET SPARK
- We will use Spark 1.5.0
- 1 copy from the SNS site
- 1 copy from the user content directory
- Do you need to download it from spark.apache.org/download.html

ASSIGNMENT 1
- Use a filter to select values less than your age
- Submit 3 screen captures of the results of above program

SUMMARY
- Data engine
- Spark
- MLlib
- Machine learning
- Deep learning
- Deep neural networks
- Deep learning for TensorFlow