Cloud-Based Text Analytics: Harvesting, Cleaning and Analyzing Corporate Earnings Conference Calls

MICHAEL (CHUANCAI) ZHANG, VIKRAM GAZULA, DAN STONE, HONG XIE
Thanks!

• Jim Griffioen - Director of Center for Computational Science
• Gatton College of Business - $
• Von Allmen School of Accountancy - $
• Amazon Web Services (AWS) – help and support
• Vikram Gazula – IT manager - Center for Computational Science
• My coauthors
The research problem

- Corporate earnings conference calls convey information to financial markets
- Existing analysis of conference calls = “bag of words” analysis
  - Simple, short word lists
  - No analysis of sentences, paragraphs, context, or meaning
- Our goal: analyze conference call data using emerging “holistic” text analytics (i.e., Coh-Metrix)
- Research question: Does call “cohesion” matter to markets?
  - Cohesion = relations among words, types of words, sentences and paragraphs in a document (8 dimensions)
The practical problem

• Cohmetrix Software
  • Good news:
    • Linguistically state of the art, includes lexicons (complete dictionaries), syntax, domain knowledge (i.e., Latent Semantic Analysis), rhetorical structure
  • Bad news:
    • Not open-source (can’t reverse engineer)
    • Computationally slow

• Conference call data
  • Available, “big” and dirty (~ 200,000 files)
The race

• First-year research papers → due in 4 months (i.e., 120 days)
• Scope:
  • ~ 200,000 data files
• The PhD student...... was nervous
The process - conceptually

**Harvest (dirty) files**
- Download, open, select, copy, paste, save

**Clean**
- Remove html + all non-English

**Analyze**
- Run Coh-Metrix
Project: Manual & Local Resources – Estimated Days to Completion

Days

- Harvest: 41 days
- Clean: 406 days
- Analyze: 270 days
Project: Manual – Estimated Days to Completion

Total days = 717!
(717 > 120)!
Help! Automate / Scale Processes

Harvest (dirty) files
  • Web Crawler using Stata

Clean
  • Regular expressions in Stata
  • four stage parsing strategy

Analyze
  • Vikram (Michael helping): Run on AWS cloud
Why AWS (EC2- Elastic Compute)?

• No local UKY resources to run Coh-Metrix (Windows) at scale
• AWS - platform for software testing using “clean” installs (no software conflicts & correct available tools)
• Prototype: create working machines
• Post-prototyping, create new “virtual machines” for rapid scalability and load sharing
• Cost savings - Spot Market ($) vs On Demand pricing ($$$) vs buying hardware ($$$$
• AWS $100 credit for prototyping
Analyzing files on AWS

Problem:
• Coh-Metrix software does not run in parallel
• Each file separately loaded and processed
• Processing time varies (file size + Cohmetrix analysis (metadata))

Solution:
• Knapsack problem: use one-Dimensional Bin Packing Algorithm
• Minimize number of bins (machines), process all files, equalize processing time, minimize cost
The Knapsack problem (Wikipedia)

• Given n items to put in a sack, each with a unique weight, determine the number of items to include in m sacks so that the total weight is equalized

• Here: Given 200,000 files, each with a unique processing time, determine the needed virtual machines, so that total processing time is equalized (and therefore total cost is minimized)
How to load balance 200K files across virtual machines

• Bin Packing Solution:

  • Input: – 200K+ files with varying sizes (few KB to several MB)

  • Analyze the distribution of file sizes across multiple VM’s with minimal wastage of CPU time (and money!) across virtual machines

  • Task: – Find a packing of files in equal-sized bins that minimizes the number of bins (Virtual Machines) used
Load Balancing and Bin Packing

AWS VM EC2 cluster

E.g., one virtual machine

Coh-Metrix Process

Virtual machine

Corporate Earning files

Corporate Earning files

Corporate Earning files

Corporate Earning files
Running Coh-Metrix on AWS Spot Market

• Task demands: 200,000 files can take 5 to 30 minutes to process
• Processing: running many copies of software on each machine (~ 25)
  • Specify: hardware - 32 core virtual machines
• Identify AWS zones (physical locations) to run software (minimize cost)
• Spread (binpack): Match files to virtual machines (how many machines?)
• The process:
  • Step 1: Create Virtual machines (based on prototype)
  • Step 2: Deploy machines (Map to AWS zones and binpack)
  • Step 3: Monitor Processing (Spot Market).
    • If outbid or prices changes, then bid higher and / or return to Step 2
    • Over time, learned to do this more efficiently
Results

- It worked!
- Complete results in ~ 90 days
- Cost ~ $1,000
What’s next? Additional “holistic” analyses of market information

• SEC data?
• Social media data?
• Audit