Honours Computer Science Thesis

rally, a one stop-shop for all reddit data

by

Kevin J. Eger

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

B.SC. COMPUTER SCIENCE HONOURS

in

Unit 5

(Computer Science)

THE UNIVERSITY OF BRITISH COLUMBIA

(Okanagan)

April 2016

Abstract

Reddit is *the front page of the internet*, a slogan the company has coined and rightfully lived up to. It is a website which brings together members of all communities in a similar style to a typical forum but with much more structure and a lot more traffic. Due to the open nature of reddit, it generates a large amount of traffic, averaging over 200 million unique visitors a month. With such traffic screams the demand for data analysis through a human-interpretable medium which this thesis covers. Data analysis on reddit has been done before however this thesis focuses on bringing the data gathered in to a easily consumable format. We will explore the implementation and results of querying the reddit API, generating aggregate statistics, querying large data dumps of historic reddit data with *Google BigQuery* and the use of unsupervised machine learning to draw powerful conclusions.

Table of Contents

Abstract . . . . . **ii**

Table of Contents . . . . . **iii**

List of Tables . . . . . **iv**

List of Figures . . . . . **v**

Acknowledgements . . . . . **vi**

Chapter 1: Introduction **1**
1.1 Reddit 1
1.2 Motivation 1

Chapter 2: Background **3**
2.1 Terms and Definitions 3
2.2 Reddit 3
2.2.1 History 3
2.2.2 Community 3

Chapter 3: Technical Stack **5**
3.1 Laravel 5
3.1.1 MVC 5
3.2 Storage 6
3.2.1 MySQL 6
3.2.2 BigQuery 6
3.3 phpRaw 7

Bibliography **9**

List of Tables

List of Figures

Acknowledgements

Work on this thesis was widely facilitated with help from Dr. Ramon Lawrence through weekly meetings where ideas and progress were discussed extensively. It is also important to acknowledge Dr. Jeff Andrews for his support in advising on machine learning techniques which were implemented as described later.

Chapter 1

Introduction

High level overview and motivation for developing this thesis.

1.1 Reddit

Reddit is a a news and entertainment website whose content is sustained by members of the community. Users submit text posts or direct links similar to a typical forum setting. Registered users can vote on submissions bringing order to the posts yielding an ordered online bulletin board. Furthermore, what makes Reddit unique is that content is subsectioned into different areas of interest called "subreddits". Some of the top subreddits include *movies*, *funny*, *AskReddit*, *food* and *news*. As of March 3rd, 2016 Reddit had 231,625,384 unique users a month viewing a total of 7,517,661,034 pages. The company was founded 10 years ago and has quickly become the most central place on the internet to partake in conversation or consume a wide array of content.

1.2 Motivation

For years data analytics has been used in many industries to give companies and organizations better business decisions and verification of their models and structures. Whether they are mining huge data sets, looking at specific use cases or aiming to prove or disprove a theory, companies and organizations alike aim to do one thing: identify and discover patterns, relationships and inferences that are not immediately apparent.

An early motivator for this thesis was some existing technology for Twit-

ter insights. The community-content driven nature of Twitter parallels that of Reddit. There has already been a lot of academic research and production level software released for Twitter data management, pattern identification and tracking. The existing infrastructure in the Twitter space can be largely replicated and modified to suit Reddit, an effort which this thesis focuses on starting.

Chapter 2

Background

To best understand this thesis and the work done, it is necessary to first be introduced to the relevant technologies and key terms which will be heavily referenced and built upon.

2.1 Terms and Definitions

TODO

2.2 Reddit

2.2.1 History

The company was founded by two new graduates of the *University of Virginia*, Steve Huffman and Alexis Ohanian, in June 2005 [Gua05]. After a couple years of growth, Reddit's traffic exploded and the service went viral. The creators were quick to release Reddit Gold, which offered new features and usability improvements providing the company with a primary source of income.

2.2.2 Community

Reddit thrives on its open nature and diverse content fully generated by the community [Atl14]. The demographics Reddit serves allows for a wide range of subject areas thus having the ability for smaller communities to

digest their niche content. Subreddits provide a very unique opportunity by raising attention and fostering discussion that may not be seen as mainstream and covered by other news or entertainment mediums.
Reddit as a company and as a community has been known for several philanthropic projects both short and long term. A few of notable efforts are as follows:

- Users donated $185,356 to Direct Relief for Haiti after the earthquake that struck the country in January 2010
- Reddit donates 10% of it's yearly annual ad revenue to non-profits voted upon by its users [Red14]
- Members from Reddit donated over $600,000 to DonorsChoose in support of Stephen Colbert's March to Keep Fear Alive [Don10]

Chapter 3

Technical Stack

Rally is a project that explores many different types of data access, processing techniques and display forms. Due to the nature of web applications, it is no surprise that rally is implemented with modular programming in mind. Several key components outlined below are what will allow this project to be easily continued and built on. The technical stack is broken in to components as follows.

3.1 Laravel

Laravel is a *PHP* web application framework with expressive, elegant syntax [**?**]. Laravel is designed primarily with the motive of removing the repetitive and often painful part of building trivial common tasks to a majority of web projects (ie: authentication, routing, sessions, etc.). Laravel aims to make the development process a pleasing one for the developer without sacrificing application functionality [**?**]. The accessible and powerful framework was chosen for it's existing familiarity and power to implement a project spanning many domains.

3.1.1 MVC

Laravel follows the traditional Model-View-Controller design pattern. Models interact with the database through the *Eloquent* ORM, to retrieve objects' information. Controllers handle the requests and retrieving data by leveraging the models. Views render the web pages and are returned to the user.

This intrinsic design pattern was followed tightly alongside the addition of a repository layer. As referenced in the [enter section here] section, *Rally* interacts with several external resources such as the Reddit API and Google BigQuery. These external resources house gigabytes of data thus storing them locally and accessing them through a model is counterproductive. To retain the structure of the MVC framework, a repository layer is built on top of both phpRaw and the BigQuery facade (discussed in detail in section [enter section]). This allows the convenience of a seemingly object oriented interaction. Not only does it allow for convenient method calls but also abstracts logic away from the controllers, leaving them as slim as possible. This is a vital design philosophy to web development as it modularizes code to ensure a more rigid flow and testable code-base.

3.2 Storage

Databases used to house the necessary persistent information for the application.

3.2.1 MySQL

MySQL is an open-source relational database management system (DBMS). In Laravel, it is the default DBMS largely because of it's *plug and play* nature. The only information being stored in the MySQL database is the caching layers as described in detail in the [insert section] section. The tables used are as follows:

[Insert Tables]

3.2.2 BigQuery

Querying massive datasets can not only be time consuming but expensive without the right hardware, infrastructure and software. *Google* alleviates this problem with *BigQuery*, an incredibly fast cloud-based storage platform. It is Infrastructure as a Service (IaaS) that handles all the hard work of both creating and accessing large data sets. Using the processing power of

Google's already existing infrastructure, a user can get up and running with BigQuery in a matter of minutes. BigQuery can be used via their web UI, command-line tool or the REST API using one of the many client libraries.

Five months ago, user /u/Stuck_In_the_Matrix of reddit collected all Reddit submission data from 2006 to 2015. He had effectively collected 200 million submission objects, each with score data, author, title, self_text, media tags and all the other attributes that are normally available via the Reddit API. The dataset complemented the Reddit comment corpus he released a couple months prior. When the data was initially made publically available, he released it as a torrent where developers interested in using it could download their own local copies of the data and process it according to their needs. Developers were all downloading the data for use either on their local machines or a cloud server. The problem with this is even with one of the most powerful desktop computers, loading the entire dataset into RAM was not feasible. Search times and joining (cross table) operations were expensive. Conveniently soon after the release of this torrent, one of the lead engineers of Google BigQuery, /u/fhoffa (Felipe Hoffa), uploaded the data to BigQuery and made the dataset publicly available. Each month, the dataset is updated with the latest information collected from the Reddit API.

With the convenience of BigQuery, it is now possible to query gigabytes of history Reddit data in a matter of seconds. Listed below are a couple examples of queries, their sizes and the execution time.

[Insert Examples]

[Insert Facade talk]

3.3 phpRaw

The Reddit API has several endpoints. It is through this endpoints where a client can retrieve posts specific to a subreddit, post a comment, moderate their account and all other actions that are normally available through the consumable web interface. For a single use or specific focus, calling the specific endpoints explicitly works fine but this strategy quickly fails as needs grow. Due to the wide array of endpoint calls utilized, it was

necessary to develop an API wrapper that allows convenient calls to the API. Such a wrapper already existed for Python, Java, C and a few other languages but not PHP.

An open source wrapper was discovered on GitHub but was no longer maintained, was not written to comply with the latest API security requirements (OAuth2) and was missing nearly half of the endpoints. Building on the work done on this API wrapper, a successful implementation was built and is what *Rally* utilizes and depends on for direct Reddit data access. The GitHub repository from the point at which it was forked and built on is linked in the appendix.

Listed below are a functions from phpRaw to give a feel for the wrapper. TODO - Examples

phpRaw was then modified to serve as a standalone vendor service brought in through Laravel's default dependency manager *Composer*. By extracting the wrapper to a separate module, updating and maintaining the endpoints is simple as they are changed over time.

Bibliography

[Atl14] Ama: How a weird internet thing became a mainstream delight, 2014 [cited March 3, 2016]. → pages 3

[Don10] Welcome redditors!, 2010 [cited March 3, 2016]. → pages 4

[Gua05] A new website makes it easier to sift the mountains of news content online - and learns what you like, 2005 [cited March 3, 2016]. → pages 3

[Red14] Decimating our ads revenue, 2014 [cited March 3, 2016]. → pages 4