MEM GRAPH for reddit Comment Sentiment Analysis

Project 1, Phase 1: NoSQL Storage Proof-of-Concept Proposal

October 3rd, 2018

Brought to you by
THE DATANAUTS

Julia Garbuz,
Cory Koster,
Jonathan Persgården,
and Daniel Wu

CSCI 5980 (Section 002), Fall 2018
Professor: Donald Sawyer

Table of Contents:

1 Overview

1.1 Background Information

Reddit is a social media website consisting of communities and sub-communities (“subreddits”) where people can have discussions on various topics. Contributors can make posts with text, images, gifs, and emojis in the discussion. Reddit members are not confined to one topic or discussion and can participate in as many discussion threads or sub-communities as they want. As they post, users may interact with other members by responding to comments or posts as well as voting “up” or “down” on them. The more users contribute and post, the more interactions they will have with an increasingly greater number of people.

1.2 Purpose / Business Problem

The Datanauts want to explore the relationship of a user’s sentiment in comments to a variety of fields including age of user, number of years using Reddit, which Subreddit the comment was in, and number of up/down votes on the comment.

Particularly, the goal of this project is to research, explore, and analyze these two more-involved relationships:

(1) The changes in a user’s sentiment over time spent as a Reddit user grouped by age at which user joined Reddit

(2) The effects on a user sentiment in a particular subreddit after participating in other subreddits with a differing general sentiment

1.3 Solution Overview

From the perspectives of speed, ease of addition of supplementary information, logical alignment to the current questions, and ability to handle high-velocity real-time data, the Datanuats believe that the best technology for performing this analysis would be the NoSQL graph database, **Memgraph**. However, the core data to be used for the Reddit comment sentiment analysis is currently stored in Google’s **BigQuery**, a NoSQL column-family database. Open-source messaging service, **Apache Pulsar**, will be used to extract the data from BigQuery and then load it to Memgraph. Prior to the load, **Python** will be used to transform the data, supplement it with additional information from Reddit, and perform “opinion mining” or “sentiment analysis” using the Lexicons Sentiment Analysis data.

2 Data Sources

2.1 Primary Dataset: Reddit Comment Dataset

The primary source of data for the analysis will be a real-time database of every publicly-available Reddit user comment. This data has already been collected and stored in a Google BigQuery database for others' use and analysis.

2.1.1 Key Fields

The fields we will be most interested in from this source are: `id`, `body`, `subreddit_id`, `parent_id`, `subreddit`, `ups`, `downs`, `created_utc`, `author`, `link_id`, and `name`.

The `body` field will undergo the most analysis to determine its "sentiment" using the Sentiment Analysis Lexicon described in section 2.3. The IDs such as `subreddit_id`, `parent_id`, `subreddit`, `author`, `link_id`, and `name` will be used to establish connections and relationships between comments and users. Additionally, the `author` field will be used to look up supplementary information about the user.

A complete data dictionary and data sample for the Reddit comment dataset is provided in **Appendix A**.

2.2 Supplementary Reddit User Data

Supplementary Reddit user data will be collected via Reddit's API as well as web scraping a user's Reddit page. The supplementary user information is needed to perform analysis on a user-age and account-age basis

Another dataset containing all Reddit user data has also been found but concerns arise in it possibly being a snapshot (not up-to-date) dataset and possibly not containing all the users whose comments are in the comment dataset. The Datanauts shall perform further analysis and then come to a decision but as of now have decided that Reddit's API is the safest and most dependable option.

Data samples from both Reddit's API and scraping of a user's webpage are provided in **Appendix B**.

2.2.1 Key Fields

The key fields that will be collected on each user include the date the user joined Reddit (`created_utc`, from Reddit's user information API) as well as their date of birth ("`cakeday`", visible on most user accounts) to determine user age (both at time of account creation and currently).

2.3 Sentiment and Emotion Lexicons

The dataset for the sentiment and emotion mapping is much smaller, around eight megabytes. This dataset is available as a text file and can be downloaded from the Sentiment and Emotion Lexicons website (http://sentiment.nrc.ca/lexicons-for-research/).

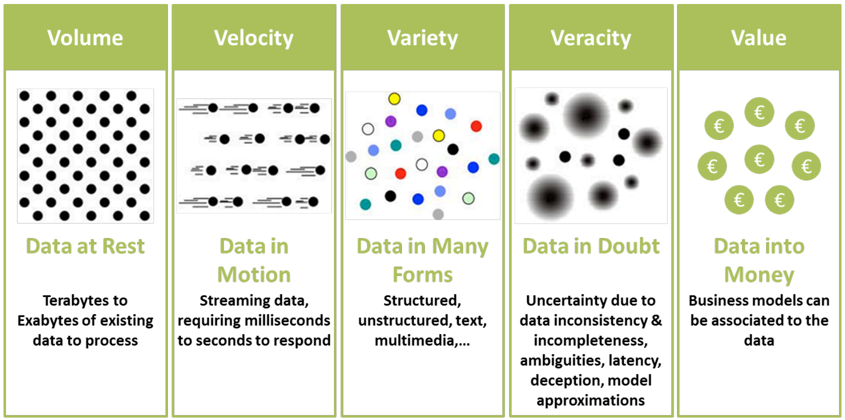
The format of the file consists of a list of words and a mapped emotion or sentiment. The layout of each line includes a term, list of synonyms, emotion or sentiment, and an indicator (0/1). The indicator signifies if the given word has an association with the given emotion or sentiment.

This dataset will be read directly from the original text file and used to process the body of each comment for sentiment analysis.

Set up instructions, a complete data dictionary, and data sample for the Sentiment and Emotion Lexicon dataset is provided in **Appendix C**.

2.4 Big Data Properties

The analysis of the comment data from Reddit can be classified as a “Big Data” problem because it meets five of the traits commonly used to identify “Big Data” problems:


Volume
Velocity
Variety
Veracity
Value
€
Data at Rest
Terabytes to Exabytes of existing data to process
Data in Motion
Streaming data, requiring milliseconds to seconds to respond
Data in Many Forms
Structured, unstructured, text, multimedia,…
Data in Doubt
Uncertainty due to data inconsistency & incompleteness, ambiguities, latency, deception, model approximations
Data into Money
Business models can be associated to the data

Adapted by a post of Michael Walker on 28 November 2012

Diagram 2.4 (a): The 5 “V”’s of Big Data

1. **Volume**: The Reddit comment dataset is currently over 250 GBs compressed. This is beyond what personal computers are capable of handling.

2. **Velocity**: The Reddit comment dataset is updated frequently with additional comment data which produces the challenge of needing to be able to handle data at a high velocity for a most up-to-date and thorough analysis.

3. **Variety**: The combination of structured user and comment information and the unstructured text fields of the comments themselves as well as variety of sources of data produces the challenge of integrating them all together for analysis.

4. **Veracity**: The Reddit comment data is collected from Reddit directly and has gone through minimal transformation to be organized in the BigQuery database. The additional Reddit user information is coming from the original source via either Reddit’s own API or by web-scraping Reddit users’ pages. Knowing the source of the data builds confidence that it will have a high signal-to-noise ratio.

5. **Value**: As a whole, the data could provide potential insight to various trends in user sentiment in relation to several other fields. Despite the challenges of handling real-time data, trends over time even without the most recent data still provide significant value.

3 Technology and System Design

3.1 Primary Storage Technology: Memgraph

Memgraph is a graph database solution which prioritizes speed, scalability and simplicity over other properties. It's built to handle real time systems and supports ACID transactions. According to its website, Memgraph supports the query language openCypher which has been developed by neo4j. Queries can be sent via command line using neo4j's command line tool or programmatically using an open source Bolt protocol driver. The Bolt protocol driver allows users to work with Memgraph in any popular programming language.

Utilizing a graph database is advantageous when trying to represent or explore deep links in data. For the use case of Reddit comments, graphical data storage allows for easy representation and traversal of the nested comment structure to examine relationships of user sentiment. Additionally, Memgraph's ability to handle high velocity data was very important to this particular use case since the comment data would require near real-time processing.

Memgraph can be deployed in a physical cluster, in the cloud, or on a virtual machine, such as Docker in this case. This allows flexibility for standing up Memgraph to run and process queries. Assuming Docker has been installed and set up, the Memgraph image can be downloaded from the Memgraph website (https://memgraph.com/docs/quick-start/). This site also contains step-by-step instructions for which commands to run in the command line tool. After installation, Memgraph is ready for receiving or querying data.

3.2 System Design

3.2.1 Overview

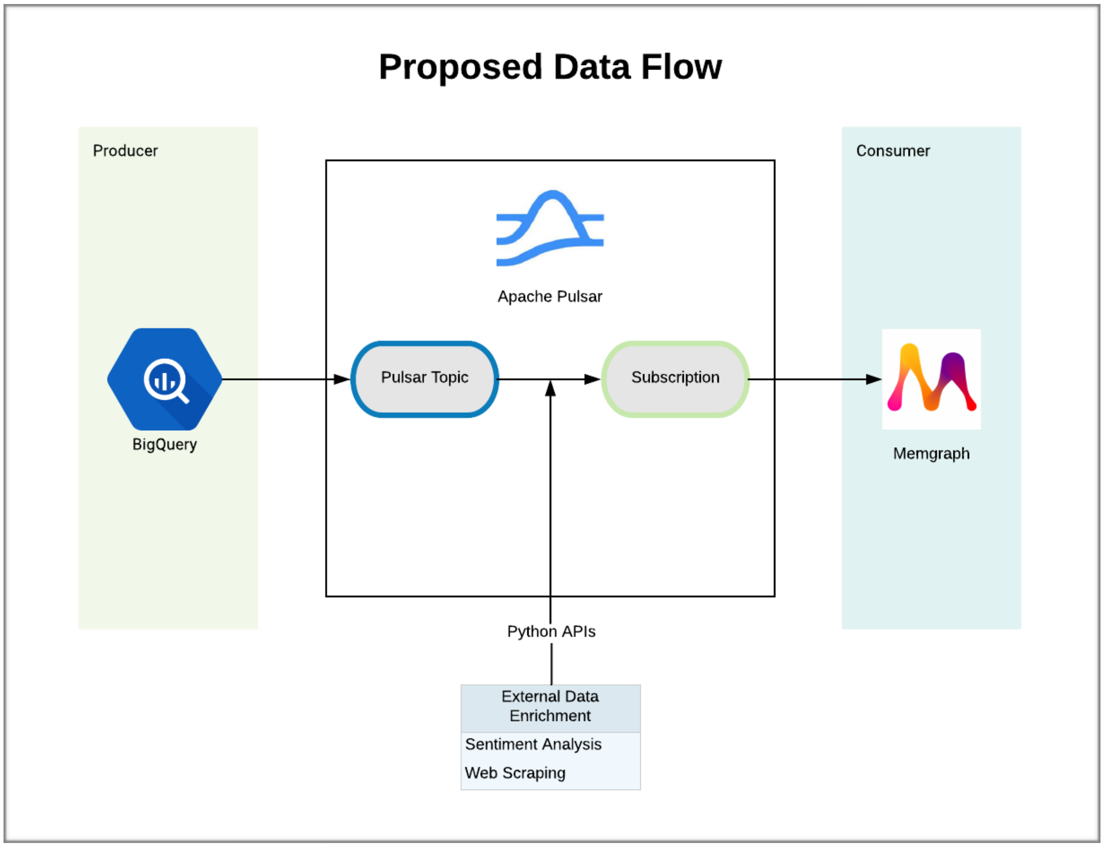


Diagram 3.2.1(a): Proposed Solution / Data-Flow Diagram

The sections below step through each component of the data flow (in order) describing the technology, why it was chosen, and how the data will moved or be transformed by that technology.

3.2.2 Google BigQuery

For this project, BigQuery will serve as the source of our data. BigQuery is a fully-managed data warehouse, provided by Google, that provides high performance and scalability. BigQuery is a columnar database and can be accessed using the Web UI, command line tools, or programmatically using the provided REST API. Additionally, BigQuery has a Rich SQL language that supports high-performance analytical queries, but we do not intend to utilize these tools.

Google provides an API for exporting data from BigQuery (https://github.com/googleapis/google-cloud-python/tree/master/bigquery). Using a `BigQueryIO.Read` transform, we will be able to read the entire table (https://cloud.google.com/dataflow/model/bigquery-io#reading-from-bigquery) and then utilizing Apache's Python client for Pulsar (http://pulsar.apache.org/docs/latest/clients/Python/) we will be able to begin streaming the data.

3.2.3 Apache Pulsar

Apache Pulsar is a solution that can be used for streaming or queueing data. It is built around the concept of producers and consumers. A producer is an application that publishes data and a consumer is an application that receives data. The data is sent through channels that are called topics. Each consumer can decide from which topics it will consume data from. This is expressed by declaring that a consumer "subscribes" to a given topic. Data in Pulsar is sent using a producer-topic-subscription-consumer model. The data-flow can be described as such: data is sent from a producer to a topic, Pulsar will then send the data from the topic to a consumers that has subscribed to that topic.

3.2.4 Python HTTP Requests and BeautifulSoup Web-Scraping

A transformational microservice will subscribe to the initial data topic. The first step in the transformation will be to supplement the data with outside information.

These data will be collected via two methods (as no source that has all of the necessary information is currently publicly available): (1) Call Reddit's publicly-available user information API (see **Appendix C**) for user account create date, and (2) Web scrape a user's public Reddit profile for their date of birth (also see **Appendix C**).

After supplementing the initial user data, user object nodes can be published to a new Pulsar topic for subscription by a loading service to Memgraph.

3.2.5 Sentiment and Emotion Analysis in Python

The comment data will then undergo sentiment and emotion analysis performed by Python against the Sentiment and Emotion Lexicon dataset described in section 2.3. Sentiment analysis is the attempt to understand what type of feelings a person expresses through a comment or post. This analysis examines the words of the comment and tries to determine whether the user had positive, negative, or neutral feelings. The analysis will be done using Python.

The comment and sentiment data will then be published to the same topic as the user information for loading into Memgraph.

3.2.6 Memgraph

Again, a microservice will subscribe to the post-transformation data topic to then finally load the data into Memgraph using the Bolt protocol driver. Diagram 3.2.6 demonstrates the proposed data model for our Memgraph instance.

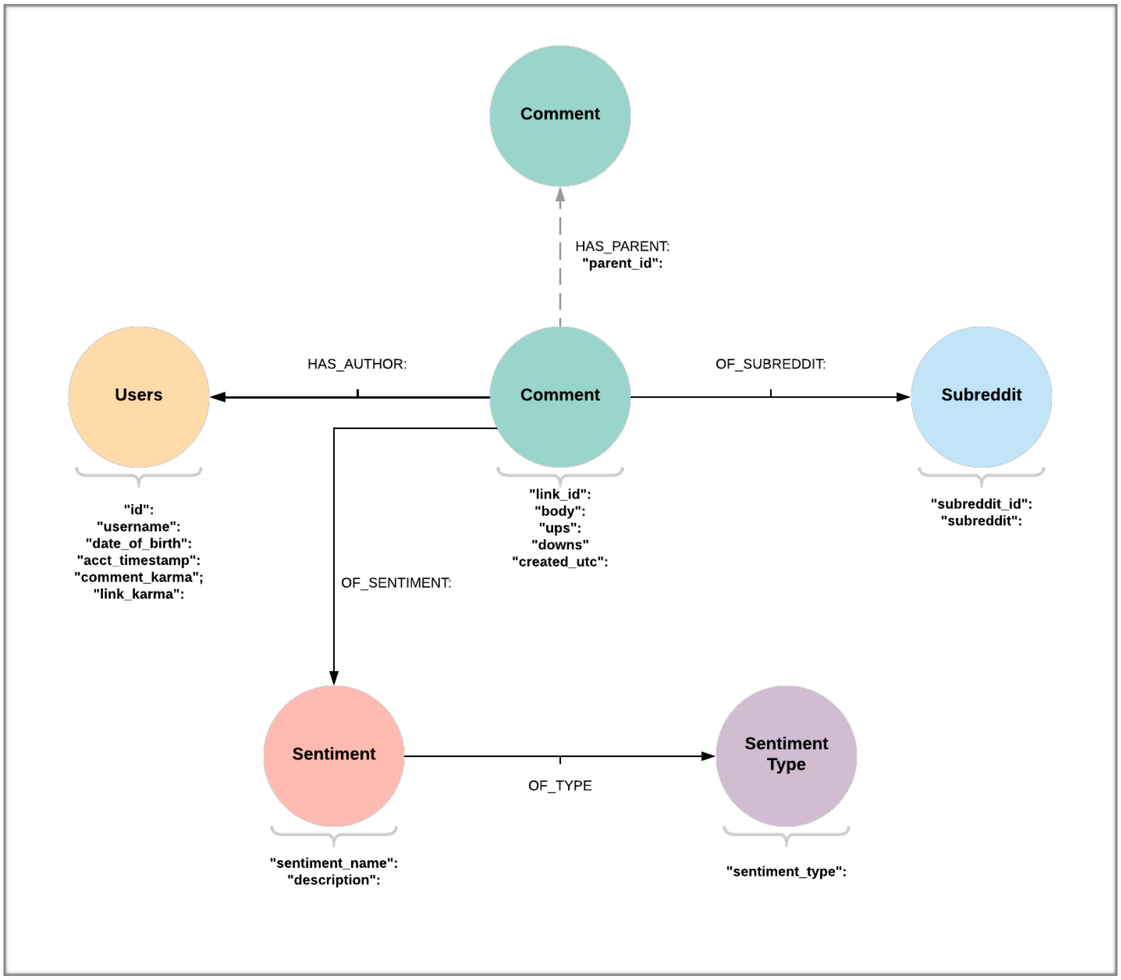


Diagram 3.2.6 (a): Memgraph Data Model

4 Analytical Questions

4.1 Easy Questions

- Minimum, maximum, and average number of comments per user
- Number of comments per sentiment
- Ratio of positively to negatively classified comments
- Number of users per age group
- Most popular subreddits (by number of comments)

4.2 Moderately Difficult Questions

- Correlation of Reddit usage to more positive or negative sentiment
- Mode of sentiment by age group
- Relationship of sentiment of comment to number of up and down votes

4.3 Challenging Questions

- The changes in a user's sentiment over time spent as a Reddit user grouped by age at which user joined Reddit
- The effects on a user sentiment in a particular subreddit after participating in other subreddits with a differing general sentiment

5 Potential Risks and Challenges

Technologies are potentially incompatible.

Given that Apache Pulsar, Google BigQuery, and MemGraph are all relatively new, it is uncertain whether the APIs, connectors, or functionality is fully fleshed out and well tested to make this project successful with the proposed technologies.

However, there exist alternative ways of accessing the database. Compressed archive dumps or alternative APIs to access the Reddit data are readily available. However these options are not ideal as they will require more ingenuity to ingest the data through alternative means.

MemGraph was released in 2017 and is currently not open source. Based on the article, “The Real-Time Distributed Enterprise Graph Database Platform | Memgraph“ (https://memgraph.com/product/), our data sources could be limited to Kafka and AWS with no ability to evaluate the source code to add our own functionality.

Furthermore, Apache Kafka is a more readily supported and mature message broker compared to Apache Pulsar. Therefore, greater support exists for Apache Kafka that could be a viable alternative if we are unsuccessful with Apache Pulsar.

Resources are limited.

Finally, if we were to extend the project to beyond live streaming Reddit data, and to look at historical Reddit data since 2015, there are over 4.6 billion comments to store and perform sentiment processing on. Our resources, both with regard to finances and time, are limited and may not be sufficient to be able to conduct a comprehensive analysis to answer or business questions.

Datasource May Not Have Expected Data.

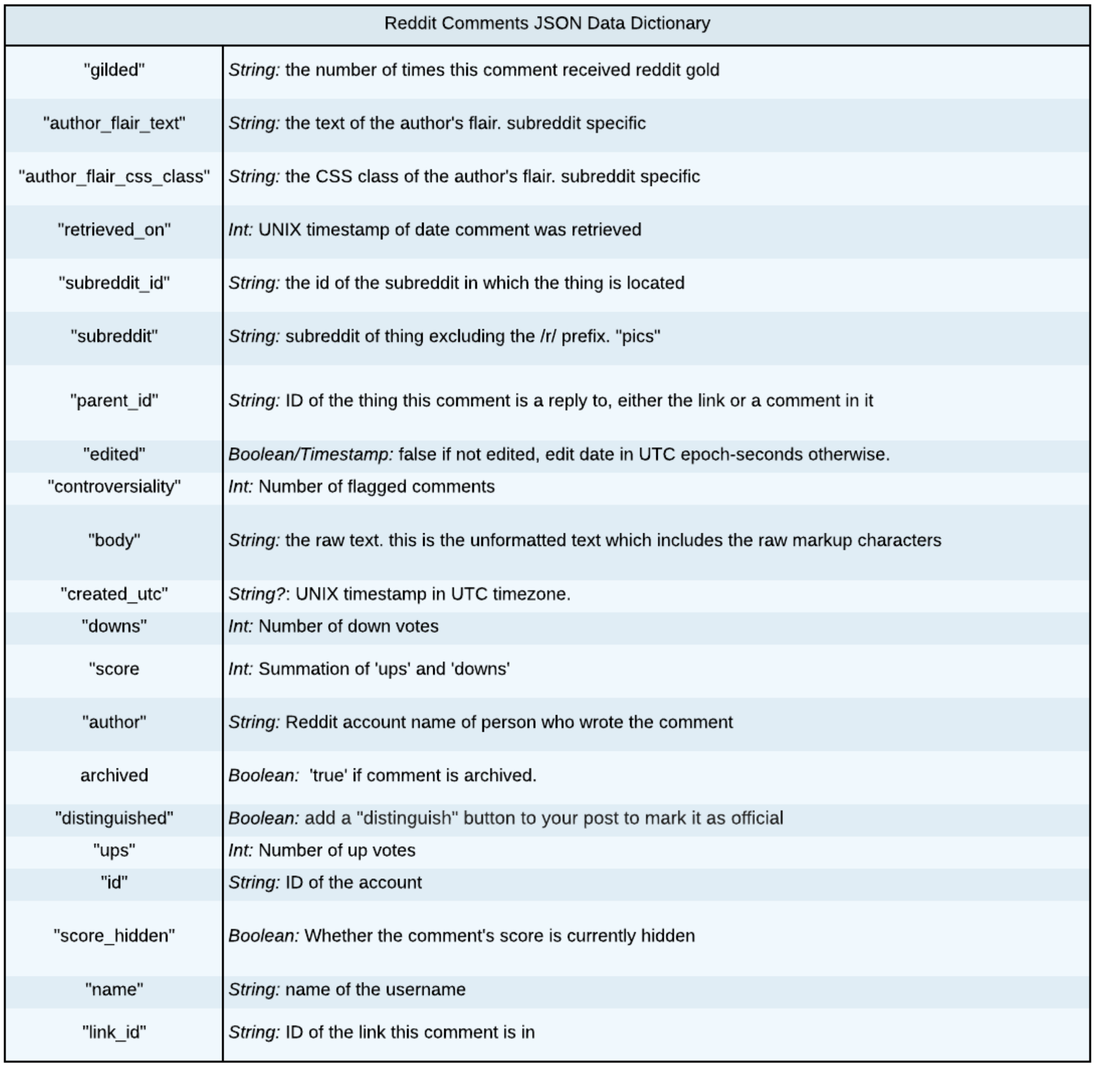
To perform thorough analysis of changes in user sentiment over time is it important that there a sufficient number of comments per user. Additionally, the potential lack of publicly available or accurately reported dates of birth could result in incorrect or far-from-truth results.

We shall try to mitigate these risks by evaluation of our “Easy Questions” prior to the more complex analyses to evaluate whether the latter will result in valuable results.

6 Appendix

Appendix A **Reddit Comment Data:** Data Dictionary and Data Sample

Data Dictionary:



Reddit Comments JSON Data Dictionary	
"gilded"	*String:* the number of times this comment received reddit gold
"author_flair_text"	*String:* the text of the author's flair. subreddit specific
"author_flair_css_class"	*String:* the CSS class of the author's flair. subreddit specific
"retrieved_on"	*Int:* UNIX timestamp of date comment was retrieved
"subreddit_id"	*String:* the id of the subreddit in which the thing is located
"subreddit"	*String:* subreddit of thing excluding the /r/ prefix. "pics"
"parent_id"	*String:* ID of the thing this comment is a reply to, either the link or a comment in it
"edited"	*Boolean/Timestamp:* false if not edited, edit date in UTC epoch-seconds otherwise.
"controversiality"	*Int:* Number of flagged comments
"body"	*String:* the raw text. this is the unformatted text which includes the raw markup characters
"created_utc"	*String?*: UNIX timestamp in UTC timezone.
"downs"	*Int:* Number of down votes
"score	*Int:* Summation of 'ups' and 'downs'
"author"	*String:* Reddit account name of person who wrote the comment
archived	*Boolean:* 'true' if comment is archived.
"distinguished"	*Boolean:* add a "distinguish" button to your post to mark it as official
"ups"	*Int:* Number of up votes
"id"	*String:* ID of the account
"score_hidden"	*Boolean:* Whether the comment's score is currently hidden
"name"	*String:* name of the username
"link_id"	*String:* ID of the link this comment is in

Data Sample:

```
{
   "gilded":0,
   "author_flair_text":"Male",
   "author_flair_css_class":"male",
   "retrieved_on":1425124228,
   "ups":3,
   "subreddit_id":"t5_2s30g",
   "edited":false,
   "controversiality":0,
   "parent_id":"t1_cnapn0k",
   "subreddit":"AskMen",
  "body":"I can't agree with passing the blame, but I'm glad to hear it's at
least helping you with the anxiety. I went the other direction and started
taking responsibility for everything. I had to realize that people make
mistakes including myself and it's gonna be alright. I don't have to be
shackled to my mistakes and I don't have to be afraid of making them. ",
   "created_utc":"1420070668",
   "downs":0,
   "score":3,
   "author":"TheDukeofEtown",
   "archived":false,
   "distinguished":null,
   "id":"cnasd6x",
   "score_hidden":false,
   "name":"t1_cnasd6x",
   "link_id":"t3_2qyhmp"
}
```

Appendix B **Supplementary Reddit User Data:** API and Web-Scrape Data Samples

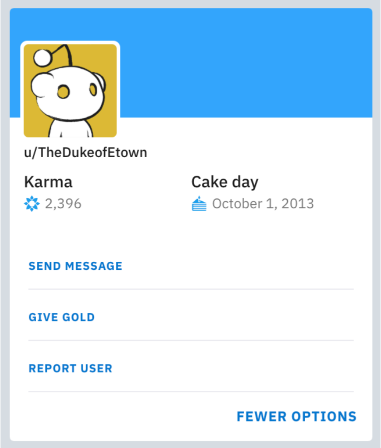
Reddit User Data API:

Accessing user data via Reddit's API is as simple as requesting data from "https://www.reddit.com/user/TheDukeofEtown/about.json" where "TheDukeofEtown" is the user's name. That API call returns the following response:

```
{
     "kind": "t2",
     "data": { "is_employee": false,
               "icon_img":    “https://www.redditstatic.com/
                              avatars/avatar_default_12_
                              DDBD37.png”,
               "pref_show_snoovatar": false,
               "name": "TheDukeofEtown",
               "is_friend": false,
               "created": 1380641637.0,
               "has_subscribed": true,
               "hide_from_robots": false,
               "created_utc": 1380612837.0,
               "link_karma": 1,
               "comment_karma": 2395,
               "is_gold": false,
               "is_mod": false,
               "verified": false,
               "subreddit": null,
               "has_verified_email": false,
               "id": “dcemf"
          }
}
```

Reddit User Webpage:

Accessing a user (for example user "TheDukeofEtown")'s Reddit page (at "https://www.reddit.com/user/TheDukeofEtown) displays the following "ID Card" with date of birth information:



u/TheDukeofEtown

Karma 2,396

Cake day October 1, 2013

Which can be “scraped” from the HTML:

```
…
<span class="s7tbhgy-3 fBffss" id="profile--id-card--highlight-
tooltip--cakeday">October 1, 2013</span>
…
```

Appendix C Sentiment and Emotion Lexicon: Set-up, Data Dictionary and Sample

Instructions:

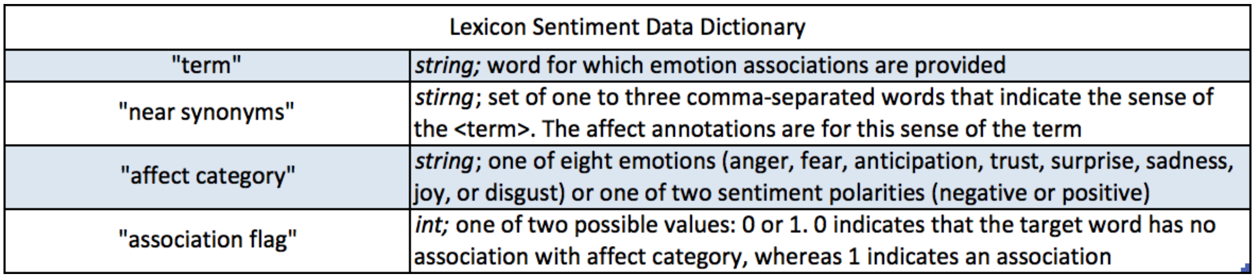
This dataset is available as a text file and can be downloaded from the Sentiment and Emotion Lexicons website (http://sentiment.nrc.ca/lexicons-for-research/).

After navigating to the homepage, the first lexicon listed under the Sentiment and Emotion Lexicons table, *NRC Word-Emotion Association Lexicon*, is the version used. Selecting this version will provide more information about the lexicon as well as the non-commercial licensed link. Accessing this license version will provide a link to download the compressed data.

Once downloaded and unzipped, the folder will contain several lexicons pertaining to different word type associations. The file is located in the folder *NRC-Sentiment-Emotion-Lexicon-v0.92*. The file is named *NRC-Emotion-Lexicon-Senselevel-v0.92.txt*.

The full file path is: /NRC-Sentiment-Emotion-Lexicons/NRC-Emotion-Lexicon-v0.92/NRC-Emotion-Lexicon-Senselevel-v0.92.txt

Data Dictionary:



Lexicon Sentiment Data Dictionary	
"term"	*string;* word for which emotion associations are provided
"near synonyms"	*stirng*; set of one to three comma-separated words that indicate the sense of the <term>. The affect annotations are for this sense of the term
"affect category"	*string*; one of eight emotions (anger, fear, anticipation, trust, surprise, sadness, joy, or disgust) or one of two sentiment polarities (negative or positive)
"association flag"	*int;* one of two possible values: 0 or 1. 0 indicates that the target word has no association with affect category, whereas 1 indicates an association

Data Sample:

```
conceit--vanity, assurance, airs       positive      0
conceit--vanity, assurance, airs       negative      1
```