Project Presentations - 2

CMSC 498J: Social Media Computing

Department of Computer Science
University of Maryland
Spring 2016

Hadi Amiri
hadi@umd.edu
Project Titles

- **G8. Trends of Trendy Words**
  - David, Kevin, Elias

- **G9. Comparing Words vs Movie Ratings**
  - Orest, Hadi, Nicholas

- **G11. Finding Trending Topics**
  - Sam, Yoshi, Halley

- **G12. Movie Lovers**
  - Sean, David, Fanglun

- **G13. Relationship Analysis between Twitter Users**
  - Daniel, Daniel, Connor
Trends of Trendy Words

Kevin Jia, Elias Gonzalez, David Kiernicki
Problem

• What types of words trend and for how long? What causes it?
  • Nouns?
  • Company slogans?
  • Celebrity endorsements?

• Possible Applications:
  • Marketing
  • Human behavior
Approach

- Looked at hashtags only
  - Easier that tracking specific words in tweets
  - Don’t need to know much about the context of the hashtags
  - 70% of people use hashtags (Social Times)

- 10 Days of Data
  - Theoretically able to apply findings to any 10 day span throughout year

- Categories of tweets
  - Tracked hashtag frequency over 10 day period
  - Observed similarities in trendiness
Data Collection

- Twitter Stream Grab: https://archive.org/details/twitterstream
  - “Spritzer” stream: 1% of all tweets
- July 2015
  - 46 GB
  - First 10 days
Getting Tweets

- Analyzed first 10 days of July 2015
- Collected all hashtags
- Analyzed all hashtags used more than 10,000 times over the first 10 days
  - Likely more than a small community tweeting about it
What’s Considered Trending

• No accepted formal definition
• Hashtags with frequency greater than 2,000 per day

  1% of twitter data used * 2,000 occurrences per day
  = 200,000 estimated occurrences over all of Twitter

• Likely more than just a community tweeting about it
Types of Trends

- **Short-term trending but recurring**
  - Trends for maximum of 2 days
  - Occurs at least once per year
- **Short-term but likely won’t trend again**
  - Trends for a maximum 2 days
  - Caused by a random event
  - Unlikely to trend again in the future
- **Long-term trending**
  - Trends greater than 2 days
  - Various causes
Split into Categories

• Application triggered
• Celebrities
• Television/media
• Nouns/verbs
• Current Events
• Spam
• Slang
• User Challenges
Short Term Trends
Short-term Recurring

Characteristics:
• Becomes trending over a short period of time, several times
Slang

Example:
- #tbt
  - Throwback Thursday

Characteristics:
- Trends once a week
  - Every Thursday
- Encourages Participation
  - People share their own throwback images and tweets
Current Events

Characteristics:
- Event occurs every year
- Trends only during a two day period

Examples:
- #Wimbledon
- #4thofjuly
- #sdcc
Short-term Non-recurring

Characteristics:

• Goes from trending to non-trending within the span of 2 days.
• Unlikely to become trending again
Celebrities

Characteristics:
• Tied to a recent event
• Unlikely to trend again

Examples:
• #followmejohnson
• #action1D
Long Term Trends
Nouns/Verbs and Common Slang

Characteristics:
• Daily life
  • Listen to music everyday
• Relatable to people 30 years or younger

Examples:
• #music
• #deals
Spam

Characteristics
• Purpose is often to solicit follows
• Some increases due to days of the week
  • Follow Friday

Examples:
• #followme
• #ff
Television/Media

Characteristics:
• International
• Focus on hashtags
  • Teen Choice Awards tweets a list of hashtags to use
  • Must include hashtag in order for vote to count
• Shows & awards for teenagers

Examples:
• #kcamexico
• #teenchoice
Challenges

Characteristics:
- Sudden spikes
- More consistent dropoffs
- Promoted social media celebrity Em Ford
- Requires user interaction
Comparison

**Long Term**
- Many revolve around people under 30 years old
- Relate to things that many people experience
  - Disregarding spam
- International contribution

**Short Term**
- Random events
- Days of the week/year, holidays
- One hit wonders
Undetermined Category

- Apps could be automatically posting
- Not necessarily user triggered
- Hard to determine if there’s a cause for trending or just an automated part of a user’s daily routine.
if hashtag gets a frequency of 2,000 for 2 consecutive days {
  if hashtag is a one word noun or verb {
    long term
  }
  if hashtag is slang {
    if slang is not time sensitive and refers to a daily activity {
      long term
    }
  }
}

if hashtag includes "follow" {
  long term
}

if hashtag relates to Television/Media {
  if television show requires voting {
    long term
  } else if hashtag refers to shows/actor in awards show {
    long term
  }
}

if hashtag is a challenge {
  if challenge requires user social media interaction {
    long term
  }
}
Algorithm

if 1 <= trend <= 2 days {
    if event is on a cycle {
        short term recurring
    } else {
        short term nonrecurring
    }
} else {
    if all above fails {
        no trend
    }
}
Accuracy

• Tested last 10 days of July 2015
• Algorithm correct if hashtag returned correct category
• Got 60 out of 86 predictions correct (~70%)

<table>
<thead>
<tr>
<th>Hashtag</th>
<th>Prediction</th>
<th>Actual</th>
<th>Correct?</th>
</tr>
</thead>
<tbody>
<tr>
<td>#5yearsofone_direction</td>
<td>stn</td>
<td>stn</td>
<td>yes</td>
</tr>
<tr>
<td>#android</td>
<td>It</td>
<td>It</td>
<td>yes</td>
</tr>
<tr>
<td>#androidgames</td>
<td>no trend</td>
<td>It</td>
<td>no</td>
</tr>
<tr>
<td>#anitta</td>
<td>stn</td>
<td>stn</td>
<td>yes</td>
</tr>
</tbody>
</table>
Reasons People Use Social Media

• Largest age group of Twitter users are 25-34 years old (comScore)
• ~15 reasons people use social media, which fall under three categories (Heinonen). Of these categories, the ones that involve active participation on twitter are:
  • Information
    • Sharing and accessing opinions, reviews, and ratings
  • Entertainment
    • Self-expression
  • Social
    • Collaborative experiencing
    • Belonging and bonding
    • Staying in touch
    • Social networking
Issues

• Ambiguity of the hashtags
  • #rt
    • Retweet
    • Russia Today
  • #ff
    • Flashback Friday
    • Follow Friday

• Have to analyze content of tweets along with hashtag
• Size of data
  • Hard to analyze more than a few weeks given time constraint
• Determining how long to look at data
  • Some trend at end of 10 days
Future expansion on idea

• Given the tools needed
  • Use twitter crawler and follow tweets in real time
  • Track tweets over months rather than days
  • Better pinpoint the causes of the tweets

• Expand concept and algorithm to other social media:
  • Instagram uses hashtags
  • Facebook has topics and hashtags

• Machine learning
  • More concise algorithm
  • Build a real classifier
  • Better classifications
Predict Movie Rating by Scripts

By Orest Pankiw, Hadi Khalil, Nick Whims
2014 Motion Picture Market Statistics

U.S./Canada Box Office (US$ Billions)

Source: Rentrak Corporation – Box Office Essentials (Total), MPAA (3D)

<table>
<thead>
<tr>
<th>Year</th>
<th>3D Box Office</th>
<th>Non-3D Box Office</th>
</tr>
</thead>
<tbody>
<tr>
<td>2005</td>
<td>$8.8</td>
<td></td>
</tr>
<tr>
<td>2006</td>
<td>$9.2</td>
<td></td>
</tr>
<tr>
<td>2007</td>
<td>$9.6</td>
<td></td>
</tr>
<tr>
<td>2008</td>
<td>$9.6</td>
<td></td>
</tr>
<tr>
<td>2009</td>
<td>$10.6</td>
<td></td>
</tr>
<tr>
<td>2010</td>
<td>$10.6</td>
<td></td>
</tr>
<tr>
<td>2011</td>
<td>$10.2</td>
<td></td>
</tr>
<tr>
<td>2012</td>
<td>$10.8</td>
<td></td>
</tr>
<tr>
<td>2013</td>
<td>$10.9</td>
<td></td>
</tr>
<tr>
<td>2014</td>
<td>$10.4</td>
<td></td>
</tr>
</tbody>
</table>

- 3D Box Office percentages: 1%, 1%, 2%, 10%, 21%, 18%, 17%, 16%, 14%
<table>
<thead>
<tr>
<th>Rank</th>
<th>Title</th>
<th>Distributor</th>
<th>Box Office (USD MM)</th>
<th>Rating</th>
<th>3D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Guardians Of The Galaxy*</td>
<td>Disney</td>
<td>$332.9</td>
<td>PG-13</td>
<td>✓</td>
</tr>
<tr>
<td>2</td>
<td>The Hunger Games: Mockingjay, Part 1*</td>
<td>Lionsgate</td>
<td>313.3</td>
<td>PG-13</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Captain America: The Winter Soldier</td>
<td>Disney</td>
<td>259.8</td>
<td>PG-13</td>
<td>✓</td>
</tr>
<tr>
<td>4</td>
<td>The Lego Movie</td>
<td>Warner Bros.</td>
<td>257.8</td>
<td>PG</td>
<td>✓</td>
</tr>
<tr>
<td>5</td>
<td>Transformers: Age Of Extinction</td>
<td>Paramount</td>
<td>245.4</td>
<td>PG-13</td>
<td>✓</td>
</tr>
<tr>
<td>6</td>
<td>Maleficent</td>
<td>Disney</td>
<td>241.4</td>
<td>PG</td>
<td>✓</td>
</tr>
<tr>
<td>7</td>
<td>X-Men: Days Of Future Past</td>
<td>20th Century Fox</td>
<td>233.9</td>
<td>PG-13</td>
<td>✓</td>
</tr>
<tr>
<td>8</td>
<td>Dawn Of The Planet Of The Apes</td>
<td>20th Century Fox</td>
<td>208.5</td>
<td>PG-13</td>
<td>✓</td>
</tr>
<tr>
<td>9</td>
<td>Big Hero 6*</td>
<td>Disney</td>
<td>204.6</td>
<td>PG</td>
<td>✓</td>
</tr>
<tr>
<td>10</td>
<td>The Amazing Spider-Man 2</td>
<td>Sony</td>
<td>202.9</td>
<td>PG-13</td>
<td>✓</td>
</tr>
<tr>
<td>11</td>
<td>Godzilla</td>
<td>Warner Bros.</td>
<td>200.7</td>
<td>PG-13</td>
<td>✓</td>
</tr>
<tr>
<td>12</td>
<td>22 Jump Street</td>
<td>Sony</td>
<td>191.7</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Teenage Mutant Ninja Turtles</td>
<td>Paramount</td>
<td>191.2</td>
<td>PG-13</td>
<td>✓</td>
</tr>
<tr>
<td>14</td>
<td>The Hobbit: The Battle Of The Five Armies*</td>
<td>Warner Bros.</td>
<td>189.5</td>
<td>PG-13</td>
<td>✓</td>
</tr>
<tr>
<td>15</td>
<td>Interstellar*</td>
<td>Paramount</td>
<td>179.4</td>
<td>PG-13</td>
<td>✓</td>
</tr>
<tr>
<td>16</td>
<td>How To Train Your Dragon 2</td>
<td>20th Century Fox</td>
<td>177.0</td>
<td>PG</td>
<td>✓</td>
</tr>
<tr>
<td>17</td>
<td>Gone Girl*</td>
<td>20th Century Fox</td>
<td>166.2</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Divergent</td>
<td>Lionsgate</td>
<td>150.9</td>
<td>PG-13</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Neighbors</td>
<td>Universal</td>
<td>150.2</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Frozen</td>
<td>Disney</td>
<td>137.6</td>
<td>PG</td>
<td>✓</td>
</tr>
<tr>
<td>21</td>
<td>Ride Along</td>
<td>Universal</td>
<td>134.9</td>
<td>PG-13</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Rio 2</td>
<td>20th Century Fox</td>
<td>131.5</td>
<td>G</td>
<td>✓</td>
</tr>
<tr>
<td>23</td>
<td>Lucy</td>
<td>Universal</td>
<td>126.7</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Lone Survivor</td>
<td>Universal</td>
<td>124.9</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>The Fault In Our Stars</td>
<td>20th Century Fox</td>
<td>124.9</td>
<td>PG-13</td>
<td></td>
</tr>
</tbody>
</table>

*Film still in theaters in 2015; total reflects box office earned from January 1 – December 31, 2014
# 2015 DOMESTIC GROSSES

Total Grosses of all Movies Released in 2015

#1-100 - #101-200 - #201-300 - #301-400 - #401-500 - #501-600 - #601-699

<table>
<thead>
<tr>
<th>Rank</th>
<th>Movie Title (click to view)</th>
<th>Studio</th>
<th>Total Gross / Theaters</th>
<th>Opening / Theaters</th>
<th>Open</th>
<th>Close</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Star Wars: The Force Awakens</td>
<td>BV</td>
<td>$936,121,508</td>
<td>4,134</td>
<td>12/18</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>Jurassic World</td>
<td>Uni.</td>
<td>$652,270,625</td>
<td>4,291</td>
<td>6/12</td>
<td>11/19</td>
</tr>
<tr>
<td>3</td>
<td>Avengers: Age of Ultron</td>
<td>BV</td>
<td>$459,005,868</td>
<td>4,276</td>
<td>5/1</td>
<td>10/8</td>
</tr>
<tr>
<td>4</td>
<td>Inside Out</td>
<td>BV</td>
<td>$356,461,711</td>
<td>4,158</td>
<td>6/19</td>
<td>12/10</td>
</tr>
<tr>
<td>5</td>
<td>Furious 7</td>
<td>Uni.</td>
<td>$353,007,020</td>
<td>4,022</td>
<td>4/3</td>
<td>7/24</td>
</tr>
<tr>
<td>6</td>
<td>Minions</td>
<td>Uni.</td>
<td>$336,045,770</td>
<td>4,311</td>
<td>7/10</td>
<td>12/17</td>
</tr>
<tr>
<td>7</td>
<td>The Hunger Games: Mockingjay · Part 2</td>
<td>LGF</td>
<td>$281,723,902</td>
<td>4,175</td>
<td>11/20</td>
<td>2/25</td>
</tr>
<tr>
<td>8</td>
<td>The Martian</td>
<td>Fox</td>
<td>$228,433,663</td>
<td>3,854</td>
<td>10/2</td>
<td>3/17</td>
</tr>
<tr>
<td>10</td>
<td>Spectre</td>
<td>Sony</td>
<td>$200,074,609</td>
<td>3,929</td>
<td>11/6</td>
<td>4/7</td>
</tr>
<tr>
<td>11</td>
<td>Mission: Impossible · Rogue Nation</td>
<td>Par.</td>
<td>$195,042,377</td>
<td>3,988</td>
<td>7/31</td>
<td>10/29</td>
</tr>
<tr>
<td>12</td>
<td>Pitch Perfect 2</td>
<td>Uni.</td>
<td>$184,296,230</td>
<td>3,660</td>
<td>5/15</td>
<td>7/30</td>
</tr>
<tr>
<td>13</td>
<td>The Revenant</td>
<td>Fox</td>
<td>$183,470,242</td>
<td>3,711</td>
<td>12/25</td>
<td>-</td>
</tr>
<tr>
<td>14</td>
<td>Ant-Man</td>
<td>BV</td>
<td>$180,202,163</td>
<td>3,888</td>
<td>12/17</td>
<td>12/17</td>
</tr>
<tr>
<td>16</td>
<td>Hotel Transylvania 2</td>
<td>Sony</td>
<td>$165,700,110</td>
<td>3,786</td>
<td>9/25</td>
<td>3/3</td>
</tr>
<tr>
<td>17</td>
<td>Fifty Shades of Grey</td>
<td>Uni.</td>
<td>$166,167,230</td>
<td>3,655</td>
<td>2/13</td>
<td>5/7</td>
</tr>
<tr>
<td>18</td>
<td>The SpongeBob Movie: Sponge Out of Water</td>
<td>Par.</td>
<td>$162,994,032</td>
<td>3,680</td>
<td>2/6</td>
<td>5/28</td>
</tr>
<tr>
<td>19</td>
<td>Straight Outta Compton</td>
<td>Uni.</td>
<td>$161,197,785</td>
<td>3,142</td>
<td>8/14</td>
<td>10/22</td>
</tr>
<tr>
<td>20</td>
<td>San Andreas</td>
<td>WB</td>
<td>$155,190,832</td>
<td>3,812</td>
<td>5/29</td>
<td>10/15</td>
</tr>
<tr>
<td>21</td>
<td>Mad Max: Fury Road</td>
<td>WB</td>
<td>$153,636,354</td>
<td>3,722</td>
<td>5/15</td>
<td>9/24</td>
</tr>
<tr>
<td>22</td>
<td>Daddy's Home</td>
<td>Par.</td>
<td>$150,357,137</td>
<td>3,483</td>
<td>12/25</td>
<td>4/7</td>
</tr>
<tr>
<td>23</td>
<td>The Divergent Series: Insurgent</td>
<td>LG/S</td>
<td>$130,179,072</td>
<td>3,875</td>
<td>3/20</td>
<td>7/9</td>
</tr>
<tr>
<td>24</td>
<td>The Peanuts Movie</td>
<td>Fox</td>
<td>$130,178,411</td>
<td>3,902</td>
<td>11/6</td>
<td>3/31</td>
</tr>
</tbody>
</table>
Can We Guarantee $$s??
Our Approach!
Methodology Outline

1. Parse movie titles into readable format
2. Api call to omdb.api
3. Create vw input file
4. Scrub vw file
5. Learn different classifiers based on different data
OMDB API

Response:

```json
{"Title":"The Avengers","Year":"2012","Rated":"PG-13","Released":"04 May 2012","Runtime":"143 min","Genre":"Action, Adventure, Sci-Fi","Director":"Joss Whedon","Writer":"Joss Whedon (screenplay), Zak Penn (story), Joss Whedon (story),"; "Actors":"Robert Downey Jr., Chris Evans, Mark Ruffalo, Chris Hemsworth", "Plot":"Earth's mightiest heroes must come together and learn to fight as a team if they are to stop the mischievous Loki and his alien army from enslaving humanity,"; "Language":"English, Russian","Country":"USA"; "Awards":"Nominated for 1 Oscar, Another 30 wins & 77 nominations;","Poster":"http://ia.media-imdb.com/images/MV5BMTI1MTU4N15BMl5BanBnXkFtZTcwODY0Nw@@_V1_SX300.jpg","Metascore":"69","imdbRating":"8.1","imdbVotes":"960,059","imdbID":"tt0848228","Type":"movie","Response":"True"}
```

Request:

```
http://www.omdbapi.com/?t=toy+story+3&y=&plot=short&r=json
```
Vowpal Wabbit

```perl
#!/usr/bin/perl -w
my $N = 1000;
for ($i = 1; $i <= $N; $i++) {
    my $a = rand(1);
    my $b = rand(1);
    my $c = rand(1);
    my $d = rand(1);
    my $e = rand(1);
    my $y = $a + 2*$b + 3*$c + 4*$d + 5*$e;
    printf "%g | a:%g b:%g c:%g d:%g e:%g\n", $y, $a, $b, $c, $d, $e;
}
```

Step 2) We run the script and save its output in a training-set:

```
$ generate-trainset.pl > abcde.train
```

Step 3) Running vw-varinfo on the training-set we get:

```
$ vw-varinfo abcde.train
FeatureName HashVal MinVal MaxVal Weight RelScore
^e 180798 0.00 1.00 +5.0000 100.00%
^d 193030 0.00 1.00 +4.0000 80.00%
^c 140873 0.00 1.00 +3.0000 60.00%
^b 244212 0.00 1.00 +2.0000 40.00%
^a 24414 0.00 1.00 +1.0000 20.00%
Constant 116050 0.00 0.00 +0.0000 0.00%
```
Formatting for Vowpal Wabbit

---


Scrubbing

Word does not exist at least 5 times
Word does not exist in at least 5 scripts
Script contains less than 50 words
Word is considered “bad”
What makes a good word?

At least three letters

Doesn’t appear in every file

   Stopwords

   Numbers

Not a character name

No spaces

Not punctuation or formatting
Results
<table>
<thead>
<tr>
<th>FeatureName</th>
<th>HashVal</th>
<th>MinVal</th>
<th>MaxVal</th>
<th>Weight</th>
<th>RelScore</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;heart&quot;</td>
<td>60372</td>
<td>0.00</td>
<td>6.00</td>
<td>+0.0127</td>
<td>100.00%</td>
</tr>
<tr>
<td>&quot;hands&quot;</td>
<td>189726</td>
<td>0.00</td>
<td>11.00</td>
<td>+0.0108</td>
<td>85.57%</td>
</tr>
<tr>
<td>&quot;inside&quot;</td>
<td>234742</td>
<td>0.00</td>
<td>9.00</td>
<td>+0.0108</td>
<td>85.50%</td>
</tr>
<tr>
<td>&quot;talk&quot;</td>
<td>72884</td>
<td>0.00</td>
<td>8.00</td>
<td>+0.0093</td>
<td>73.75%</td>
</tr>
<tr>
<td>&quot;ready&quot;</td>
<td>248193</td>
<td>0.00</td>
<td>9.00</td>
<td>+0.0091</td>
<td>71.42%</td>
</tr>
<tr>
<td>&quot;day&quot;</td>
<td>146201</td>
<td>0.00</td>
<td>11.00</td>
<td>+0.0089</td>
<td>70.13%</td>
</tr>
<tr>
<td>&quot;seen&quot;</td>
<td>245569</td>
<td>0.00</td>
<td>7.00</td>
<td>+0.0088</td>
<td>69.91%</td>
</tr>
<tr>
<td>&quot;might&quot;</td>
<td>22896</td>
<td>0.00</td>
<td>7.00</td>
<td>+0.0078</td>
<td>61.88%</td>
</tr>
<tr>
<td>&quot;fire&quot;</td>
<td>21070</td>
<td>0.00</td>
<td>11.00</td>
<td>+0.0068</td>
<td>55.68%</td>
</tr>
<tr>
<td>&quot;place&quot;</td>
<td>205054</td>
<td>0.00</td>
<td>9.00</td>
<td>+0.0077</td>
<td>60.68%</td>
</tr>
<tr>
<td>&quot;coming&quot;</td>
<td>161197</td>
<td>0.00</td>
<td>11.00</td>
<td>+0.0073</td>
<td>57.50%</td>
</tr>
<tr>
<td>&quot;away&quot;</td>
<td>10902</td>
<td>0.00</td>
<td>21.00</td>
<td>+0.0069</td>
<td>54.23%</td>
</tr>
<tr>
<td>&quot;wow&quot;</td>
<td>59116</td>
<td>0.00</td>
<td>11.00</td>
<td>+0.0067</td>
<td>54.06%</td>
</tr>
<tr>
<td>&quot;looking&quot;</td>
<td>217318</td>
<td>0.00</td>
<td>11.00</td>
<td>+0.0066</td>
<td>52.17%</td>
</tr>
<tr>
<td>&quot;found&quot;</td>
<td>45875</td>
<td>0.00</td>
<td>7.00</td>
<td>+0.0064</td>
<td>50.57%</td>
</tr>
<tr>
<td>&quot;old&quot;</td>
<td>205880</td>
<td>0.00</td>
<td>11.00</td>
<td>+0.0062</td>
<td>48.85%</td>
</tr>
<tr>
<td>&quot;die&quot;</td>
<td>76282</td>
<td>0.00</td>
<td>14.00</td>
<td>+0.0058</td>
<td>45.96%</td>
</tr>
<tr>
<td>&quot;sorry&quot;</td>
<td>83536</td>
<td>0.00</td>
<td>13.00</td>
<td>+0.0058</td>
<td>45.52%</td>
</tr>
<tr>
<td>&quot;friends&quot;</td>
<td>51027</td>
<td>0.00</td>
<td>7.00</td>
<td>+0.0057</td>
<td>44.97%</td>
</tr>
<tr>
<td>&quot;new&quot;</td>
<td>137063</td>
<td>0.00</td>
<td>17.00</td>
<td>+0.0057</td>
<td>44.92%</td>
</tr>
<tr>
<td>&quot;kill&quot;</td>
<td>52295</td>
<td>0.00</td>
<td>7.00</td>
<td>+0.0056</td>
<td>44.27%</td>
</tr>
<tr>
<td>&quot;try&quot;</td>
<td>64551</td>
<td>0.00</td>
<td>11.00</td>
<td>+0.0056</td>
<td>43.80%</td>
</tr>
<tr>
<td>&quot;gets&quot;</td>
<td>185662</td>
<td>0.00</td>
<td>13.00</td>
<td>+0.0054</td>
<td>42.76%</td>
</tr>
<tr>
<td>&quot;move&quot;</td>
<td>201749</td>
<td>0.00</td>
<td>13.00</td>
<td>+0.0051</td>
<td>40.05%</td>
</tr>
<tr>
<td>&quot;door&quot;</td>
<td>246777</td>
<td>0.00</td>
<td>18.00</td>
<td>+0.0050</td>
<td>39.74%</td>
</tr>
<tr>
<td>&quot;let&quot;</td>
<td>86246</td>
<td>0.00</td>
<td>15.00</td>
<td>+0.0050</td>
<td>39.66%</td>
</tr>
<tr>
<td>&quot;give&quot;</td>
<td>175713</td>
<td>0.00</td>
<td>22.00</td>
<td>+0.0047</td>
<td>37.23%</td>
</tr>
<tr>
<td>&quot;kid&quot;</td>
<td>189816</td>
<td>0.00</td>
<td>18.00</td>
<td>+0.0046</td>
<td>36.61%</td>
</tr>
<tr>
<td>&quot;nothing&quot;</td>
<td>5736</td>
<td>0.00</td>
<td>11.00</td>
<td>+0.0046</td>
<td>36.56%</td>
</tr>
<tr>
<td>&quot;sure&quot;</td>
<td>99728</td>
<td>0.00</td>
<td>13.00</td>
<td>+0.0046</td>
<td>36.22%</td>
</tr>
<tr>
<td>&quot;need&quot;</td>
<td>14630</td>
<td>0.00</td>
<td>21.00</td>
<td>+0.0046</td>
<td>35.99%</td>
</tr>
<tr>
<td>&quot;boy&quot;</td>
<td>101798</td>
<td>0.00</td>
<td>13.00</td>
<td>+0.0045</td>
<td>35.65%</td>
</tr>
<tr>
<td>&quot;show&quot;</td>
<td>227220</td>
<td>0.00</td>
<td>11.00</td>
<td>+0.0045</td>
<td>35.39%</td>
</tr>
<tr>
<td>&quot;way&quot;</td>
<td>42855</td>
<td>0.00</td>
<td>22.00</td>
<td>+0.0044</td>
<td>34.46%</td>
</tr>
<tr>
<td>&quot;voice&quot;</td>
<td>167796</td>
<td>0.00</td>
<td>10.00</td>
<td>+0.0044</td>
<td>34.37%</td>
</tr>
<tr>
<td>&quot;food&quot;</td>
<td>99446</td>
<td>0.00</td>
<td>8.00</td>
<td>+0.0043</td>
<td>34.06%</td>
</tr>
<tr>
<td>&quot;great&quot;</td>
<td>84380</td>
<td>0.00</td>
<td>11.00</td>
<td>+0.0043</td>
<td>33.56%</td>
</tr>
<tr>
<td>&quot;made&quot;</td>
<td>250513</td>
<td>0.00</td>
<td>9.00</td>
<td>+0.0042</td>
<td>33.18%</td>
</tr>
<tr>
<td>&quot;makes&quot;</td>
<td>61960</td>
<td>0.00</td>
<td>7.00</td>
<td>+0.0042</td>
<td>33.01%</td>
</tr>
<tr>
<td>&quot;hand&quot;</td>
<td>156841</td>
<td>0.00</td>
<td>15.00</td>
<td>+0.0041</td>
<td>32.70%</td>
</tr>
<tr>
<td>&quot;stay&quot;</td>
<td>131198</td>
<td>0.00</td>
<td>15.00</td>
<td>+0.0041</td>
<td>32.46%</td>
</tr>
<tr>
<td>&quot;remember&quot;</td>
<td>116397</td>
<td>0.00</td>
<td>13.00</td>
<td>+0.0041</td>
<td>32.44%</td>
</tr>
<tr>
<td>&quot;first&quot;</td>
<td>59416</td>
<td>0.00</td>
<td>16.00</td>
<td>+0.0041</td>
<td>32.33%</td>
</tr>
</tbody>
</table>

---

**Animation**

**SciFi**
Comparison Runs
(multiclass)

<table>
<thead>
<tr>
<th>loss</th>
<th>example</th>
<th>current</th>
<th>current</th>
<th>current</th>
<th>current</th>
<th>weight</th>
<th>label</th>
<th>predict features</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000000 0.000000</td>
<td>1</td>
<td>1.0</td>
<td>4</td>
<td>4</td>
<td>139</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.000000 0.000000</td>
<td>2</td>
<td>2.0</td>
<td>4</td>
<td>4</td>
<td>232</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.000000 0.000000</td>
<td>4</td>
<td>4.0</td>
<td>4</td>
<td>4</td>
<td>118</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.000000 0.000000</td>
<td>8</td>
<td>8.0</td>
<td>4</td>
<td>4</td>
<td>171</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.062500 0.125000</td>
<td>16</td>
<td>16.0</td>
<td>4</td>
<td>4</td>
<td>240</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.312500 0.625000</td>
<td>32</td>
<td>32.0</td>
<td>3</td>
<td>4</td>
<td>135</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.296875 0.281250</td>
<td>64</td>
<td>64.0</td>
<td>4</td>
<td>4</td>
<td>359</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.304868 0.312500</td>
<td>128</td>
<td>128.0</td>
<td>4</td>
<td>4</td>
<td>241</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

finished run
number of examples per pass = 216
passes used = 1
weighted example sum = 216.000000
weighted label sum = 0.000000
average loss = 0.333333
total feature number = 49520
Num weight bits = 18
learning rate = 0.5
initial_t = 0
power_t = 0.5
using no cache
Reading datafile = Test.vw
num sources = 1
average since example example current current current
loss last counter weight label predict features
0.000000 0.000000 1 1.0 4 4 143
0.000000 0.000000 2 2.0 4 4 145
0.000000 0.000000 4 4.0 4 4 166
0.125000 0.250000 6 6.0 4 4 168
0.250000 0.375000 16 16.0 4 4 385
0.375000 0.500000 32 32.0 4 4 141

finished run
number of examples per pass = 60
passes used = 1
weighted example sum = 60.000000
weighted label sum = 0.000000
average loss = 0.315667
total feature number = 12999

Num weight bits = 18
learning rate = 0.5
initial_t = 0
power_t = 0.5
using no cache
Reading datafile = Test.vw
num sources = 1
average since example example current current current
loss last counter weight label predict features
0.000000 0.000000 1 1.0 4 4 97
0.000000 1.000000 2 2.0 3 4 220
0.500000 0.500000 4 4.0 4 4 83
0.500000 0.500000 3 3.0 3 4 111
0.375000 0.250000 16 16.0 4 4 435
0.312500 0.250000 32 32.0 3 4 322
0.343750 0.375000 64 64.0 2 4 516

finished run
number of examples per pass = 65
passes used = 1
weighted example sum = 65.000000
weighted label sum = 0.000000
average loss = 0.338452
total feature number = 13745
Expected: Genre would be strong indicator of words that defined success

Actual: All movies in a genre, good and bad, share similar word choice

Good movies tend to have a more sophisticated text

Bad movies overplay cheesy dialog and word choice

Pattern not obviously apparent, but one clearly exists

Can predict at a rate higher than a 50/50 guess

Weights are relatively miniscule

The presence of a single word won't make a movie

No real negative classifiers

No word can clearly tarnish a movie either

Including word typically considered vulgar
Sources

http://www.boxofficemojo.com/yearly/chart/?yr=2015

Finding Trending Topics

Charles Collins, Yoshi Fujimoto, and Halley Weitzman
The Problem

Twitter uses hashtags to group tweets together and to decide the “Trending Topics” on the platform.

Hashtags are not an accurate depiction of what’s being talked about on Twitter.
The Problem continued

Why are hashtags not accurate at computing trending topics?

- Not all tweets have hashtags
- Twitter is easy to spam/make bots
- Many hashtags refer to the same topic but are different
The Goal

Develop a method for grouping tweets together by their subject matter to find the most popular topics on Twitter.
The Data Collection Stack

Twitter Search API + Twitter Stream API

Node.js script

Forever

Twitter

Moment

Twilio
The Data Processing Stack

Python 2.7
Pycluster/Scipy/NumPy for k-means clustering
Word2Vec (optional) to create word vectors
NLTK Corpora for stopword filtering
The Methodology

Data Collection

Gathered large collection of tweets
Tweaked algorithm to customize tweet filtering
Hosted script on digital ocean
Combined results of multiple queries

Data Processing

Found true labels for tweets based on hashtags
Cleaned up the tweets by removing hyperlinks, stopwords, punctuation
Created word vectors for each tweet (with and without Word2Vec)
Used Pycluster kcluster algorithm (k-means clustering)
The Methodology continued

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
The Methodology continued

```
0 1 0 0 0 0 1 0 1
1 0 0 0 0 0 1 0 0
0 1 1 0 0 0 0 0 0
0 0 0 0 1 1 1 0 0
1 0 0 0 0 0 0 0 1
```

Diagram: Scatter plot with dimensions 0 and 1.
The Evaluation

We calculated accuracy measures by separating

True Positive Rate = \frac{TP}{TP + FP}

True Negative Rate = \frac{TN}{TN + FN}

False Positive Rate = \frac{FN}{TN + FN}

False Negative Rate = \frac{FP}{TP + FP}
### The Results continued

<table>
<thead>
<tr>
<th>DataSet</th>
<th>True Positive Rate</th>
<th>True Negative Rate</th>
<th>False Positive Rate</th>
<th>False Negative Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000 Binary Tweets</td>
<td>41%</td>
<td>72%</td>
<td>59%</td>
<td>28%</td>
</tr>
<tr>
<td>2000 Binary Tweets</td>
<td>1%</td>
<td>98%</td>
<td>99%</td>
<td>2%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DataSet</th>
<th>Cluster 1</th>
<th>Cluster 2</th>
<th>Cluster 3</th>
<th>Cluster 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>4000 4-ary Tweets</td>
<td>2636</td>
<td>327</td>
<td>602</td>
<td>435</td>
</tr>
</tbody>
</table>
The Evaluation

Model ultimately unsuccessful

Lack of data

Ambiguous word meanings

Homonyms

Slang
The Insights

To make this model successful

- Need a way to process ambiguous terms
- Possibly use NLP to determine relevance to hashtag
- More data would need to be collected

Hashtags are placeholder

- Twitter likely aware of their inaccuracy

Use Word2Vec/ML to compare similarities of tweets

- Library creates a dictionary of words
- Can then compares how similar words/phrases are to one another
References

http://www.nltk.org

https://pypi.python.org/pypi/Pycluster

https://www.scipy.org

Movie Lovers ❤

David Steinberg, Fanglun Zhu, Sean Brody
How do advertisers know who are the best people to advertise to?

Current method: Twitter’s built-in user targets

Targets mass amount of user, not singling out the most influential users

Targets them based on what twitter thinks are their likes and dislikes

Users may not be very active on Twitter

Expensive, but not efficient
Influential users

High amount of Followers

High amounts of Tweets

High amount of retweets

Large ratio of times their tweets were retweeted (or liked) to the number of tweets
Example network

Blue: User
Gray: Tweet
Red: Retweet
Dataset: Tweepy

Live stream of Twitter posts using tweepy

Look for tweets with certain keywords

- Movie, Film, Cinema, Theatre
- Plural and #hashtag versions as well
Dataset: Tweeppy

Status:

User ID
Text
Favorite count
Retweet count
(If it is a retweet) ID of the original tweet

User

Username
Display Name
Processing the Database

1. Initial database of retweeted tweets
2. Get the list of tweets that are retweets
3. Get the users that created those tweets
4. Parse those users’ tweets for tweets containing keywords
   a. Same keywords for the original tweepy search
5. Calculate influence score for the user, and save it
Influence Function

User’s influence score = $x/(yz)$

$X = \text{Number of retweets on tweets containing the movie keywords}$

$Y = \text{Total number of tweets containing movie keywords}$

$Z = \text{Number of followers the User has}$
Influence Function Examples

High Follower count, Hight tweet count, High retweet count
   Score: Medium

Low follower count, Hight tweet count, High retweet count
   Score: high

Low follower count, Hight tweet count, low retweet count
   Score: Low
Example network
Example network

1 retweet / 2 tweets = 1/2

12 retweets / 4 tweets = 3
Conclusion

Higher score = higher percentage chance of being more influential

Advertisers can target the highest-scoring users

- Saves money
- Tweets would go to new users organically
- As opposed to mass targeted advertising
Possible Changes in the Future

- Account for the number of likes the tweets has
- Account for people retweeting the retweet of the original tweet
- Account for users writing replies to tweets/retweets
Group 13

Twitter
Relationships
Connor Ford
Daniel Zadorozhnyy
Daniel Barnard
Problem

■ Guess nature of relationships between two users

■ Determine factors most indicative of the type of relationship

■ Determine confidence in our guesses
Initial Proposal

- Determine what type of relationship two people have by the text they use when tweeting each other
  - Romantic, friendship, close friendship, professional, family, fan/celebrity, etc.
- Use this data to try and determine how users interact within these relationships
Tools

**Tweepy**

An easy-to-use Python library for accessing the Twitter API.

**TextBlob**
Tweepy

- Python library for accessing the Twitter API
- Cache tweets from universal stream as JSON objects
- Get users involved in Twitter interaction
- Get metadata about users, such as number of followers/followed a user has
TextBlob

- Python library for text analysis
- Provides natural language processing tasks
  - Parts-of-speech tagging
  - Noun phrase extraction
  - Sentiment analysis
  - Classification
  - Translation
  - etc.
User Categories

- General User
- Person of Interest
  - Celebrity
  - Artist
  - Politician
- Professional
  - Organization
  - Company
  - Product
Relationship Categories

- Promotion
  - General
  - Co-branding

- Fans
  - Person
  - Product
  - Group
  - Organization
  - Popular Account

- Friends
  - General
  - Close
  - Acquaintances
  - Family
  - Work
  - Romantic
Promotion - Co-branding vs General
Attributes of Interest

- Followers to following ratio
- Frequency of contact between two users
- Contact initiation ratio between two users
Following vs Followers
Methods

1. Pull tweets from a live stream
   - Limit to tweets with mentioned users (including @ tag)
   - Collect training data and save its json to a file (~100 tweets)

1. Manually label training
   - Subjective, but good way to deal with natural language
Methods

3. Generate Model from training data

- Create word vector of word frequency, remove stop words, create cumulative word frequencies
- Collect follower information of user writing tweet and user mentioned
- Sentiment analysis of text (TextBlob().sentiment)
- Assemble into features into model
## Model Example

<table>
<thead>
<tr>
<th>Sub Label: Fan, Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top Terms:</td>
</tr>
<tr>
<td>sign - 3</td>
</tr>
<tr>
<td>petition - 2</td>
</tr>
<tr>
<td>nfl - 2</td>
</tr>
<tr>
<td>election - 2</td>
</tr>
<tr>
<td>fans - 2</td>
</tr>
<tr>
<td>Average Sentiment: 0.148</td>
</tr>
<tr>
<td>Follower Ratio: 0.00214</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sub Label: Fan, Person</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top Terms:</td>
</tr>
<tr>
<td>love - 4</td>
</tr>
<tr>
<td>music - 3</td>
</tr>
<tr>
<td>will - 3</td>
</tr>
<tr>
<td>make - 3</td>
</tr>
<tr>
<td>thank - 2</td>
</tr>
<tr>
<td>Average Sentiment: 0.140</td>
</tr>
<tr>
<td>Follower Ratio: 0.00472</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Label: Friends</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top Terms:</td>
</tr>
<tr>
<td>day - 3</td>
</tr>
<tr>
<td>get - 2</td>
</tr>
<tr>
<td>now - 2</td>
</tr>
<tr>
<td>dinner - 2</td>
</tr>
<tr>
<td>love - 2</td>
</tr>
<tr>
<td>Average Sentiment: 0.170</td>
</tr>
<tr>
<td>Follower Ratio: 20.1681</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sub Label: User, YouTube</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top Terms:</td>
</tr>
<tr>
<td>youtube - 7</td>
</tr>
<tr>
<td>video - 5</td>
</tr>
<tr>
<td>liked - 4</td>
</tr>
<tr>
<td>total - 2</td>
</tr>
<tr>
<td>playlist - 2</td>
</tr>
<tr>
<td>Average Sentiment: 0.196</td>
</tr>
<tr>
<td>Follower Ratio: 0.00312</td>
</tr>
</tbody>
</table>
Methods
4. Label tweets automatically via comparison to the generated model

- Extract tweet attributes (active/passive users, followers, sentiment)
- Calculate scores on attributes and find most appropriate match among primary label relationship models
- Now given a primary label, repeat calculations and comparisons to match among secondary labels
5. Assess accuracy through manual auditing

- Measure ratio of matching labels with an algorithm
  - \( \text{Recall} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}} \)
  - \( \text{Precision} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}} \)
Experiment

Model generated from 217 tweet/label pairs

Run on sets of 60 unclassified tweets
Results

Looking at all criteria:

- Accuracy/Precision was often 0, except for the User, Youtube category,
  - Accuracy: 1.0  Precision: 0.235

Looking at just word vectors:

- Again, Accuracy/Precision was 0 for many, but fewer than looking at all criteria
  - Fan, Organization - Accuracy: 0.667  Precision: 0.667
  - Fan, Person - Accuracy: 1.0  Precision: 0.143
  - User, Youtube - Accuracy: 1.0  Precision: 0.75
  - Professional, Co-branding - Accuracy: 0.333 Precision: 0.167
Insights

- Since many categories appear much less frequently, they are hard to train for and easy to miss.
- Generated Model relies and varies heavily on training data.
- Some categories have less varying, more exclusive data (ie Youtube).
- Large amount of attributes requires very large training dataset for each category.
  - Otherwise they hurt accuracy.
Challenges

- Computational intensivity of some features (seeing if two users are mutual followers)
- Subjectivity of language
- TextBlob python package is still beta (v0.12)
- SPAM, knowing what to ignore, auto-generated tweets
- Large amount of training data required, manually time consuming, human error
- Fluidity of categories
- Evaluation of accuracy tested against manually checked results
Challenges

- Twitter API use limit

15 Minute Windows

Rate limits are divided into 15 minute intervals. Additionally, all endpoints require authentication, so there is no concept of unauthenticated calls and rate limits.

There are two initial buckets available for GET requests: 15 calls every 15 minutes, and 180 calls every 15 minutes.