Sentiment Analysis of Comments on Hacker News: First Steps

Max Joslyn

May 11, 2015

Introduction

The goal of **sentiment analysis**, also called sentiment or opinion mining, is to extract the mood or attitude of a text's author. Techniques and knowledge from statistics, linguistics, and computer science must be combined to identify such information accurately. One example in the academic literature is Hu and Liu 2004. This paper illustrates one prominent pplications of sentiment mining, which is the investigation of reviews (for any kind of product: movie, gadget, food, etc.) The information gained from the analysis helps determine which features of a product are most liked or disliked by users. This in turn guides business decisions, such as deciding which features to prioritize for revision in the next version of the product.

Data

Hacker News (HN) is a news aggregator focused on startups, science, and technology. The data set for this project consists of comments made by HN users on articles posted by other users. Hacker News comments are well-suited to sentiment analysis for these reasons:

- HN is an English-language site.
- Site policy and culture keeps discussion relatively civil.
- By Internet standards, comments are long and well-edited.
- The website can be accessed programmatically through a crude but functional official API, as well as through community packages which predate the official one.

Research Question

What stories do HN commenters react to positively? negatively?

Methods

Using a Python script, which in turn employs the package `hackernews-scraper`, 1000 recent comments were downloaded from Hacker News. My script is included in this project. The comments are in JSON format, which is a standard for representing objects as text. If you are unfamiliar with JSON, here is an example. Suppose we we want to store a person's name and surname, their eye color, and their user ID as a JSON file. It would look like this:

{ "name":"Bob" "surname":"Johnson" "eyes":"green" "ID": 12345 }

The JSON comment files are imported to R with the `jsonlite` package.

The comments are cleaned of HTML tags, and the HTML code for the apostrophe (') is removed. Other HTML codes are not touched.

Polarity is a basic measure of an item's sentiment. It has two components:

1. Whether the item is positive or negative.
2. How large the value of the positivity or negativity is.

The polarity of comments is calculated, and then the polarity of reaction to each story is calculated from the polarities of its comments.

Code has been run behind the scenes: here is the list of the top stories by level of positive sentiment.

(Note: this list actually contains recent stories with the 3rd- through 12th-highest positive polarity of comments, and not the 1st- through 12th-highest. I've removed the actual top two stories, which have extremely high positive polarity, because I suspect they are a fluke of the primary sentiment-detection algorithms. I don't have proof, though, so it merits further investigation.)

```
##                                                                 story_title
## 1                                                  Don't believe Orson Welles
## 2  A computing scientist's approach to a once-deep theorem of Sylvester's (1988)
## 3        "Map Maker will be temporarily unavailable for editing starting May 12"
## 4          Show HN: rust-rss - library for serializing the RSS web content format
## 5                            Why the myth of the "10X" programmer is so destructive
## 6                                               Implementing a Virtual Machine in C
## 7                                                 Arq for Windows finally available
## 8                                               A Lisp syntax with less parentheses
## 9        Unnecessary medical care is harming patients physically and financially
## 10                                                            Dart's Macro Language
```

Next, here are the recent stories with the highest **negative** polarity of comments. Note that the stories with the **highest** negativity, i.e. the most negative response, are at the **top** of this list.

```
##                                                                  story_title
## 1                                        Why we should't judge a country by its GDP
## 2                                   Fast Approximate Logarithms, Part I: The Basics
## 3  How New York and San Francisco's Tight Housing Markets Are Hurting the Economy
## 4                                                  A re-introduction to JavaScript
## 5                                                       Code Review Best Practices
## 6                                                        Stupid certificate tricks
## 7                                          How do you dismantle a nuclear submarine?
## 8                                  Python 3 in Science: the great migration has begun
## 9                         Is the Universe a Giant Computer Simulation? Here's the Evidence
## 10                                       Accused of Spying for China, Until She Wasn’t
```

Further Investigation

My work here is rudimentary at best. Here are some questions or steps to be taken that would improve this analysis.

- More precise selection of comments, such as by setting a range of dates to scrape within.
- Restriction of comments to those scraped from articles on a particular topic or field.
- Examining the relation between the points a story receives (which determine its visibility on the site) and the sentiment of its comments.
- How does comment sentiment change from when an article is first linked on HN, to when it has gained some popularity on the site?

- Removal of more unnecessary HTML alt-character codes than the apostrophe.
- Do positive-polarity comments necessarily indicate a well-received article? Consider this scenario: the first few commenters point out the obvious flaws in the story. Later commenters engage in optimistic discourse about how to improve the story, rather than rehashing its flaws again. In this case, there could be a lot of positive sentiment in the comments, even if the article really wasn't very well-received by the users.
- As mentioned above: there seem to be flaws or weak spots in the sentiment tools provided by the qdap package. Some of this might be fixed by better data selection and cleaning, and some it might be fixed by tweaking the word lists used internally by the package.
- It would also be worth investigating the academic literature on sentiment analysis for cutting-edge algorithms and directions in analysis. I was only able to read one or two papers during the course of this project, and there's a lot out there.

Notes

This is my final project for Math 241: Case Studies in Statistical Analysis, a class at Reed College with instructor **Albert Kim**.