The Sound of Social Media

Analyzing User-Generated Content on Firm-Hosted Social Media Pages

Mochen Yang

FARCON 2017

August 23, 2017

About me

- My name is **Mochen Yang**, a Ph.D. candidate at Department of Information and Decision Sciences
- I do research on topics such as social media, user-generated content, and machine learning
- I teach an undergraduate course on data analytics
- I'm generally interested in how we can create business value from data using statistical analysis and machine learning techniques

About you

A diverse and sophisticated audience!

- Industries: financial services, manufacture, heathcare, retail, technology, research institutions, consulting, . . .
- Background: students, data analysts, analytics architects, researchers, managers/directors, engineers, consultants, . . .

About this talk

What this talk IS about:

- A guided-tour of how to *collect*, *analyze*, and *understand* user-generated content on social media platforms
- A series of business and technical issues to consider along the way
- Some ideas about extracting the value of user-generated content, or online textual content in general

What this talk IS NOT about:

- A collection of technical details including web scraping, text mining, neural networks,...
- Programming tutorial

Background: Facebook Business Pages

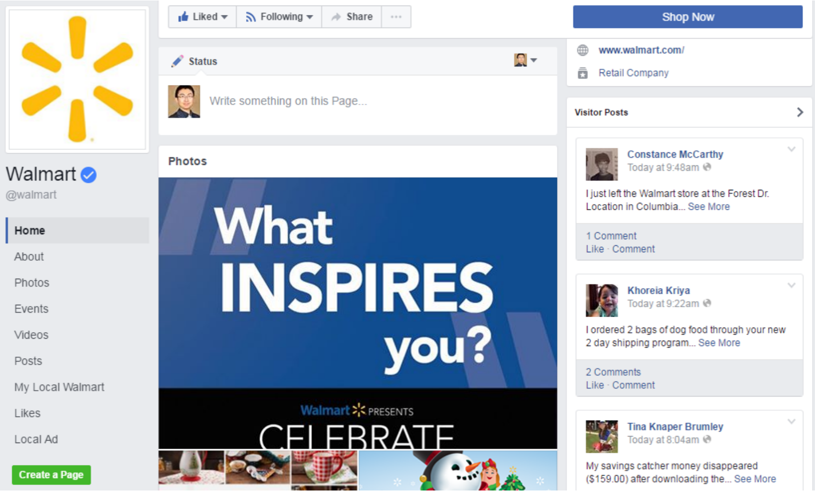


Figure 1: Walmart's Facebook business page

Background: Facebook Business Pages

- 60 million pages
- 1 billion visits/month
- 350 posts/day
- 0.8 likes/post
- 0.7 comments/post

Figure 2: Walmart's Facebook business page

What Do We Want to Know?

Q1 [What]: What do people talk about on a company's Facebook business page? $\leftarrow$ Main focus of this talk

Q2 [So What]: What are some business implications of this user-generated content?

Q3 [How]: How does the comapany harvest its value and/or deal with its challenge?

Analyzing User-Generated Posts: Ingredients

1. Collect: collect user-generated posts efficiently and automatically
2. Summarize: obtain some aggregated content categories of the posts
3. Label: categorize each post systematically and at scale
   - We want to leverage machine learning methods for this task
4. Analyze: analyze the data to obtain some insights

Roadmap

1. **Collect**
2. Summarize
3. Label
4. Analyze

Think about Ethics First

Before you start collecting data, think through these questions:

- Is it legal/ethical to collect this data?
- Does it abide necessary rules and regulations?
  - E.g., in academia, we have Institutional Review Board (IRB)
- Can you take measures to protect privacy?
  - E.g., anonymize data
- ...

Automated Data Collection via Facebook API

API stands for **Application Programming Interface**, a specialized type of "language" for building applications. It allows developers to communicate with the service provider

- Consider a "Log-in with Facebook" button, what communications do we need?
- Many companies open up their services to developers via API
- Facebook has a well-developed set of APIs, known as Graph API
- We can use API to collect public information on Facebook in a programmatic way

To-Do List (with demos)

You need to do the following:

- A Facebook Developer account, often tied with your own Facebook account
- Create an "App", get the corresponding *app id* and *app secret* for authentication purpose - these are your ID
- Obtain *access token*, a time-sensitive permission to request for data
- Send actual data requests, via tools you like. I use python and libraries `facebook`
- Process results
- Automate
    - Deal with paging
    - Deal with request rate
    - Deal with data persistence

Roadmap

1. Collect
2. **Summarize**
3. Label
4. Analyze

Obtain Aggregated Content Categories

We want to know the salient "topics" that users talk about

- This is an open problem, many solutions exists
- Domain is familiar:
  - Rely on previous understanding/framework
  - Rely on expertise
- Domain is new:
  - Data-driven approach
  - Human-driven approach $\leftarrow$ I used this

Data-Driven Approach: Clustering

- Organizing data points/objects (e.g., Facebook posts) into homogeneous (and, hopefully, meaningful) groups
- Each group is called a cluster
- Ideally, we want clustering results to have two properties:
  1. High intra-similarity: data points in the same cluster should be similar to each other
  2. Low inter-similarity: data points in different clusters should be different from each other
- Clustering analysis is a type of **exploratory** data analytics

Data-Driven Approach: LDA for Topic Modeling

Latent Dirichlet Allocation (LDA) in English:

- User specifies the number of topics to look for
- Each post is modeled as a mixture of all topics with certain proportions
- Each topic is modeled as a mixture of all unique words with certain porportions
- From actual posts, learn/estimate these mixture proportions

Interpret LDA output:

- For each post, look at its most salient topics
- For each topic, look at its most salient words
- Subjectively interpret what each "topic" stands for, and what each post talks about

Limitations of Data-Driven Approach

- It is data-driven, unaware of the context/domain
- It is exploratory, human interpretations are needed anyway
- It requires hard-to-get input in order to run
- It does not work well with short texts (unless carefully tuned), which are typical on social media

Human-Driven Approach: Grounded Theory Approach

Grounded Theory Approach is an iterative process of theory discovery (in this case, content category discovery):

1. **Open Coding**:
    - Hire several human assistants to manually read a small, randomly selected, subset of posts and write down topics they find salient
    - Consolidate topics, resolve disagreements
    - Potentially iterate until topics are “saturated”
2. **Structured Coding**: Use the established content categories to systematically label other posts

Our Content Categories

- Positive testimonial and appreciations
- Complaints about product and service quality
- Complaints about money-related issues
- Complaints about Corporate Social Responsibility issues
- Questions and Suggestions
- Irrelevant Messages

Roadmap

1. Collect
2. Summarize
3. **Label**
4. Analyze

How to Do Structured Coding/Labeling?

We want to build a machine learning classification model, for several reasons:

- Scalability
- Cost-efficiency
- Continuous usage

A **classification model** predicts certain well-defined *categorical* outcome based on some predictors (a.k.a. features/attributes), based on certain classification algorithm.

- It is a type of **predictive** data mining technique

How to Build Classification Model?

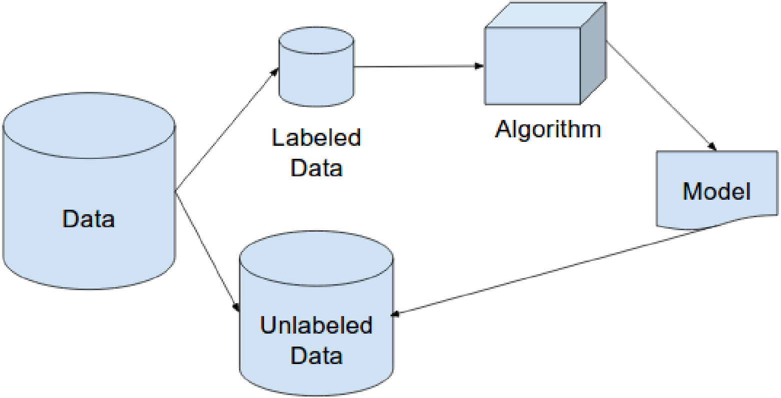


Figure 3: Build a Classification Model

Get Labeled Data: Amazon Mechanical Turk (AMT)

Figure 4: Mechanical Turk, 18th century, artificial artificial intelligence

Brief History of AMT

- Originally developed in-house by Amazon to detect duplicate product postings
- Now probably the largest online "human intelligence" labor market
- Scalable workforce on-demand
- Used by researchers and practitioners to complete a series of tasks:
  - **"Labor" tasks**: label data, tag images, digitize books,...
  - **Subject pool**: Participate in experiments
  - **Crowdsource**: provide feedback for product ideas, etc.

How does it Work?

1. **Requesters** create tasks to be completed, called **Human Intelligence Tasks** (HITs), with payment levels specified in advance.
2. **Turkers** (workers) browse tasks and accept the ones they want to work on
3. Tukers complete the tasks, requesters examine their quality
4. Requesters either accept or reject the results
5. If accepted, turkers get paid the promised amount, and Amazon gets paid an additional fee

What you need to become a requester:

- Register for a “requester account”
- Set up “Amazon Payment account” - an unpleasant process that may require a green card/citizenship
- (Optional) set up an AWS account for API access

Know Your Employees: Turker Demographcis

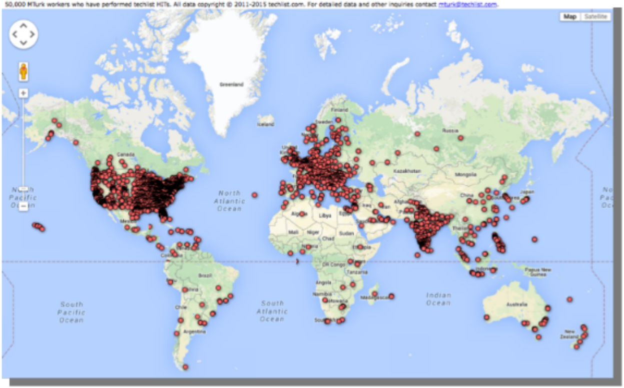


Figure 5: Where are they?

Know Your Employees: Turker Demographchis

Source: `http://www.behind-the-enemy-lines.com/2015/04/demographics-of-mechanical-turk-now.html`

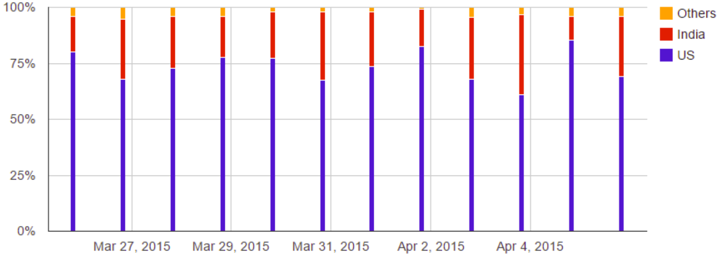


Figure 6: Where are they (bar chart)?

Know Your Employees: Turker Demographcis

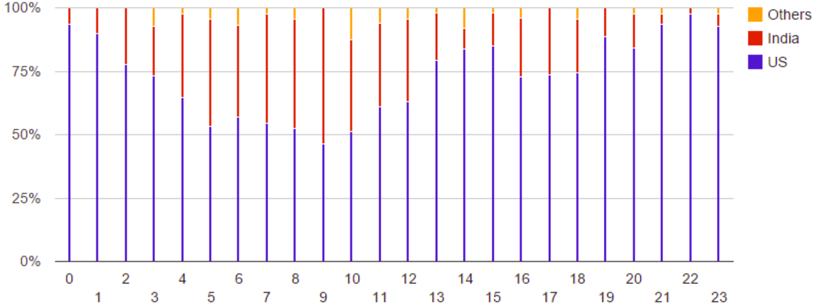


Figure 7: Be aware of timezone

Know Your Employees: Turker Demographcis

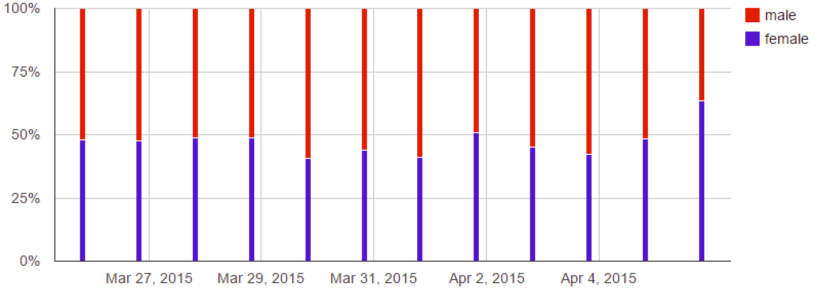


Figure 8: Balanced gender

Know Your Employees: Turker Demographcis

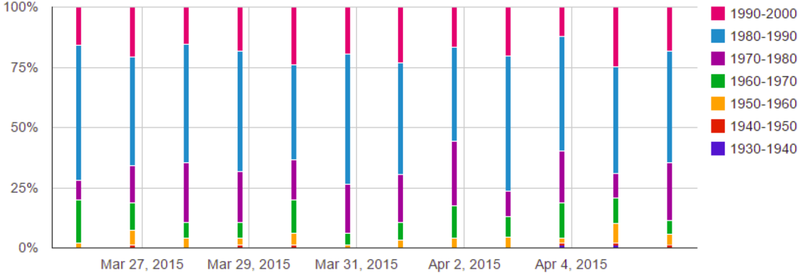


Figure 9: Half are 30-year-olds

Know Your Employees: Turker Demographcis

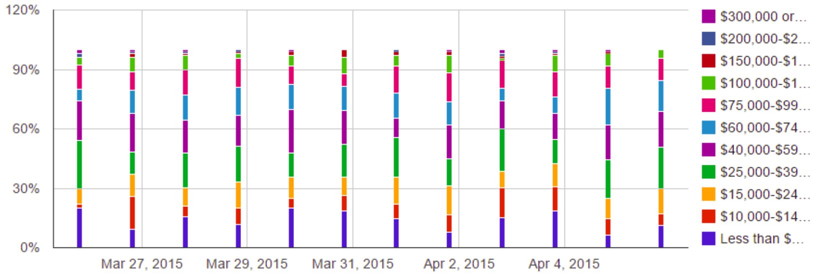


Figure 10: Median income around 50K/year for US turkers

How Much to Pay?

Median wage on AMT is about \$1.38/hour

- Short tasks (a few minutes) often award around 10 cents
- Requesters can revoke payment (if justified) or add bonus (if wanted)
- Unreasonably low payment hurts participation, unusually high payment does not really help quality much
- Turkers can rate requesters, so be aware of reputation

Can You Screen? Yes You Can!

Qualification is a specialized "marker" that can be used to select desired participants:

- *System qualifications*: qualification types created by Amazon
  - Location
  - Previous acceptance/rejection rate
  - "Master"
- *Premium qualifications*: all kinds of characteristics (age, political affiliation, online behaviors, marital status, family status,...), come at extra cost
- *User defined qualifications*: you can make your own qualification type
  - E.g., create a *qualification test*, those who score high enough get to do your tasks and earn money

Automate, Again

Like Facebook, AMT has its own API. You can use it to:

- Manage your HITs (create, change, track, delete,...)
- Manage qualifications (create, score, grant/reject,...)
- Contact workers

There is an R package `MTurkR` with easy-to-use functions to make API calls

Miscellaneous Issues with AMT

- AMT is not good for all tasks
  - Tasks that are not easy to explain/understand - it's hard to train turkers to do complicated tasks
  - Tasks that take too long to complete
  - Tasks that are too subjective (unless the goal is to get diverse opinions)
- Give fair payment
- Don't forget quality check
  - E.g., have multiple turkers label the same post and take majority vote

Building Predictive Classification Model: General Process

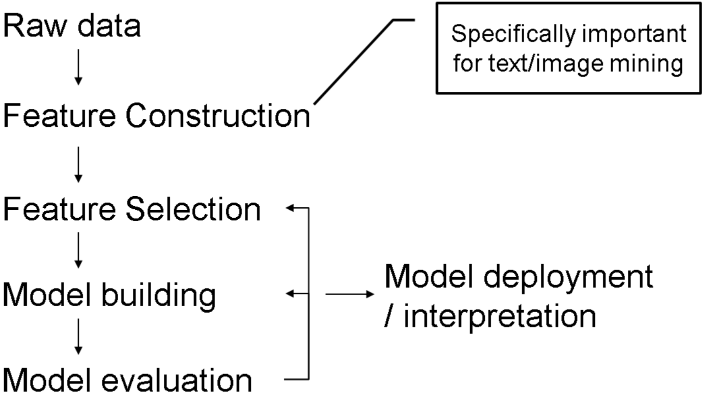


Figure 11: Build Predictive Model

Text to Numbers: Bag-of-Words Approach

- Simple, commonly used way of representing textual data
- Each post is broken down to a set of individual words
- Each unique word is a feature/variable
- Several ways to construct numeric value of each feature
    - Binary
    - Frequency
    - TF-IDF

Bag-of-Words: Simple Example

D1	Welcome to data analytics!
D2	Data analytics study data.
D3	Data Mining finds patterns from data.
D4	Text Mining finds patterns from text.

Figure 12: A simple corpus

The unique words {welcome, to, data, analytics, study, mining, finds, patterns, from, text}

Bag-of-Words: Simple Example

	welcome	to	data	analytics	study	mining	finds	patterns	from	text
D1	1	1	1	1	0	0	0	0	0	0
D2	0	0	1	1	1	0	0	0	0	0
D3	0	0	1	0	0	1	1	1	1	0
D4	0	0	0	0	0	1	1	1	1	1

Figure 13: Binary representation

Bag-of-Words: Simple Example

	welcome	to	data	analytics	study	mining	finds	patterns	from	text
D1	1	1	1	1	0	0	0	0	0	0
D2	0	0	2	1	1	0	0	0	0	0
D3	0	0	2	0	0	1	1	1	1	0
D4	0	0	0	0	0	1	1	1	1	2

Figure 14: Frequency representation

Bag-of-Words: Simple Example

	welcome	to	data	analytics	study	mining	finds	patterns	from	text
D1	1.39	1.39	0.29	0.69	0	0	0	0	0	0
D2	0	0	0.58	0.69	1.39	0	0	0	0	0
D3	0	0	0.58	0	0	0.69	0.69	0.69	0.69	0
D4	0	0	0	0	0	0.69	0.69	0.69	0.69	2.77

Figure 15: TF-IDF representation

Bag-of-Words: Limitations

1. Little information about relations among words
   - "I love data analytics" and "Data analytics loves me" have exactly the same representation
   - Considering phrases as features can mitigate this problem, at the cost of having a lot more features
2. Almost no information about the context in which words appear
   - "Take a picture" and "A Hollywood picture", the word "picture" has different but related meanings
   - Even considering part-of-speech cannot solve this polysemy issue
3. Result in sparse representation, causing computational burden
   - Lots of words only appear in very few posts

Text to Numbers: Word Embedding Approach

Word Embedding is a drastically different way of representing textual data that becomes popular recently due to the *success of deep learning* and *availability of big data*

- It captures rich semantic information based on an important assumption in linguistics:
  - Words that appear in the same context have similar meanings
  - E.g., "cat jumps over the table" and "dog jumps over the table", "cat" is therefore similar to "dog" because they appear in the same context
- Many implementations and flavors, let's look at `Word2Vec`
  - Created at and popularized by `Google`

Word2Vec: Intuitive Introduction

- Each word is represented by a **vector** of numbers (hence the name)
- Modeler specifies dimension of each vector, and a window of context
  - Window: how many words before and after are consider to be the "context"
- The vectors are "learned" from huge amounts of textual data
- Two algorithms to learn the vectors:
  - Use surrounding words to predict a focal word
  - Use a focal word to predict surrounding words

Word2Vec: Demo with Facebook Posts

- About 0.5 million user posts on Facebook business pages
- About 300,000 unique words

Use Word2Vec: Recurrent Neural Network

Motivation: why do we need Recurrent Neural Network (RNN) for content classification with word embeddings?

- Why neural network: to take advantage of word-level rich representation
- Why "recurrent": to take advantage of the sequential nature of text
  - Recurrent means a sequence of things that are connected with one another
- A RNN is suitable to deal with sequential data, such as text or speech

Use Word2Vec: Recurrent Neural Network

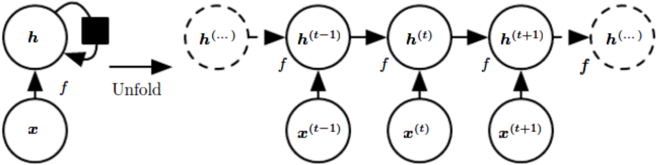


Figure 16: A simple RNN

Intuition: mimic a **reading** process:

- Steps (t): there are multiple steps, naturally correspond to the sequence of words
- Input (x): a sequence of words, one word each step
- Configurations: inner states (h) and transition functions (f), specifies how internal status of the network changes over time
- Output: content category prediction

Roadmap

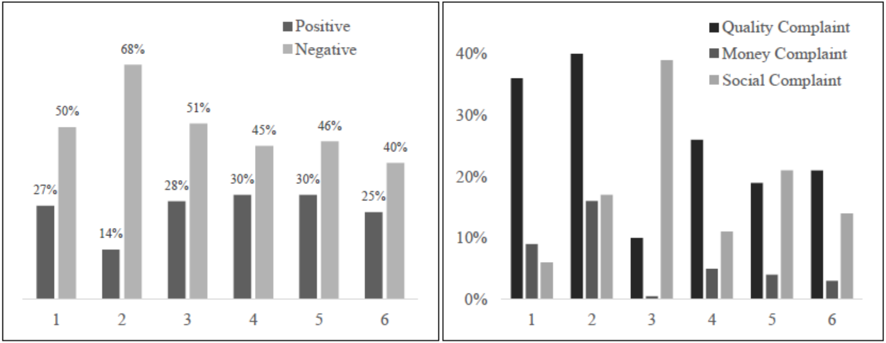
1. Collect
2. Summarize
3. Label
4. **Analyze**

Final Step: Analyze

Our data:

- All user-generated posts (0.5 million) created in 2012, on 40 Fortune-500 firms' Facebook business pages across 6 consumer-facing industries
  - Airline, Commercial Banking, General Merchandiser, Specialty Retailers, Food and Drug Store, Consumer Products
- We developed 7 content categories
  - Positive testimonials, complaints about quality/money/ethics, questions, suggestions, irrelevant messages
- We hired AMT turkers to manually labeled 12,000 posts
  - Content categories and sentiment (positive/negative)

Descriptive Analyses



Note. Industry 1 – Airline; 2 – Commercial Banks; 3 – Consumer Products; 4 – Food and Drug Stores; 5 – General Merchandisers; 6 – Specialty Retailers.

Figure 17: Sentiment and Content distributions across industry

Descriptive Analyses

A few interesting things:

- Across industries, there are more negative user posts than positive ones
- Among different types of complaints, the distribution differs across industries
  - In airlines and commercial banks, most complaints are about quality of products/services
  - In consumer products and general merchandisers, more complaints are about ethics and PR issues than about quality

Statistical Analyses

We used regression analyses to understand which type of posts is associated with more/fewer engagement from other users, measured in likes and comments

- On average, negative posts attracted more likes and comments than positive posts
- Ethics-related complaints tend to receive more likes than quality-related complaints
- Quality-related complaints tend to receive more comments than ethics-related complaints

What are some implications to companies?

Add Data Mining to the Picture: A Common Pitfall

Use classification model to label a much larger sample of posts and run analyses

- Pros: large sample size helps detecting subtle patterns
- Pitfall: data mining predictions are never error-free
  - These errors are called **measurement error** in statistics
  - They make your data "noisy", and lead to biased estimations
  - Harmful even if *error is completely random*, i.e., no "averaging out"

Add Data Mining to the Picture: Remedy

Trouble-maker comes to rescue!

- Data mining methods come with performance measures (accuracy, predicion, recall, . . .), these are good **quantifications of error**
- With error quantification, there are statistical methods to *correct for biases*
- Check out our recent paper on this topic: Mind the Gap: Accounting for Measurement Error and Misclassification in Variables Generated via Data Mining, Mochen Yang, Gediminas Adomavicius, Gordon Burtch, Yuqing Ren link to paper

Thank You! Questions?