Mining and Analyzing Social Media: Part 1

Dave King

January 7, 2013

Abstract

Overview of the data mining and analysis of social media, exploring the application of various data mining, textual mining and analytical techniques to **social media data sources**. The focus will be on the **practical application** of these techniques for the purposes of:

- Monitoring of social media sources
- Analyzing content to identify leading issues and sentiment
- Analyzing and forecasting trends
- Identifying and profiling influential participants, subgroups and communities

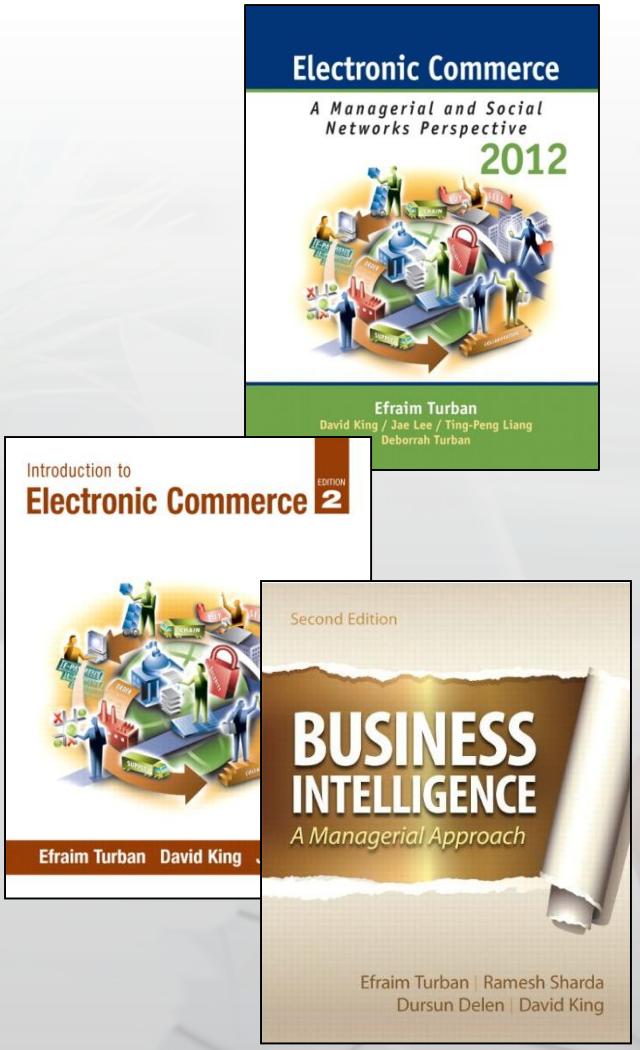
Agenda: Part 1

- My Biography
- Resources
- Social Media Defined
- Data Mining Example
- Text Mining Processes
- Using Text Mining for Prediction
- Brief Look at Programming for Prediction

Agenda: Part 2

- Sentiment Analysis & Opinion Mining Defined
  - Business Interest & Software Packages
  - Levels of Analysis
  - Automated Classification
- Social Network Analysis
  - Defined
  - History
  - Basic techniques and measures
  - Ego and Social-Centric Analysis

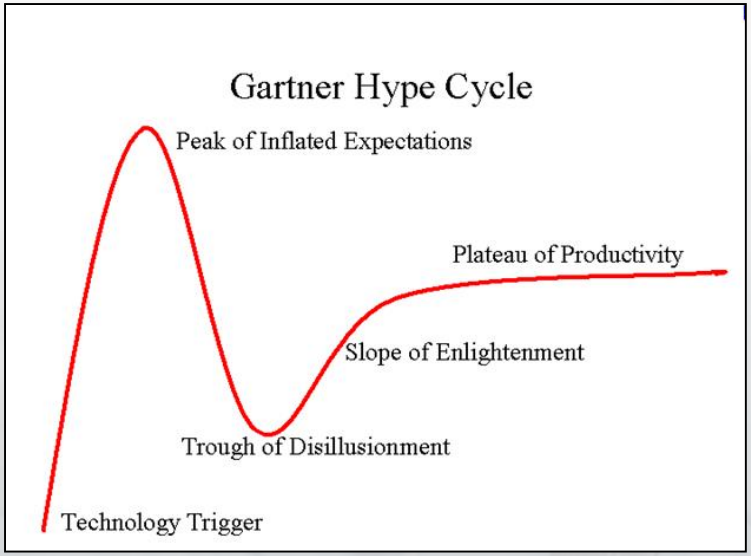
Biography: Dave King



- EVP of Product Development and Management at JDA Software
- 30 years in enterprise package software business
- 15 years as university professor
- 15 years as Co-Chair of the Internet & Digital Economy Track (HICSS)
- Long time interest in various aspects of E-Commerce, Business Intelligence, Analytics (including Text Analytics)

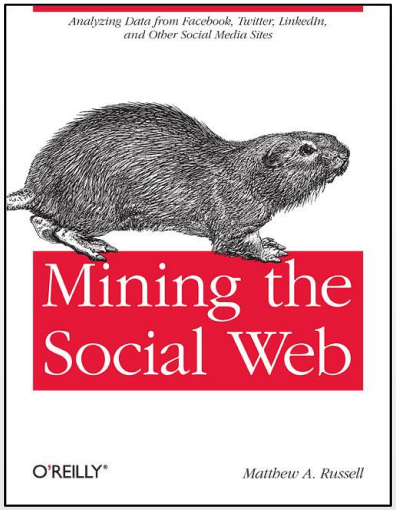
Personal Experiences with Analytics

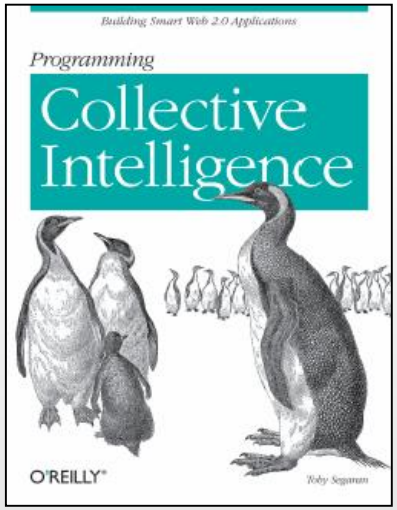
- Taught applied statistics and math modeling
- In software R&D
  - Optimization in the 80s
  - Natural Language Frontends
    - NLI Query & CMU Robotics Lab
  - EIS Competitive Analysis
    - Dow Jones and Reuters
    - Verity Topics
    - NewsAlert
  - InXight's Hyperbolic Tree
  - Supply Chain Analytics
  - Sentiment Analysis for Retailers

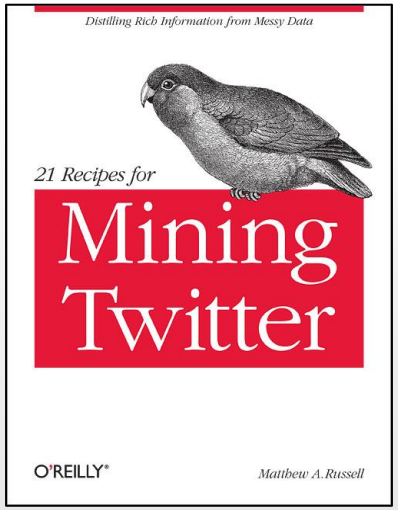


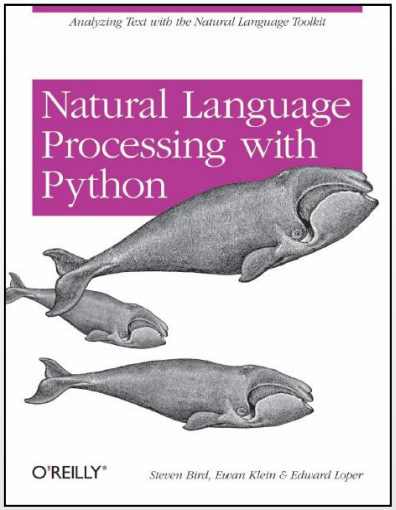
- In the case of many of these advanced techniques, often the audiences have been small, sometimes bewildered, and often fleeting.

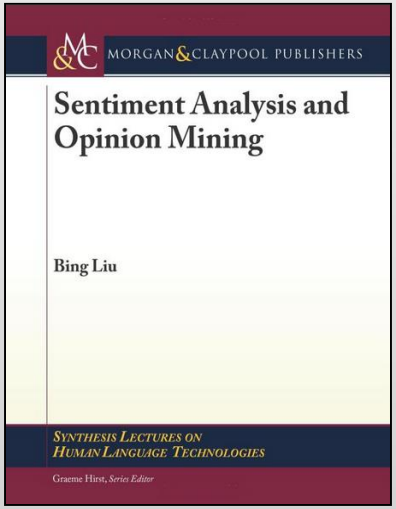
Text Mining Resources

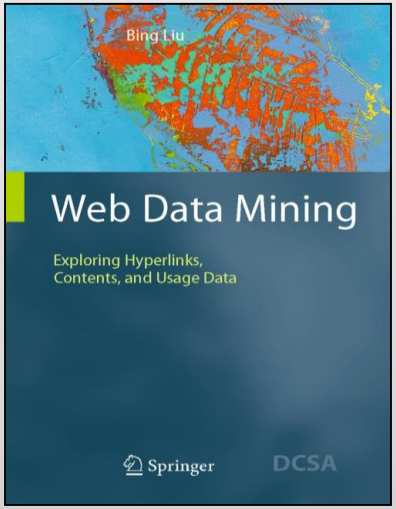


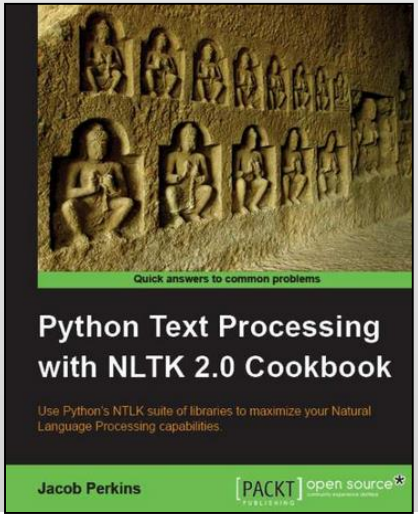


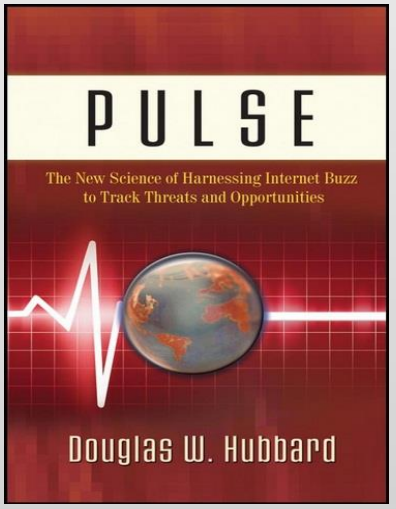




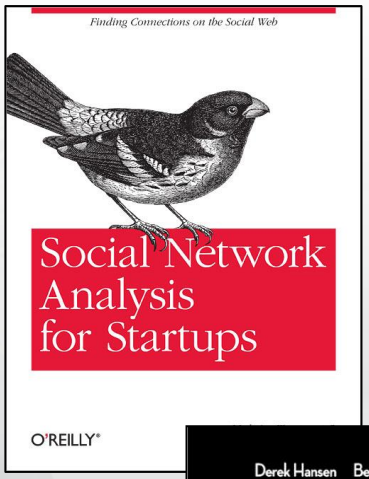


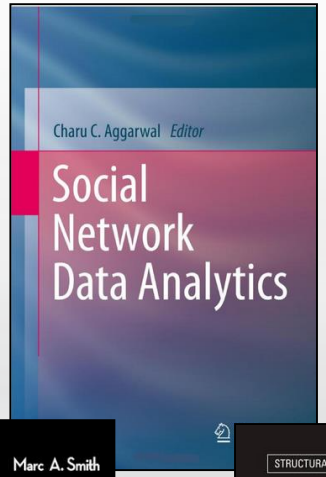


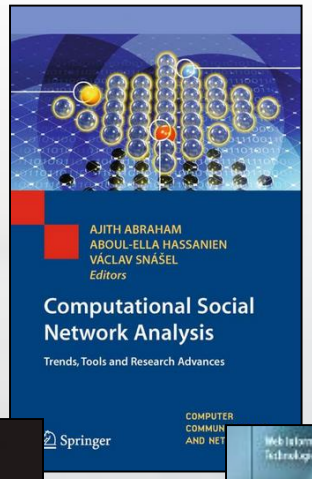


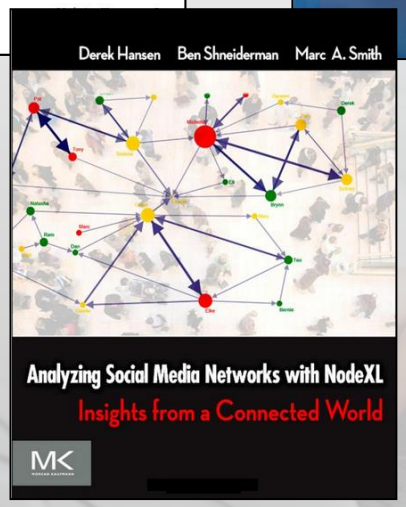


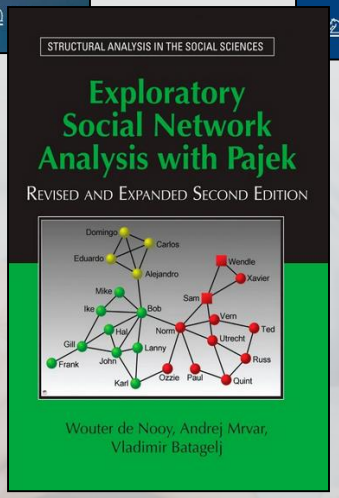
Social Networking Analysis Resources

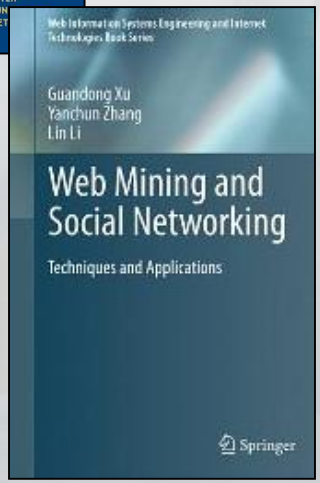












What is Social Media?

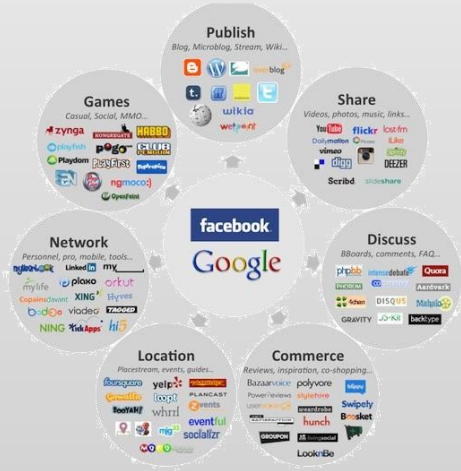
Defined

are online technologies and practices for social interaction enabling sharing opinions, insights, experiences, perspectives and media itself.

Defined

is the media we use to be social. That's it.

Social Media Types: Take Your Pick



Social Media is Still Huge!
Alexa Traffic Oct 6, 2012

Rank	Website	Type
1	Facebook	Social
2	Google	Search
3	YouTube	Social
4	Yahoo!	Search
5	Baidu.com	Search
6	Wikipedia	Social
7	Windows Live	Search
8	Twitter	Social
9	QQ.COM	Portal
10	Amazon.com	E-Commerce
11	Blogspot.com	Social
12	LinkedIn	Social
13	Taobao.com	E-Commerce
14	Google India	Search
15	Yahoo! Japan	Search

Social Media is Still Huge!
Growth in Registered Users 2011 to 2012

Facebook: 750M -1B

Twitter: 200M - 500M

LinkedIn: 100M – 175M

Social Media is Still Huge!
If Social Media sites were countries…

China: 1.4B

India: 1.2B

Facebook: 1.0B

Twitter: 500M

US: 310M

Social Media is Still Huge!
Usage Per Day

Analyzing Social Media: Two Paths

Media - Content

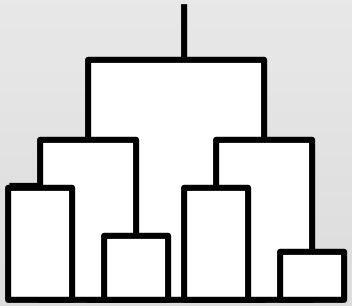
Social - Network

Analyzing Social Media: Two Paths

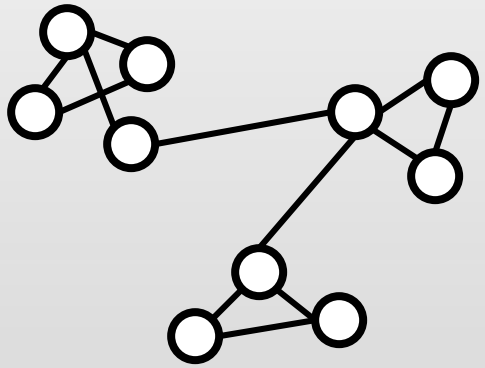
An Example: Which Blogs are Similar?

	Term1	Term2	Term3	…	TermM
Blog1	1	0	0	…	1
Blog2	0	0	1	…	0
Blog3	0	1	0	…	1
…	…	…	…	…	…
BlogN	0	0	0	…	1

	Blog1	Blog2	Blog3	…	BlogN
Blog1	-	1	0	…	1
Blog2	0	-	1	…	0
Blog3	1	1	-	…	0
…	…	…	…	-	…
BlogN	1	0	1	…	-



Cluster Analysis (e.g. K-Means)



Social Network (Graph) Analysis

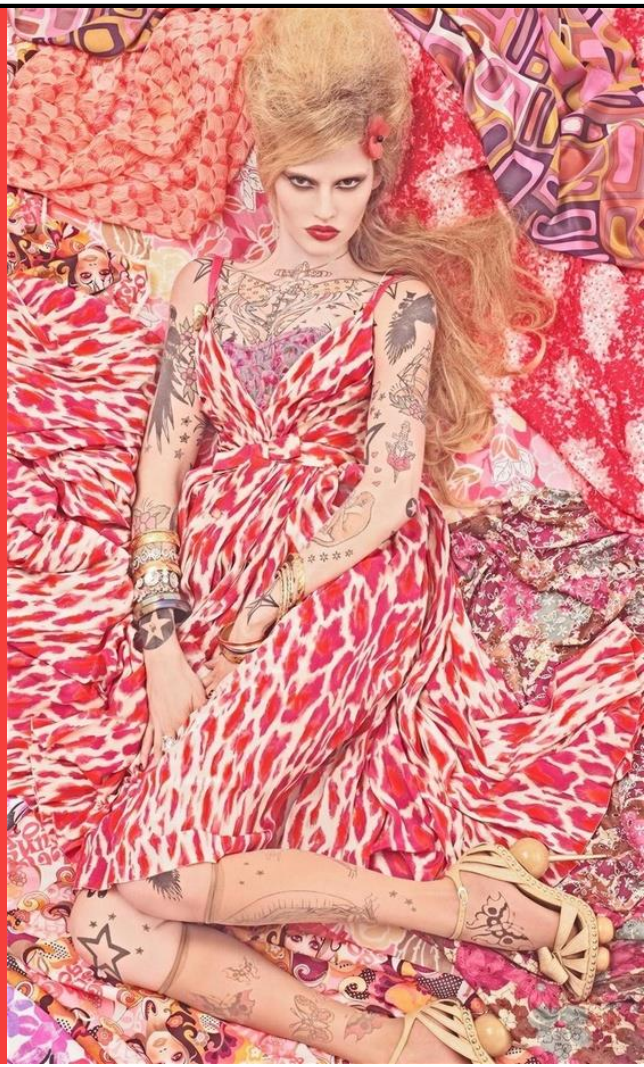
social Media Formats

- Articles
- Comments
- Messages
- Reviews
- Ratings
- Rankings
- Pictures
- Videos
- Music
- Locations
- Tags
- …

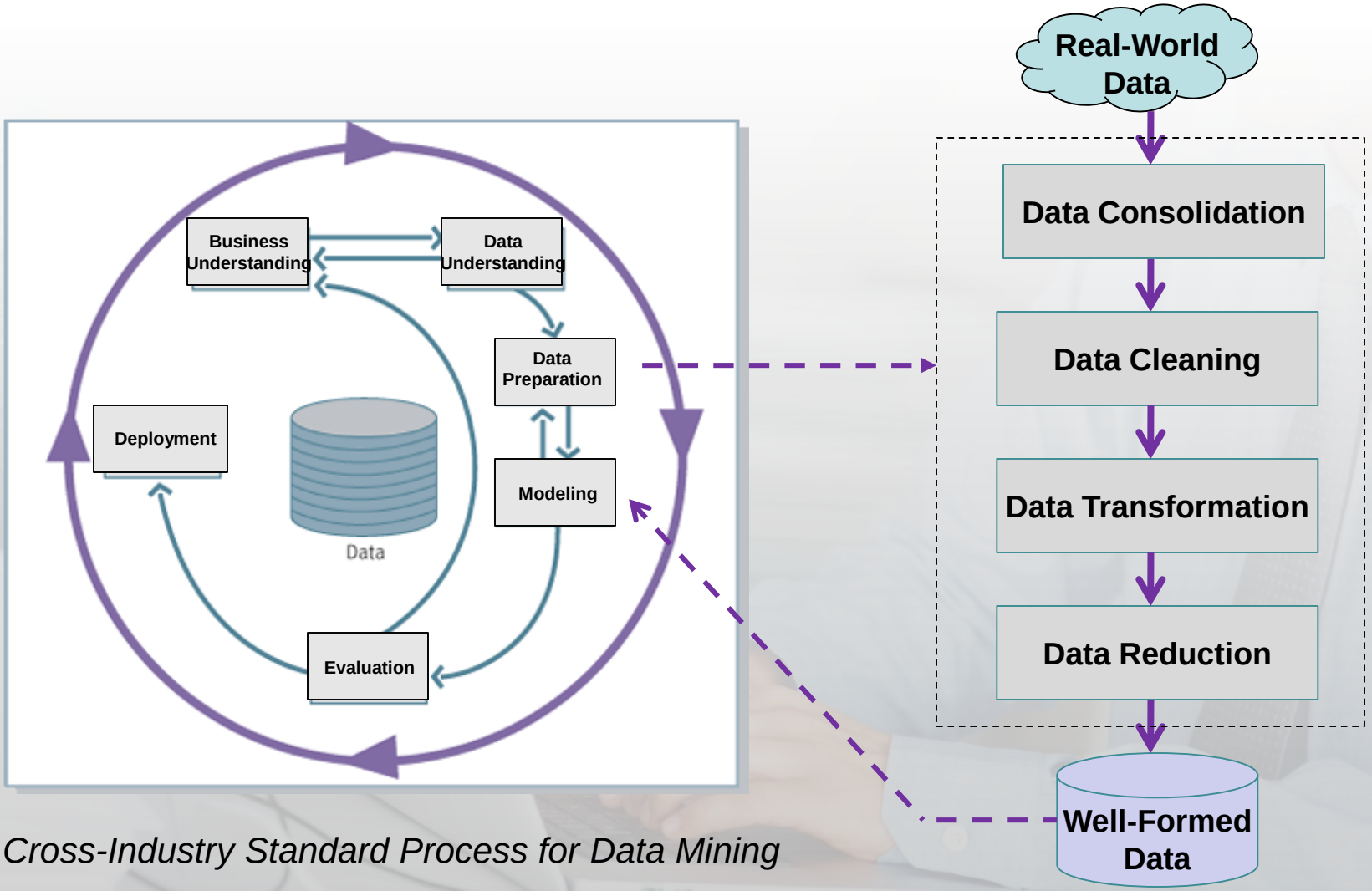
social Media Data: One Commonality

Data Mining: Defined

Discovering meaningful patterns from large data sets using pattern recognition technologies.

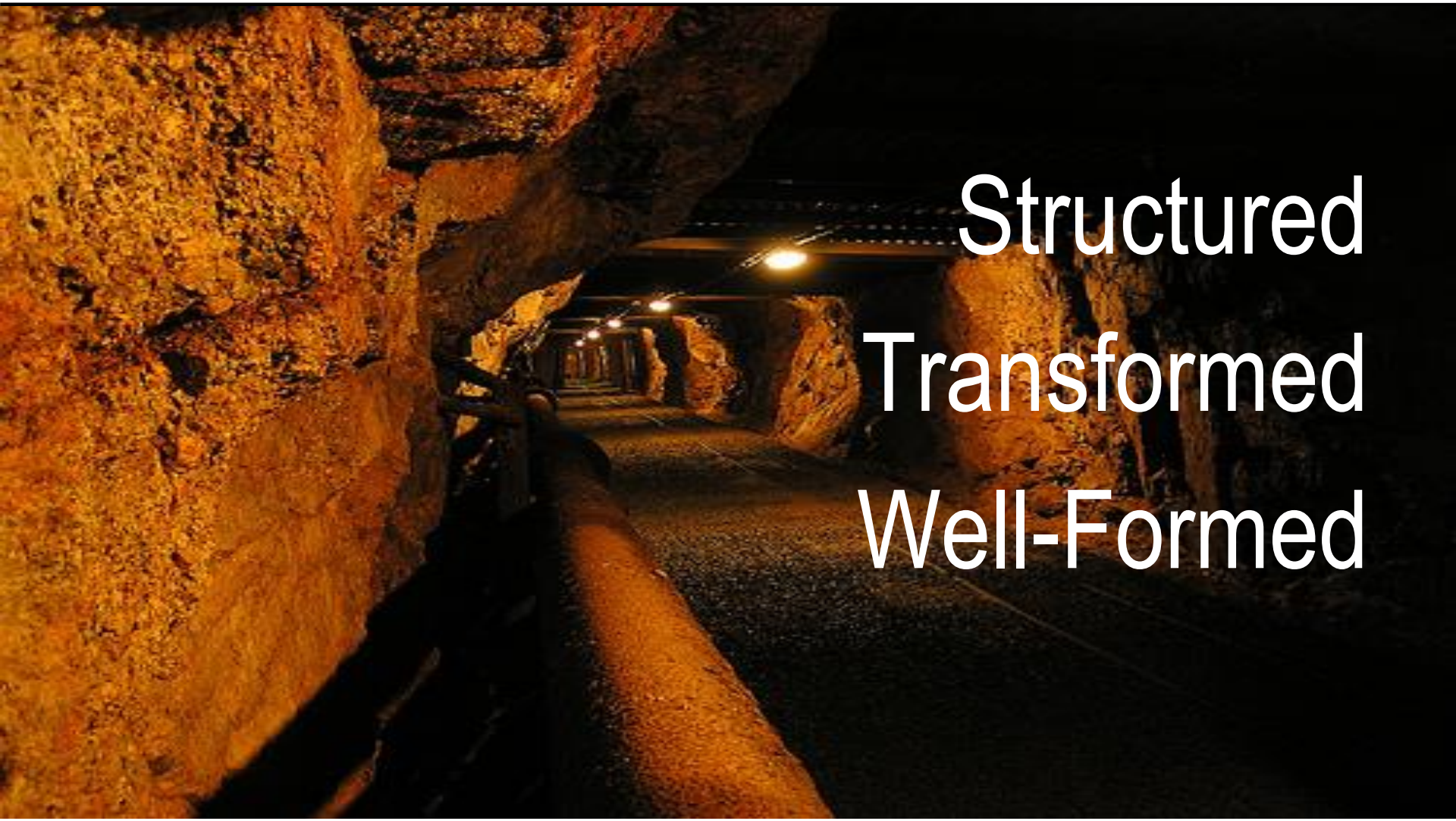


Data Mining: CRISP-DM



Cross-Industry Standard Process for Data Mining

Data Mining:
General Data Assumptions



Data Mining: Simple Example

Market Basket Analysis: Applications

- Cross Selling
- Product Placement
- Affinity Promotion
- Customer Segmentation Analysis

25

Market Basket Analysis: The Ole Beer and Diaper Legend…

Transaction	Item
1	Bread
1	Milk
2	Beer
2	Bread
2	Diapers
2	Eggs
3	Beer
3	Cola
3	Diapers
3	Milk
4	Beer
4	Bread
4	Diapers
4	Milk
5	Bread
5	Cola
5	Diapers
5	Milk

Binary Representation

Transaction	Beer	Bread	Cola	Diapers	Eggs	Milk
1	0	1	0	0	0	1
2	1	1	0	1	1	0
3	1	0	1	1	0	1
4	1	1	0	1	0	1
5	0	1	1	1	0	1

Goal: Empirically determine those *itemsets* that occur frequently together in a set of transactions, producing a set of *Association Rules* of the form LHS -> RHS

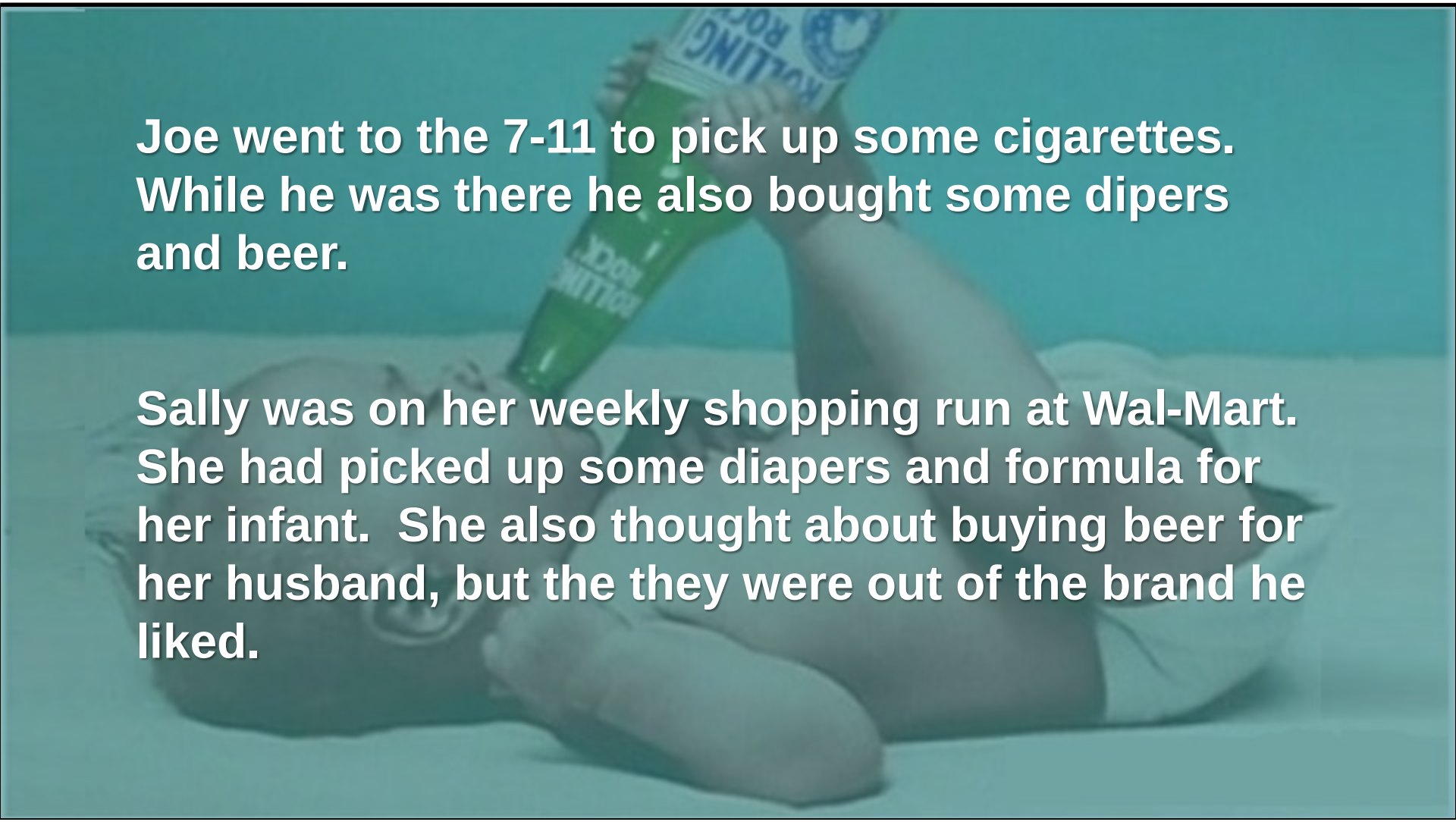
Market Basket Analysis: The Ole Beer and Diaper Legend…

Transaction	Beer	Bread	Cola	Diapers	Eggs	Milk
1	0	1	0	0	0	1
2	1	1	0	1	1	0
3	1	0	1	1	0	1
4	1	1	0	1	0	1
5	0	1	1	1	0	1

Concept	Definition	Example
Itemset	A specific collection of items in transaction	{Diapers, Beer}
Support Count	Number of transactions with itemset	Support {Diapers,Beer} = 3
Transactions	No of transactions = N	N=5
Association Rule	Implication rule of form LHS->RHS where LHS & RHS are itemsets	{Diapers} -> {Beer}
Rule Support	No. of times rule appears in dataset #tuples(LHS & RHS}/N	3/5 = .6
Rule Confidence	No. of times RHS occurs in transactions with LHS #tuples(LHS, RHS)/#tuples(LHS)	3/4 = .75

Rule Lift	Strength of Association over random co-occurrence of LHS and RHS
	#tuples(LHS,RHS}/N)/(#tuples(LHS)/N x #tuples(RHS)/N)
	Confidence(RHS/LHS)/Support(RHS)
	Support(LHS,RHS)/Support(LHS)xSupport(RHS)
		(3/5)/((4/5)x(3/5)) = 1.25

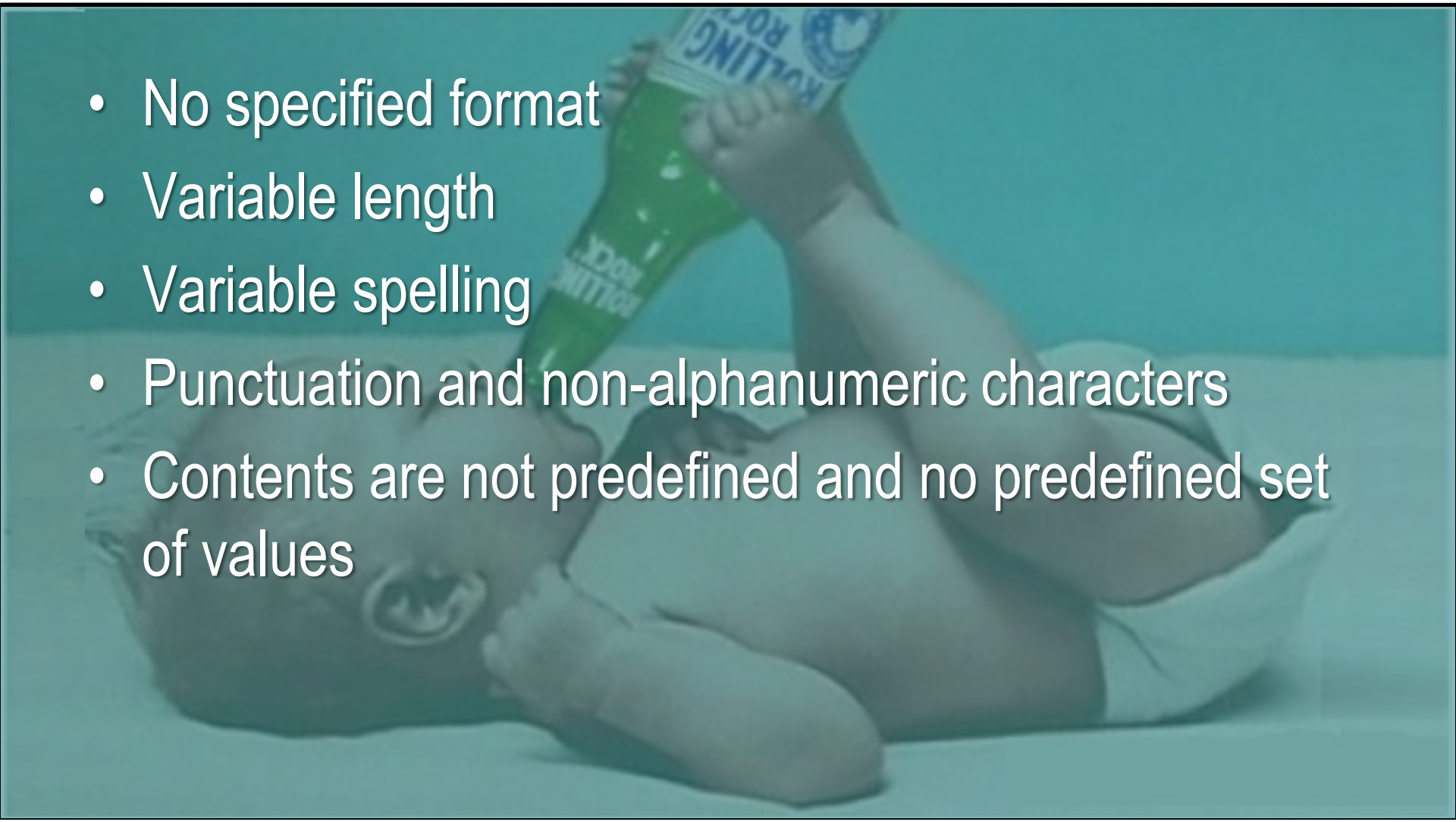
Market Basket Analysis: What if it were a text analysis problem?



Joe went to the 7-11 to pick up some cigarettes. While he was there he also bought some dipers and beer.

Sally was on her weekly shopping run at Wal-Mart. She had picked up some diapers and formula for her infant. She also thought about buying beer for her husband, but the they were out of the brand he liked.

Market Basket Analysis: What if it were a text analysis problem?



- No specified format
- Variable length
- Variable spelling
- Punctuation and non-alphanumeric characters
- Contents are not predefined and no predefined set of values

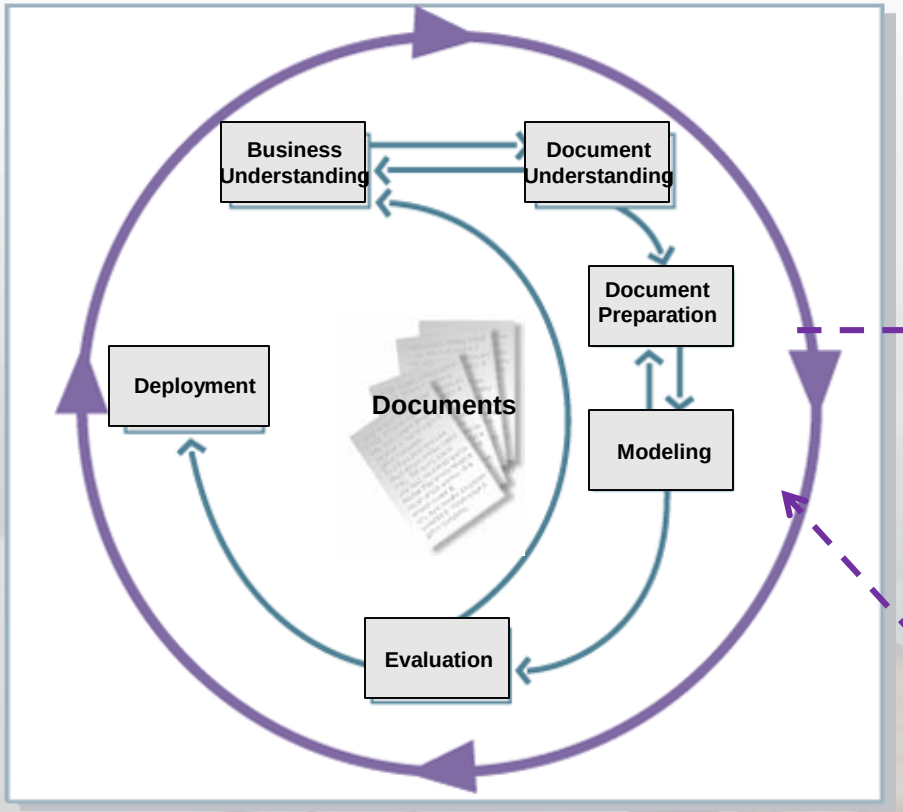
Text Mining (aka Text Analytics): Defined

Using natural language processing & data mining to discover patterns in a collection of “documents”

Text Mining: Document Collections

- Word Documents
- PDFs
- Emails
- IM Chat
- Web Pages
- Blogs
- Tweets
- Open ended surveys
- Transcripts of Helpline calls

Text Mining: CRISP-Like Processes



Real-World Text Data

Document Consolidation

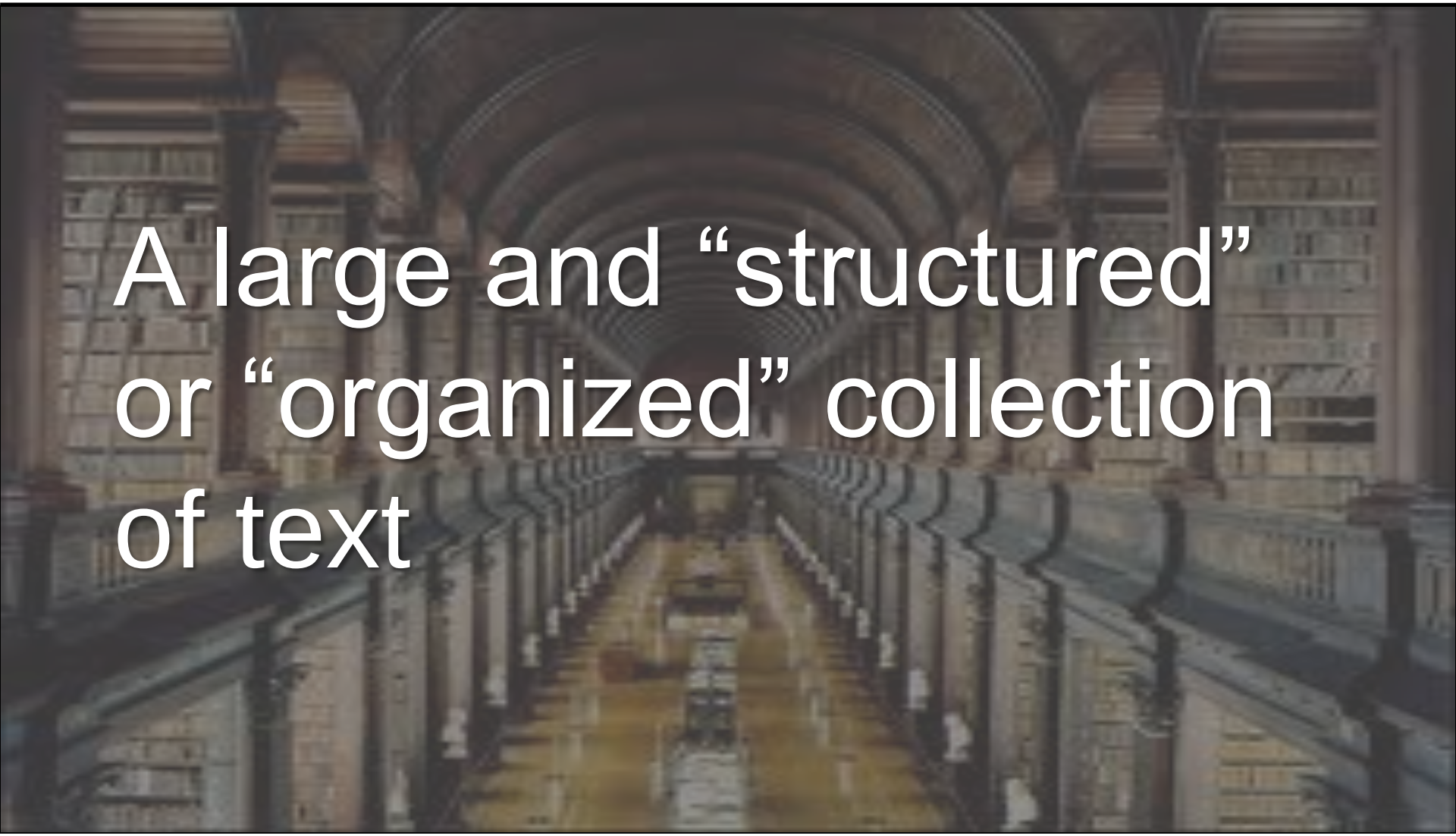
Establish the Corpus

Corpus Refinement (Token, Stem, Stop…)

Feature Selection & Weighting

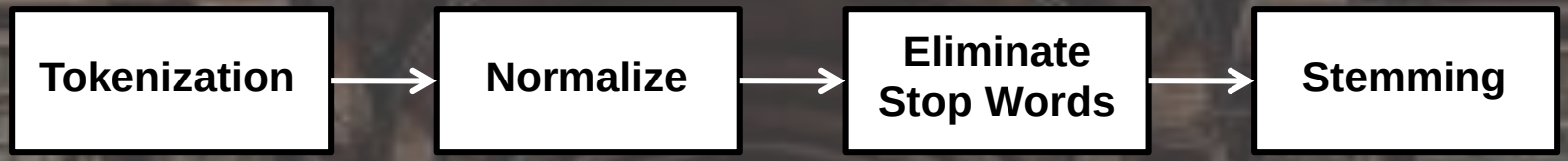
Doc-Term Matrix*

Text Mining:
Creating the corpus



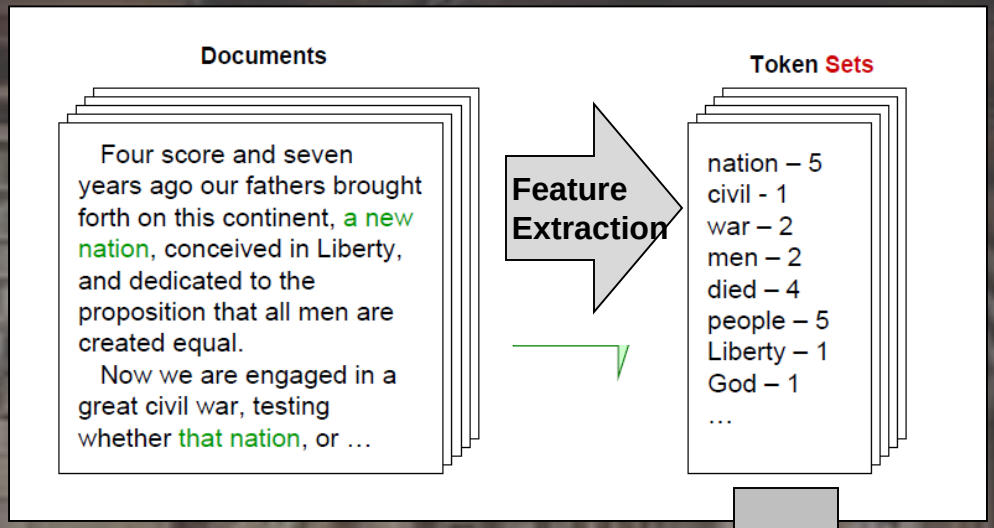
Text Mining Process: Corpus Refinement

Common representation of tokens within and between documents



- Tokenization —Parse the text to generate terms. Sophisticated analyzers can also extract phrases from the text.
- Normalize — Convert them to lowercase.
- Eliminate stop words — Eliminate terms that appear very often (e.g. the, and, …).
- Stemming — Convert the terms into their stemmed form—remove plurals and different word forms (e.g. achieve, achieves, achieved – achiev) [note: word about synonyms – WordNet Synset]

Text Mining Processes: Feature Extraction & Weighting



"Bag of Words, Terms or Tokens"

Vector Representation:
Word, Term or Token/Doc Matrix

	Token1	Token2	Token3	Token4	…
Doc1	1	2	2	4	
Doc2	4	2	3	0	
Doc3	1	1	1	0	
Doc4	1	1	1	2	
…					

"Bag of Words" (BOW) or Vector Space Model (VSM): Words or Tokens are attributes and documents are examples

Text Mining Processes: Transforming Frequencies

- Binary Frequencies: tf =1 for tf>0; otherwise 0
- Term Frequencies: tf(i,j)/Sum of tf(i,j) in Doc K
- Log Frequencies: 1 + log(tf) for tf>0; otherwise 0
- Normalized Frequencies: Divide each frequency by SQRT of Sum of Squares of the frequencies within the vector (column)
- Term Frequency–Inverse Document Frequency
  - TF * IDF
  - Inverse Document Frequency: log(N/(1+D)) where N is total number of docs and D is number with term

Text Mining Processes: Twitter Example – Problem Features

- Each tweet <= 140 characters (avg. 10-15 words/message)
- Heavy presence of non-alpha symbols, abbrevs, misspellings and slang
- Tweets often include retweets (original tweet repeated)

Text Mining Processes: Twitter Example

One of the things I love and adore about Twitter … is how its open API has lit a fierce fire of innovation when it comes to analytics. Anyone and their brother and ma-in-law can develop a tool, and they have! Much to the benefit of the rest of us.

Occam's Razor
by Avinash Kaushik

Text Mining Processes: Twitter Example – Twitter API

Get Search

- http://*search.twitter.com/search.json?q=<query>*
- search.twitter.com/search.json?q=Obama&rpp=100&page=5


```
{"completed_in":0.073,"max_id":288352119608725504,"max_id_str":"288352119608725504","next_page":"?page=6&max_id=288352119608725504&q=Obama&rpp=100","previous_page":"?
page=4&max_id=288352119608725504&q=Obama&rpp=100","page":5,"query":"Obama","refresh_url":"?since_id=288352119608725504&q=Obama","results":[{"created_at":"Mon, 07 Jan
2013 18:31:17 +0000","from_user":"reprose","from_user_id":38468522,"from_user_id_str":"38468522","from_user_name":"Chapin
Rose","geo":null,"id":288352119608725504,"id_str":"288352119608725504","iso_language_code":"en","metadata":
{"result_type":"recent"},"profile_image_url":"http:\/\/a0.twimg.com\/profile_images\/202161439\/chapin_head_shot_normal.jpg","profile_image_url_https":"https:\/\/si0.
twimg.com\/profile_images\/202161439\/chapin_head_shot_normal.jpg","source":"&lt;a
href=&quot;http:\/\/www.facebook.com\/twitter&quot;&gt;Facebook&lt;\/a&gt;","text":"Madigan's spokesan just said no pension votes today...house returns at 2p.m. in
the meantime, what do people... http:\/\/t.co\/rjLA0t5h","to_user":null,"to_user_id":0,"to_user_id_str":"0","to_user_name":null},{"created_at":"Mon, 07 Jan 2013
18:31:16 +0000","from_user":"Paulj567","from_user_id":34306288,"from_user_id_str":"34306288","from_user_name":"Paul Elliott
Johnson","geo":null,"id":288352118199439360,"id_str":"288352118199439360","iso_language_code":"en","metadata":
{"result_type":"recent"},"profile_image_url":"http:\/\/a0.twimg.com\/profile_images\/2337344850\/8yakdfme4u31uw1af0ne_normal.jpeg","profile_image_url_https":"https:\/
\/si0.twimg.com\/profile_images\/2337344850\/8yakdfme4u31uw1af0ne_normal.jpeg","source":"&lt;a
href=&quot;http:\/\/www.tweetdeck.com&quot;&gt;TweetDeck&lt;\/a&gt;","text":"Agree! RT @ali I'm convinced President Obama wants us to come close to taking out Hagel
and use it to paint the midterm \"Radical GOP.\"","to_user":null,"to_user_id":0,"to_user_id_str":"0","to_user_name":null},{"created_at":"Mon, 07 Jan 2013 18:31:16
+0000","from_user":"wilxTV","from_user_id":16456122,"from_user_id_str":"16456122","from_user_name":"wilxTV","geo":null,"id":288352117859696640,"id_str":"2883521178596
96640","iso_language_code":"en","metadata":
```


Text Mining Processes: Twitter Example – JSON

- JSON (JavaScript Object Notation)
- Lightweight, Text-Based, Data-Interchange Format
- Built on Two Structures:
  - A collection of name:value pairs. In various languages, this is realized as an *object*, record, struct, dictionary, hash table, keyed list, or associative array.
  - An ordered list of values. In most programming languages, this is realized as an *array*, vector, list, or sequence

Text Mining Processes: Twitter Example – Establish Corpus

Query

search.twitter.com/search.json?q=%3A)+feel+ feeling&rpp=100&page=5

search.twitter.com/search.json?q=%3A(+feel+feeling&rpp=100&page=5

API

Result

{…
results:[
{iso_language_code: en,
to_user_name: Andrea,
…'
text: u”Love is everything in this world! Its a feeling like no other. I can't wait 2 feel that emotion again..but patience is key :-)”
…
created_at: Thu, 18 Oct 2012 20:11:22 +0000,
…'}}
…}

Text Mining Processes: Twitter Example – Simple Question

Are there any language differences between

“feeling” tweets containing ☺ and ☹ symbols?

Text Mining Processes: Twitter Example – Establish Corpus

text: value pairs in JSON object

Remove RTs

1. Love is everything in this world! Its a feeling like no other. I can't wait 2 feel that emotion again..but patience is key :-)
2. @mzxAmaZiiN Whats up ma. How ya feeling? Lemme make that soul feel better. :)
3. "..I've got a good feeling about today :P Something makes me feel i might make a sale or two :) *fingers crossed* #etsy #shop #seller #cra...
4. @IWontForgetDemi Awww poor thing :(hate feeling sick! Hope you feel better soon
5. @IzabelaLeafsfan no the worst feeling ever is when u feel like total crap. cuz u think no one luvs u :(
6. …

Text Mining Processes: Twitter Example – Doc Preparation

Tokenization → **Normalize** → **Eliminate Stop Words** → **Stemming**

Tweet:
Love is everything in this world! Its a feeling like no other. I can't wait 2 feel that emotion again..but patience is key :-)

Words:
['Love', 'is', 'everything', 'in', 'this', 'world!', 'Its', 'a', 'feeling', 'like', 'no', 'other.', 'I', "can't", 'wait', '2', 'feel', 'that', 'emotion', 'again..but', 'patience', 'is', 'key', ':-)']

Tokens:
['Love', 'is', 'everything', 'in', 'this', 'world', '!', 'Its', 'a', 'feeling', 'like', 'no', 'other.', 'I', 'ca', "n't", 'wait', '2', 'feel', 'that', 'emotion', 'again..but', 'patience', 'is', 'key', ':', '-', ')']

Text Mining Processes: Twitter Example – Doc Preparation

Tokenization → Normalize → Eliminate Stop Words → Stemming

- **Normalize**
  - ['love', 'is', 'everything', 'in', 'this', 'world!', 'its', 'a', 'feeling', 'like', 'no', 'other.', 'i', "can't", 'wait', '2', 'feel', 'that', 'emotion', 'again..but', 'patience', 'is', 'key', ':-)']
- **Alpha**
  - ['love', 'is', 'everything', 'in', 'this', 'its', 'feeling', 'like', 'no', 'wait', 'feel', 'that', 'emotion', 'patience', 'is', 'key']
- **Remove Stopwords**
  - ['love', 'everything', 'feeling', 'like', 'wait', 'feel', 'emotion', 'patience', 'key']
- **Stemming**
  - ['love', 'everyth', 'feel', 'like', 'wait', 'feel', 'emot', 'patienc', 'key']

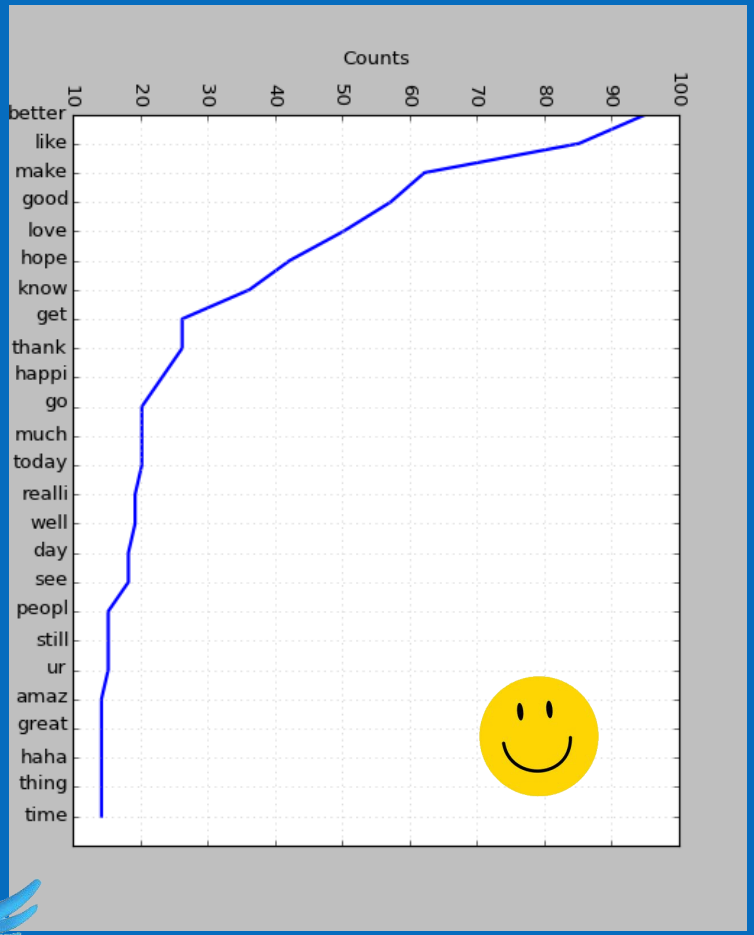
Text Mining Processes: Twitter Example – Analysis

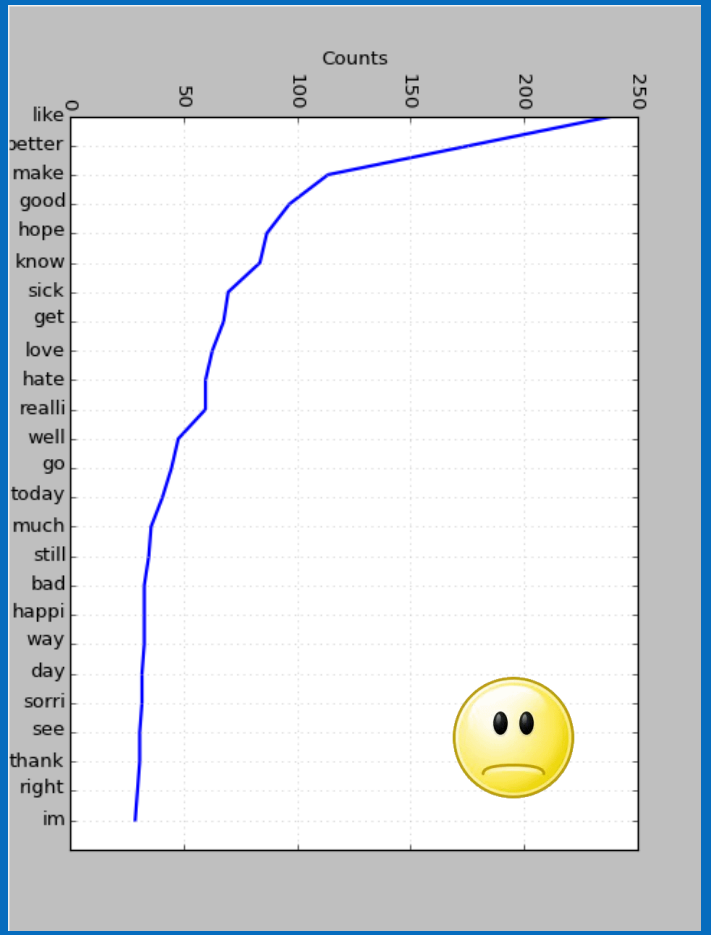
Item	Collection	List	Set	Lex Div	Aver Len in Chars	Aver No/Tweets(w/o)
Tweets	HF	498	-	-	108	-
	SF	499	-	-	100	-
Tweets w/o "RT"	HF	409	-	-	105	-
	SF	429	-	-	98	-
Words	HF	8077	2346	3	4	20
	SF	8149	2073	4	4	19
Alpha (lower)	HF	5622	1197	5	4	14
	SF	5733	1041	6	4	13
Alpha w/o Stops	HF	3400	1092	3	5	8
	SF	3469	936	4	5	8
Stems	HF	3400	978	3	4	8
	SF	3469	844	4	4	8
Stems w/o "feel"	HF	2619	977	3	4	6
	SF	2635	843	3	4	6

Text Mining Processes: Twitter Example – Analysis

Corpus	Av Word Len	Aver Sent Ln	Lexical Diversity
HF Tweets	4	20	3
SF Tweets	4	19	4
austen-emma.txt	4	21	26
austen-persuasion.txt	4	23	16
austen-sense.txt	4	23	22
bible-kjv.txt	4	33	79
blake-poems.txt	4	18	5
bryant-stories.txt	4	17	14
burgess-busterbrown.txt	4	17	12
carroll-alice.txt	4	16	12
chesterton-ball.txt	4	17	11
chesterton-brown.txt	4	19	11
chesterton-thursday.txt	4	16	10
edgeworth-parents.txt	4	17	24
melville-moby_dick.txt	4	24	15
milton-paradise.txt	4	52	10
shakespeare-caesar.txt	4	11	8
shakespeare-hamlet.txt	4	12	7
shakespeare-macbeth.txt	4	12	6
whitman-leaves.txt	4	35	12

Text Mining Processes: Twitter Example – Analysis

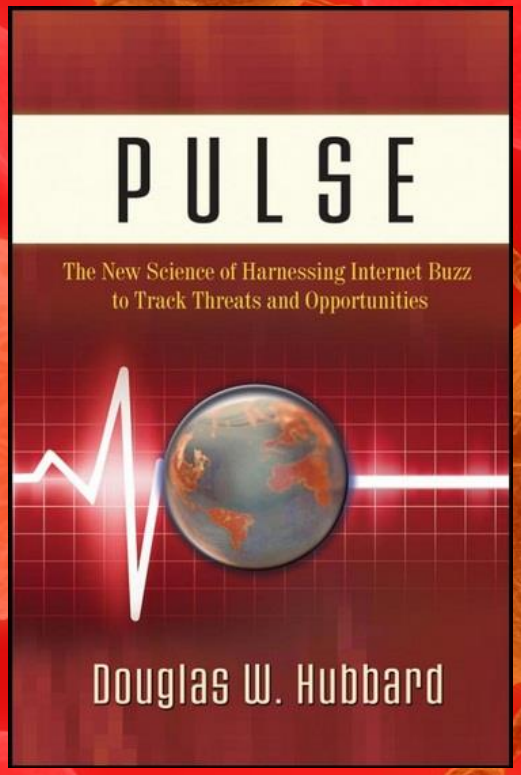




Text Mining Processes: Twitter Example – Doc-Term Matrix

3538	Total	Type	216	166	111	86	83	76	66	64	58	58	…	4	4	4	4	4	4	4	4	4	4
Total	Tweets	Face	like	better	make	good	hope	know	get	sick	hate	realli	…	ye	rain	quit	chang	stress	happier	longer	cheer	without	everyth
5	Tweet1	HF	0	1	0	0	1	0	0	0	0	0	…	0	0	0	0	0	0	0	0	0	0
0	Tweet2	HF	0	0	0	0	0	0	0	0	0	0	…	0	0	0	0	0	0	0	0	0	0
7	Tweet3	HF	0	0	0	0	0	0	0	0	0	0	…	0	0	0	0	0	0	0	0	0	0
3	Tweet4	HF	0	0	0	0	0	0	0	0	0	1	…	0	0	0	0	0	0	0	0	0	0
2	Tweet5	HF	0	0	1	0	0	0	0	0	0	0	…	0	0	0	0	0	0	0	0	0	0
1	Tweet6	HF	0	1	0	0	0	0	0	0	0	0	…	0	0	0	0	0	0	0	0	0	0
5	Tweet7	HF	1	0	0	0	0	0	0	0	0	0	…	0	0	0	0	0	0	0	0	0	0
7	Tweet8	HF	0	0	0	0	1	0	0	0	0	0	…	0	0	0	0	0	0	0	0	0	0
5	Tweet9	HF	0	1	0	0	1	0	0	1	0	0	…	0	0	0	0	0	0	0	0	0	0
2	Tweet10	HF	0	0	0	0	0	0	0	0	0	0	…	0	0	0	0	0	0	0	0	0	0
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
5	Tweet829	SF	0	0	0	0	0	0	0	0	0	0	…	0	0	0	0	0	0	0	0	0	0
5	Tweet830	SF	1	0	0	0	0	0	0	0	0	0	…	0	0	0	0	0	0	0	0	0	0
3	Tweet831	SF	0	0	0	0	0	1	0	0	0	0	…	0	0	0	0	0	0	0	0	0	0
4	Tweet832	SF	0	0	1	0	0	0	0	0	0	0	…	0	0	0	0	0	0	0	0	0	0
3	Tweet833	SF	1	0	0	0	0	0	0	0	0	0	…	0	0	1	0	0	0	0	0	0	0
3	Tweet834	SF	0	1	1	0	0	0	0	0	0	0	…	0	0	0	0	0	0	0	0	0	0
5	Tweet835	SF	1	0	0	0	0	1	0	0	1	0	…	0	0	0	0	0	0	0	0	0	0
3	Tweet836	SF	0	0	0	0	0	0	0	1	1	0	…	0	0	0	0	0	0	0	0	0	0
5	Tweet837	SF	0	1	0	0	0	0	1	0	0	0	…	0	0	0	0	0	0	0	0	0	0
3	Tweet838	SF	1	0	0	0	0	0	0	0	0	0	…	0	0	0	0	0	0	0	0	0	0

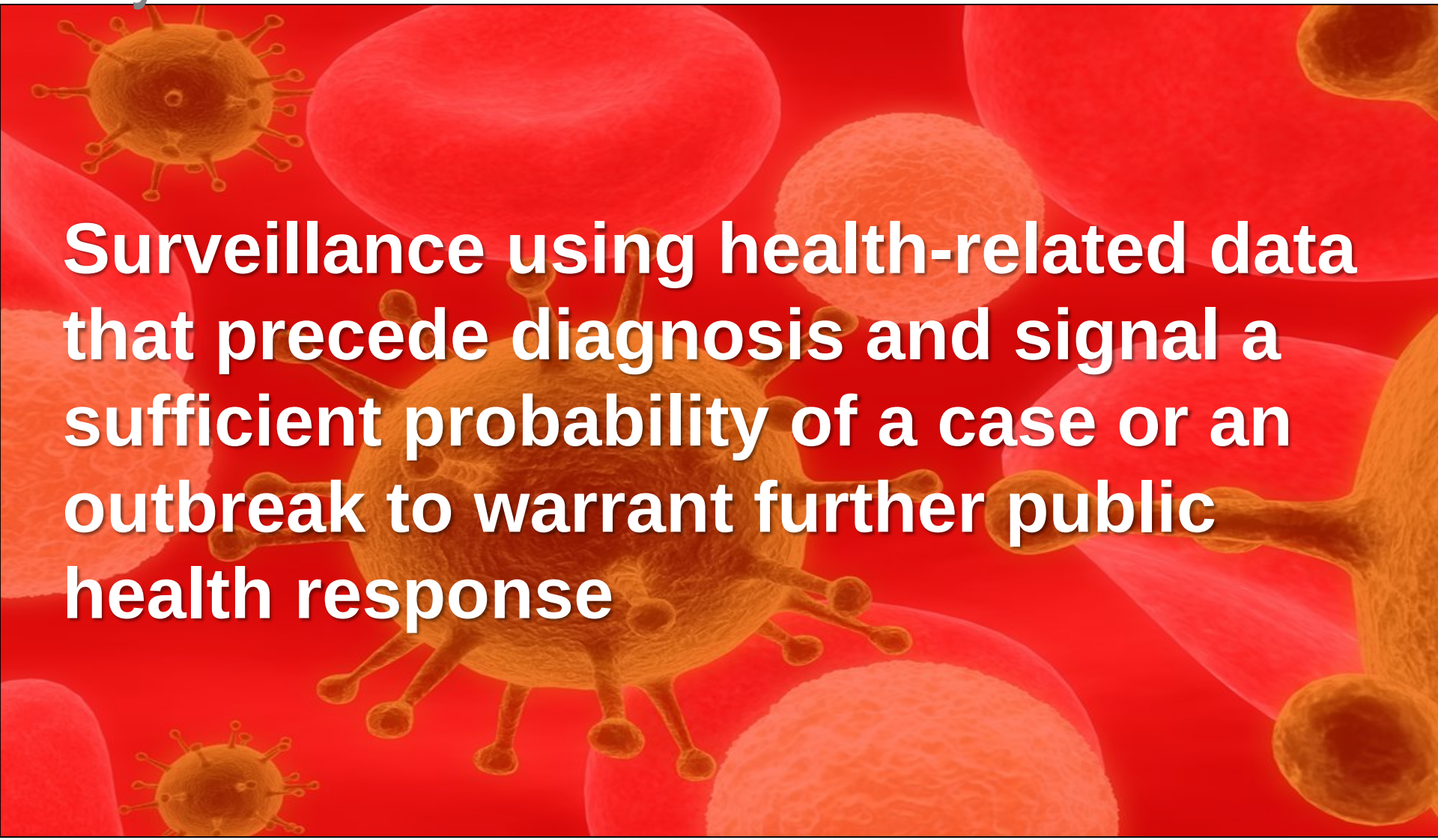
Prediction
Information + Epidemiology =



Infodemiology

Monitoring and analyzing queries from Internet search engines or peoples' status updates on microblogs for *syndromic surveillance* to predict disease outbreaks

Prediction
Syndromic Surveliance



Monitoring the Onset of the Flu Season
The Official Standard Way

Sentinel Physician

ILI - influenza-like-illness

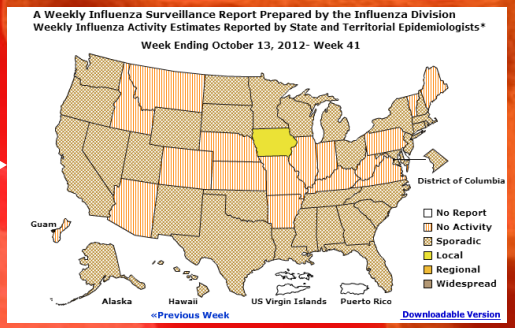
Public Health Authority

CDC

ILI Report

Aggregated Data

Public Reports



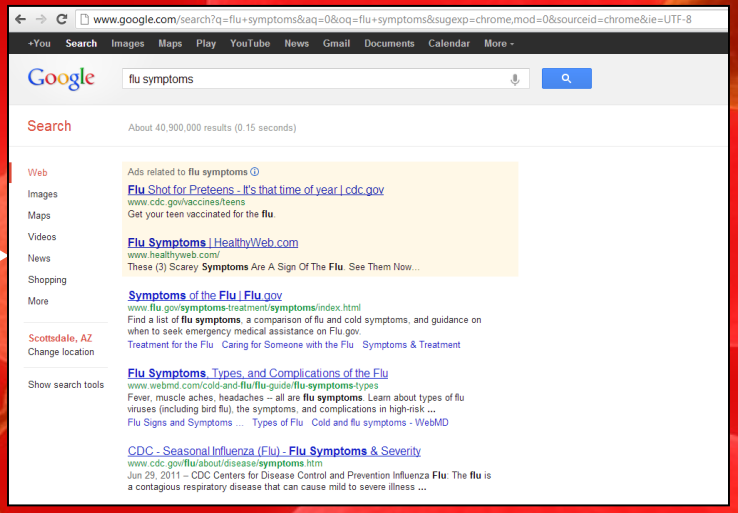
Costly
1-2 Week Lag

Prediction
Infodemiology

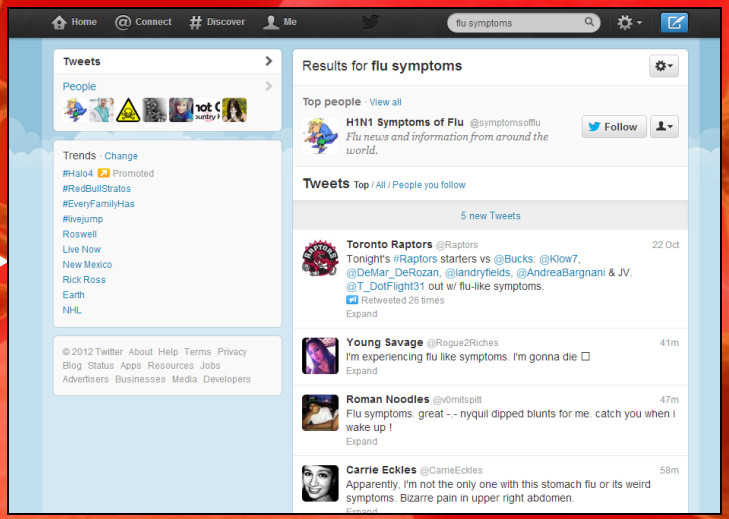
What is the first thing some people do before

they see a doctor or take OTC medicines?

search



tweet



Prediction - What are the terms, keywords, phrases…?

What is the first thing people do before they

see a doctor or take OTC medicines?

search

tweet

- **Flu**
- **Flu symptoms**
- **H1N1**
- **Swine Flu**
- **Cold**
- **Fever**
- **Headache**
- **…**

Prediction Infodemiology Studies

Authors	Title	Date (M-Y)	Type	Data Source	Dependent Variables	Explanatory Variables	Model	Results
Eysenbach	Infodemiology: Tracking Flu-Related Searches on the Web for Syndromic Surveillance	Nov-06	Search	Impressions and clicks from a Google Ad displaying the question:"Do you have the flu? Fever, Chest discomfort, Weakness, Aches, Headache, Cough." Covers Oct. 2004-May 2005	Number of influenza lab tests, number of positive influenza lab tests and number of ILI reports from Sentinel GPs reported by PHA in Canada	Number of clicks on Google Ad dealing with influenza	Linear correlation between clicks and 3 measures of publically reported influenza	Correlation of .81 with ILI data and .90 with lab tests and positive cases.
Hulth et al.	Web Queries as a Source for Syndromic Survelliance	Feb-09	Search	Queries to Swedish heath advice site (Varguiden) from June 2005 to June 2007	Weekly Lab diagnosed cases of influenza and % of ILI reports of influenza from GPs	Weekly ratio of influenza related Web queries to all queries at Varguiden site	Two linear regression models, one predicting lab diagnosed cases and the other ILI reports where the explanatory variable based based on a composite of the best predicting query terms	Average R squared was .90 for the two years.
Ginsberg et al.	Detecting influenza epidemics using search engine query data	Feb-09	Search	Google Search: Historical web logs of ILI related Google Searchs 2003-2008	% US ILI Visits in week reported to CDC	ILI-related search queries	logit(P)=b0 + b1xlogit(Q)+e where P is % visits and Q is normalized number of queries	Mean correlation of .90 between P & Q for 9 CDC US healthcare regions.
Lampos & Cristianini	Tracking the flu pandemic monitoring the Social Web	Jun-10	Tweets	Twitter: 24 weeks of Twitter corpus in UK from June '09 to Dec'09	Weekly ILI UK Health Protection Agency smoothed for daily values	Average daily "flu-score" for all tweets. Flu score for single tweet is proportion of all ILI "stem" markers (ngrams) that appear in the tweet.	linear least squares regression time series of HPA flu rates on aggregated tweet flu-scores	Average correlation for 5 UK health care regions was about .92.
Culotta	Towards detecting influenza epidemics by analyzing Twitter messages	Jul-10	Tweets	575K flu-related Twitter messages and % ILI reports from CDC for period Feb 2010 to Apr 2010.	% ILI weekly reports from CDC for specified period	% of messages reporting an ILI or related symptom (based on detailed classification and statistical procedures to determine whether ILI or not)	Compares simple logit model with multiple regression model having different counts for separate Tweet keywords & phrases	Aggregating keyword frequencies using separate keywords (multi-reg) works better than single aggregated (simple logit) predictor. Simple BOW classifier can be used to filter ILI messages. Achieved r=.78 for 5 weeks of validation data.

Prediction Infodemiology Studies

Authors	Title	Date (M-Y)		Data Source	Dependent Variables	Explanatory Variables	Model	Results
Acherekar	Predicting Flu Trends using Twitter Data	Apr-11	Tweets	ILI "influenza" visits to Sentinel physicians reported weekly by CDC from Oct 2009 to Oct 2010 and tweets mentioning "flu-like" symptoms during same time period	ILI influenza visits	Previous weeks ILI visits and aggregate number of tweets mentioning flu-like symptoms	Auto-regression model with: ILI(t) = a1*ILI(t-2) + b*Tweets(t) + e	Analysis shows time-lagged ILI is not a strong predictor but the aggregate number of tweets with flu-like systems is very strong with an r=.9846.
Chan et al.	Using Web Search Query Data to Monitor Dengue Epidemics: A New Model for Neglected Tropical Disease Surveliance	May-11	Search	Google Queries in select countries from 2003-10 including Bolivia, Brazil, India, Indonesia and Singapore	Official weekly dengue case count from Ministry of Heath or WHO	Computed daily counts of queries referencing Dengue fever for separate countries	O=b0+b1Q+e where O is official weekly count of cases in specified countries and S is dengue related search query in those countries	Very strong correlation (~.9) except for Singapore (.82).
Signorini et al.	The Use of Twitter to Track Levels of Disease Activity and Public Concern in the US during the Influenza A H1N1 Pandemic	May-11	Tweets	Several panels of tweets and ILI reports. Prediction panel includes weekly ILI %s and US tweets from Oct 2009 thru May 2010.	% ILI weekly reports from CDC for specified period. Reports include nationwide and 10 regional reports.	Fraction of influenza related tweets for total US and CDC health regions. Utilizes complex Support Vector Machine using term frequency statistics.	Support Vector Regression utilizing SVM feature sets (collection of terms occuring 10X per week) to estimate weekly ILI.	Strong SVR fit for both national and regional figures.
Paul	You are what you tweet: Analyzing Twitter for Public Health	Jul-11	Tweets	Covers a number of ailments including influenza. Based on 2 billion tweets from Many 2009 to October 2010.	% ILI weekly reports from CDC for specified period	Utilizes an Ailment Topic Aspect Model (ATAM) to determine the type of ailment associated with a tweet (in this case Flu)	Correlation between the probability the "flu ailment" designation for each week with the ILI rate in the US.	Two types of ATAM/ATAM+ models were compared. The correlations with ILI were .934 and .958.
Lampos et al.	Flu detector - Tracking epidemics on Twitter	Sep-12	Tweets	Twitter: 40 weeks of Twitter corpus in UK from June '09 to Mar '10	Weekly ILI UK Health Protection Agency smoothed for daily values	Sum across all tweets of daily "flu-score" for each tweet based on number of ILI "stem" markers that appear in the tweet.	linear least squares regression time series of HPA flu rates on aggregated tweet flu-scores	Correlations around .90 for 3 UK health care regions.

Infodemiology
Example Prediction Analysis

Nowcasting Events from the Social Web with Statistical Learning," Lampos and Cristianini, ACM IS&T, 9/11

Twitter API

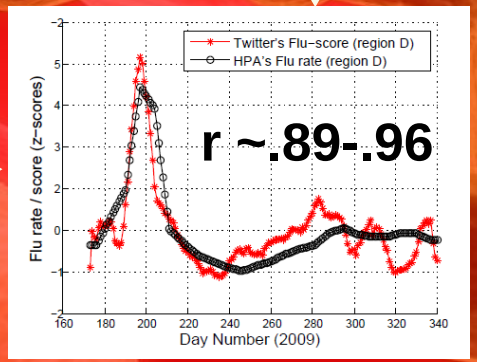
50M Tweets 5 UK Regions 6'09-12'09

N-Gram Stems ILI Markers

TABLE IV: 97 stemmed markers extracted by applying LASSO regionally. The markers are sorted in a descending order based on their weights (read horizontally, starting from the top-left corner).

lung	unwel	temperatur	like	headach	season
unusu	chronic	child	dai	appetit	stai
symptom	spread	diarrhoea	start	muscl	weaken
immun	feel	liver	plenti	antivir	follow
sore	peopl	nation	small	pandem	pregnant
thermomet	bed	loss	heart	mention	condit
high	group	tired	import	risk	carefulli
work	short	stage	page	diseas	recognis
servic	wors	case	similar	term	home
increas	exist	ill	sens	counter	better
cough	vomit	earli	neurolog	catch	onlin
fever	concern	check	drink	long	far
consid	ach	breath	flu	member	kidnei
mild	number	sick	throat	famili	water
read	includ	swine	confirm	need	nose
medic	phone	cancer	disord	unsur	suddenli
runni					

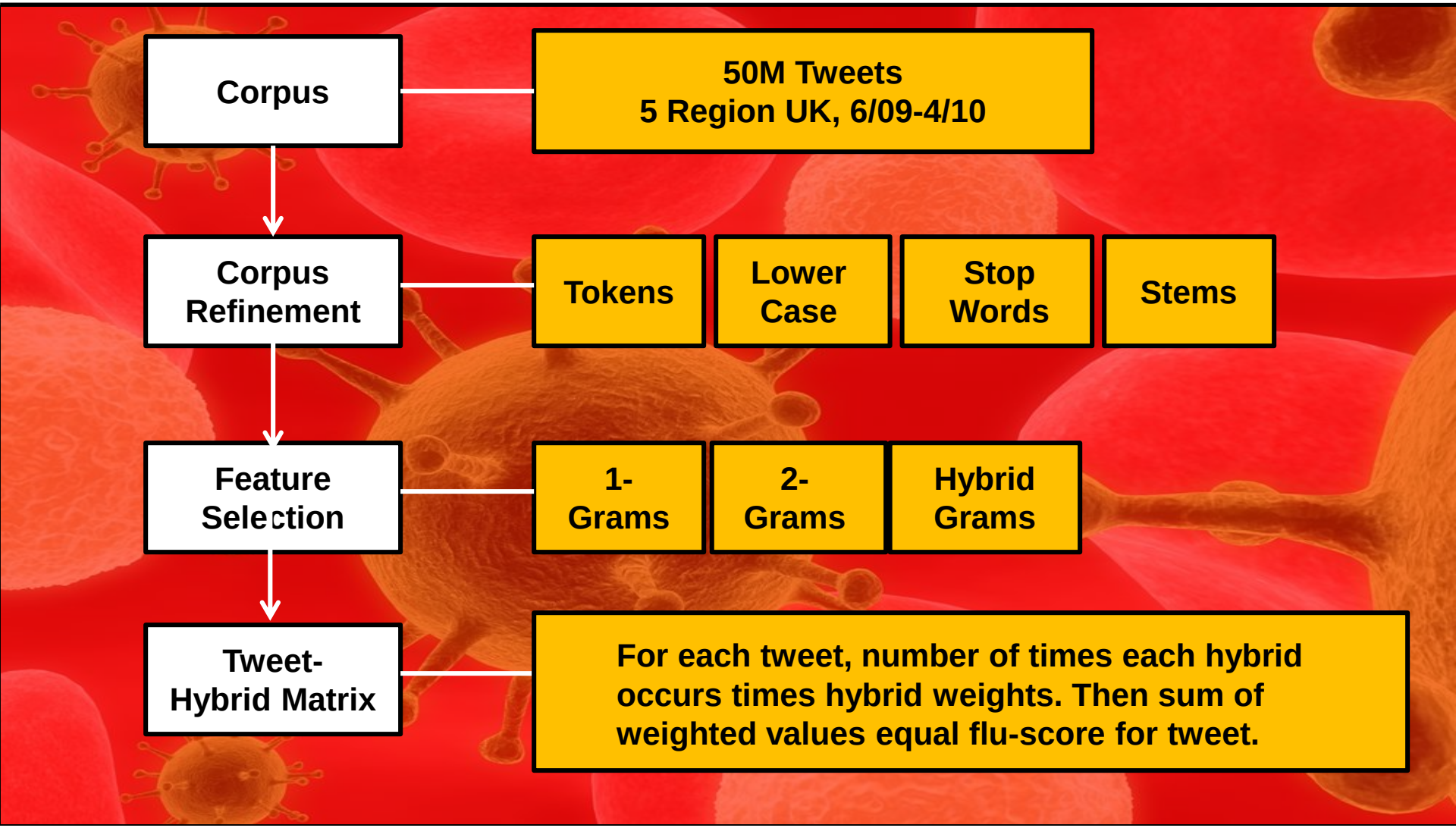
Avg. Weekly Wght. Flu-Score by Region (time t)



Weekly ILI Reports by health region (time t)

Health Protection Agency

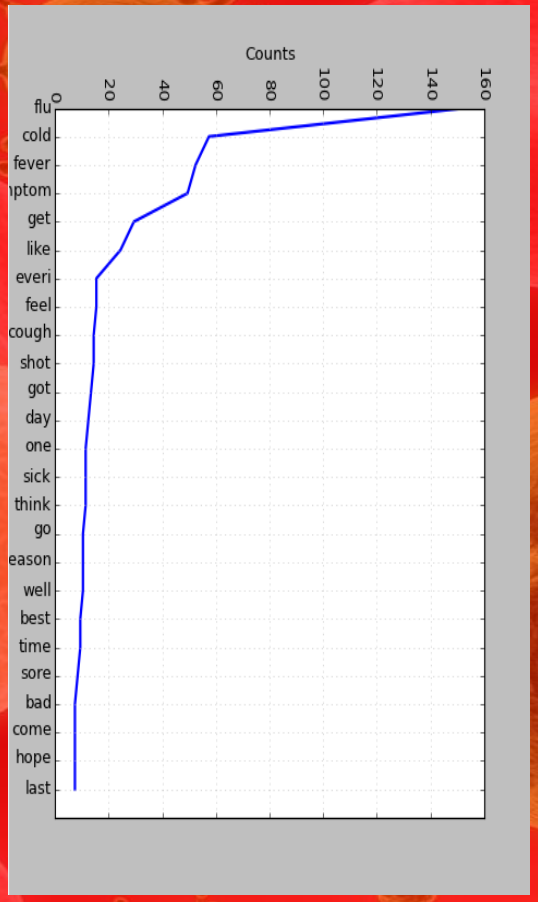
Infodemiology
Example Prediction Analysis



Infodemiology
Example Prediction Analysis

- flu :(
- *RT @OMGFacts: The US death toll from the 1918 flu epidemic was so high that it created a coffin shortage*
- *News_SwineFlu: #swineflu Delayed treatment for swine flu can lead to... http://t.co/ZSOjlvAW'*
- The last thing I want right now is the flu
- Much better with that lovely message from u,thank u! (heavy flu since 4 days ;
- *Supposedly only about 1% of people in the world present flu-like symptoms after flu shot. Unfortunately 66% of the people in my house do'*
- Back on track...Fuck you flu!
- Sounds like either the flu or Killian walked into the room.
- I'll try haha, I got the flu!
- *It costs an average company $135 a day for every employee sick at home with the flu. Encourage your employees to get vaccinated....*
- *i wouldn't recommend it, but a couple of days with violent stomach flu does make you appreciate the small things, nature's reset*
- *Is it possible to get the flu from the flu jab?*
- Sedatives + cold/flu + ana = can't get up without having to hold a wall for 2 minutes",

Infodemiology
Example Prediction Analysis (3)



Bigram	Frequency	Bigram	Frequency
(u'flu',u'shot')	11	(u'common',u'cold')	2
(u'flu',u'season')	8	(u'man',u'flu')	2
(u'flu',u'symptom')	5	(u'southeast',u'asia,')	2
(u'symptom',u'busters')	4	(u'please',u'go')	2
(u'mtm',u'blog')	4	(u'headache',u'since')	2
(u'sore',u'throat')	4	(u'get',u'flu')	2
(u'immune',u'system')	4	(u'every',u'single')	2
(u'season',u'\u2013')	4	(u"can't",u'get')	2
(u'every',u'symptom')	3	(u'gonna',u'get')	2
(u'since',u'last')	3	(u'stuffy',u'head')	2
(u'flu',u'jab')	3	(u'fever,',u'flu,')	2
(u'sore',u'throat,')	3	(u'feeling',u'like')	2
(u'feel',u'like')	3	(u'symptom',u'checker')	2
(u'jonas',u'brothers')	3	(u'runny',u'nose')	2
(u'flu',u'shots')	2	(u'riva',u'offering')	2
(u'+',u'sore')	2	(u'flu',u'pills')	2
(u'like',u"i'm")	2	(u'#nutrition',u'foods')	2
(u"didn't",u'go')	2	(u'cure',u'#colds')	2
(u"don't",u'feel')	2	(u'cold,',u'flu')	2
(u'offering',u'complimentary')	2	(u"i'm",u'gonna')	2
(u'flu-like',u'symptom')	2	(u'ear',u'infection')	2
(u'asia,',u'millions')	2	(u'flu,',u'sore')	2

Infodemiology
Example Prediction Analysis (3)

Table VIII. Feature Class U – 1-grams selected by Bolasso for Flu case study (Round 1 of 5-fold cross validation) – All weights (**w**) should be multiplied by 10^4

1-gram	w	1-gram	w	1-gram	w	1-gram	w	1-gram	w
acut	-1.034	cleav	0.735	hippocr	-6.249	properti	-0.66	speed	-0.286
afford	-0.181	complex	-0.499	holidai	-0.017	psycholog	-1.103	spike	0.145
allergi	-2.569	cough	0.216	huge	-0.33	public	0.212	stage	0.109
approv	-0.672	cruis	-1.105	***irrig***	**10.116**	radar	0.284	strength	0.873
artifici	2.036	daughter	0.187	item	-0.337	reach	0.247	strong	0.336
assembl	0.589	dilut	4.165	knock	0.261	reliev	-0.254	swine	1.262
asthmat	4.526	drag	0.098	lethal	-0.73	remain	-0.755	tast	0.13
attempt	0.375	erad	0.201	major	-0.367	rough	0.068	team	-0.031
behavior	-1.747	face	-0.008	medic	1.06	run	0.242	throat	0.07
better	0.066	fellow	0.542	member	0.354	rush	-0.159	tissu	0.533
bind	0.675	fluid	2.002	mercuri	-0.588	scari	0.198	transmit	1.352
blood	0.059	fuss	0.575	metro	-0.397	seal	-0.161	troop	0.532
boni	1.308	germ	0.211	mile	-0.081	season	-0.103	typic	0.585
bulg	-0.966	guilti	-0.608	miss	0.071	seizur	2.448	underli	0.774
caution	2.578	habit	0.619	nurs	0.223	self	0.127	unquot	8.901
cellular	-2.125	halt	1.472	perform	0.084	sik	-0.634	upcom	0.642
checklist	-1.494	harbour	-0.472	personnel	-1.451	site	0.042	wave	0.042
chicken	0.317	health	-0.241	pictur	-0.134	soak	0.413	wikipedia	0.824

Table IX. Feature Class B – 2-grams selected by Bolasso for Flu case study (Round 1 of 5-fold cross validation) – All weights (**w**) should be multiplied by 10^4

2-gram	w	2-gram	w	2-gram	w	2-gram	w
case swine	12.783	flu bad	6.641	need take	0.887	talk friend	-4.9
check code	6.27	flu jab	4.66	pain night	14.149	time knock	10.002
check site	0.568	flu relat	10.948	physic emotion	7.95	total cost	-11.582
confirm swine	**31.509**	flu symptom	7.693	sleep well	1.319	underli health	25.535
cough fit	7.381	flu web	-8.017	sore head	4.297	virus epidem	-28.204
cough lung	7.974	ground take	-15.208	spread viru	20.871	visit doctor	-12.327
cough night	16.73	health care	-0.636	stai indoor	5.482	weight loss	-0.447
die swine	9.722	healthcar worker	3.876	suspect swine	3.863	woke sweat	-33.133
effect swine	27.675	home wors	22.167	swine flu	1.153	wonder swine	11.5085
feel better	0.655	ion channel	9.755	symptom swine	5.895		
feel slightli	1.712	kick ass	-0.335	take care	0.382		

1-Grams

2-Grams

Table X. Feature Class H – Hybrid selection of 1-grams and 2-grams for Flu case study (Round 1 of 5-fold cross validation) – All weights (**w**) should be multiplied by 10^4

n-gram	w	*n*-gram	w	*n*-gram	w	*n*-gram	w
acut	-0.796	effect swine	19.835	medic	0.48	spike	0.032
afford	-0.106	erad	0.27	member	0.169	spread viru	12.918
allergi	-2.332	face	0.012	mercuri	-0.414	stage	0.101
approv	-0.516	feel better	0.15	metro	-0.365	stai indoor	1.969
artifici	1.319	feel slightli	0.775	mile	-0.092	strength	0.739
assembl	0.231	fellow	0.319	miss	0.073	strong	0.018
asthmat	2.607	flu bad	4.953	need take	0.759	suspect swine	2.503
attempt	0.322	flu jab	-0.11	nurs	0.118	swine	-0.203
behavior	-1.349	flu relat	3.183	pain night	9.823	swine flu	1.577
bind	0.437	flu symptom	1.471	perform	0.083	symptom swine	1.626
blood	0.05	flu web	-5.463	personnel	-1.359	take care	0.21
boni	0.984	fluid	1.87	physic emotion	6.192	talk friend	-2.518
bulg	-0.733	fuss	0.234	pictur	-0.124	tast	0.08
case swine	4.282	germ	0.111	properti	-0.372	team	-0.044
caution	1.174	ground take	-3.022	radar	0.287	throat	0.251
cellular	-2.072	guilti	-0.394	reach	0.201	time knock	6.523
check code	4.495	habit	0.381	remain	-0.666	tissu	-0.012
check site	0.149	halt	0.819	rough	0.075	total cost	-4.794
checklist	-1.595	health	-0.04	run	0.143	transmit	1.535
chicken	0.286	health care	-0.393	rush	-0.07	troop	0.767
cleav	0.991	healthcar worker	1.339	scari	0.109	underli	-0.221
confirm swine	**21.874**	hippocr	-6.038	seal	-0.091	underli health	11.707
cough	0.234	holidai	-0.021	season	-0.064	unquot	8.753
cough fit	2.395	home wors	6.302	seizur	2.987	upcom	0.071
cough lung	2.406	huge	-0.199	self	0.059	viru epidem	-8.805
cough night	6.748	ion channel	4.974	sik	-0.542	visit doctor	-3.456
cruis	-1.186	irrig	8.721	site	0.06	wave	0.033
daughter	0.048	item	-0.219	sleep well	0.753	weight loss	-0.296
die swine	0.196	kick ass	-0.15	soak	0.41	wikipedia	0.66
dilut	2.708	knock	0.24	sore head	2.023	woke sweat	-19.912
drag	0.147	major	-0.376	speed	-0.198	wonder swine	7.266

Hybrids

Infodemiology
Example Prediction Analysis

Table X. Feature Class H – Hybrid selection of 1-grams and 2-grams for Flu case study (Round 1 of 5-fold cross validation) – All weights (**w**) should be multiplied by 10^4

n-gram	w	*n*-gram	w	*n*-gram	w	*n*-gram	w
acut	-0.796	effect swine	19.835	medic	0.48	spike	0.032
afford	-0.106	erad	0.27	member	0.169	spread viru	12.918
allergi	-2.332	face	0.012	mercuri	-0.414	stage	0.101
approv	-0.516	feel better	0.15	metro	-0.365	stai indoor	1.969
artifici	1.319	feel slightli	0.775	mile	-0.092	strength	0.739
assembl	0.231	fellow	0.319	miss	0.073	strong	0.018
asthmat	2.607	flu bad	4.953	need take	0.759	suspect swine	2.503
attempt	0.322	flu jab	-0.11	nurs	0.118	swine	-0.203
behavior	-1.349	flu relat	3.183	pain night	9.823	swine flu	1.577
bind	0.437	flu symptom	1.471	perform	0.083	symptom swine	1.626
blood	0.05	flu web	-5.463	personnel	-1.359	take care	0.21
boni	0.984	fluid	1.87	physic emotion	6.192	talk friend	-2.518
bulg	-0.733	fuss	0.234	pictur	-0.124	tast	0.08
case swine	4.282	germ	0.111	properti	-0.372	team	-0.044
caution	1.174	ground take	-3.022	radar	0.287	throat	0.251
cellular	-2.072	guilti	-0.394	reach	0.201	time knock	6.523
check code	4.495	habit	0.381	remain	-0.666	tissu	-0.012
check site	0.149	halt	0.819	rough	0.075	total cost	-4.794
checklist	-1.595	health	-0.04	run	0.143	transmit	1.535
chicken	0.286	health care	-0.393	rush	-0.07	troop	0.767
cleav	0.991	healthcar worker	1.339	scari	0.109	underli	-0.221
confirm swine	**21.874**	hippocr	-6.038	seal	-0.091	underli health	11.707
cough	0.234	holidai	-0.021	season	-0.064	unquot	8.753
cough fit	2.395	home wors	6.302	seizur	2.987	upcom	0.071
cough lung	2.406	huge	-0.199	self	0.059	viru epidem	-8.805
cough night	6.748	ion channel	4.974	sik	-0.542	visit doctor	-3.456
cruis	-1.186	irrig	8.721	site	0.06	wave	0.033
daughter	0.048	item	-0.219	sleep well	0.753	weight loss	-0.296
die swine	0.196	kick ass	-0.15	soak	0.41	wikipedia	0.66
dilut	2.708	knock	0.24	sore head	2.023	woke sweat	-19.912
drag	0.147	major	-0.376	speed	-0.198	wonder swine	7.266

Weekly ILI Reports by health region (time t) = b0 + b1 **Avg. Weekly Wght. Flu-Score by Region (time t)** + e

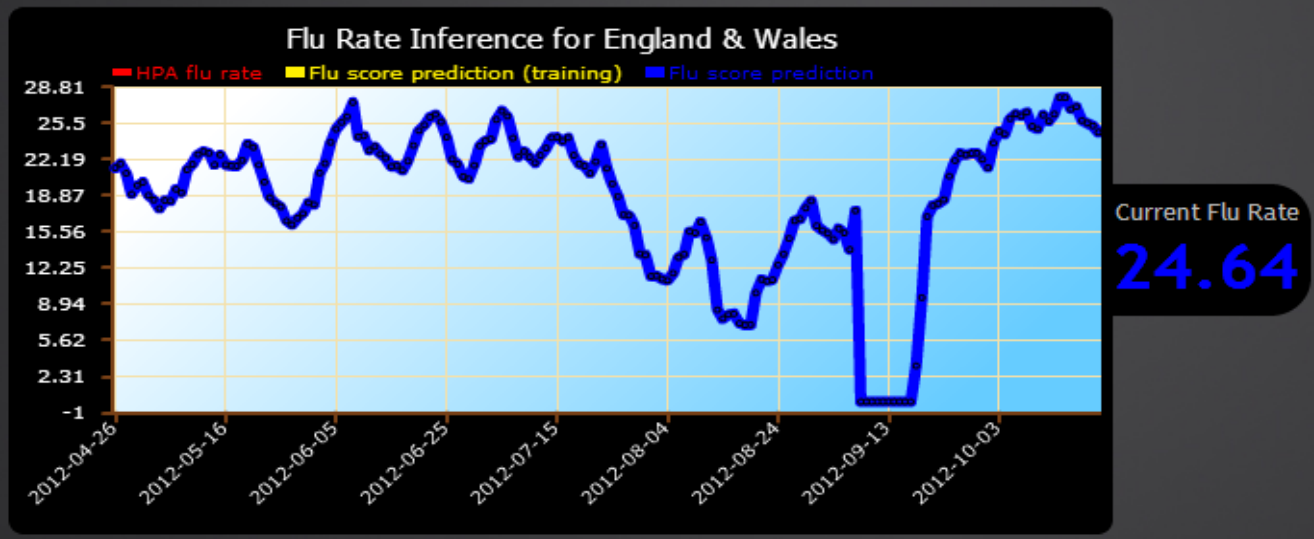
		Hybrid 1	Hybrid 2	Hybrid 3	…	Hybrid n		
Tweets	Week	wgt1	wgt2	wgt3	…	wgtn	Flu-Score	Indep Var
Tweet1	1	wgt1*N11	wgt2*N21	wgt3*N21	…	wgtn*N21	Sum1 W*N	
Tweet2	1	wgt1*N21	wgt2*N21	wgt3*N21	…	wgtn*N21	Sum2 W*N	
Tweet3	1	wgt1*N31	wgt2*N21	wgt3*N21	…	wgtn*N21	Sum3 W*N	Aver Wk-1
…	…	…	…	…	…	…	…	
Tweet m	24	wgt1*Nm	wgt2*N21	wgt3*N21	…	wgtn*N21	Summ W*N	Aver Wk-m

Infodemiology
Example Prediction Analysis

Flu Detector *tracking epidemics on Twitter*

Home | Regional Inferences | Archive | About

Here are the inferred flu rates for England & Wales in the last 6 months based on geolocated Twitter content



Current Flu Rate
24.64

Vertical axis — Inferred flu rate indicating the number of GP consultations per 100,000 citizens where the diagnosis' result was Influenza-like Illness (ILI)

Horizontal axis — Date in yyyy-mm-dd format

Check Regional Flu Inferences:
for the last 6 months
since June 2009

Flu Detector uses the content of Twitter to nowcast flu rates in several UK regions. Inferences are compared with official ILI rates from HPA. Some early performance evaluation results are available here. The methodology is described in the following publications:

Detecting Events and Patterns in Large-Scale User Generated Textual Streams with Statistical Learning Methods, V. Lampos, Ph.D. Thesis, 2012.

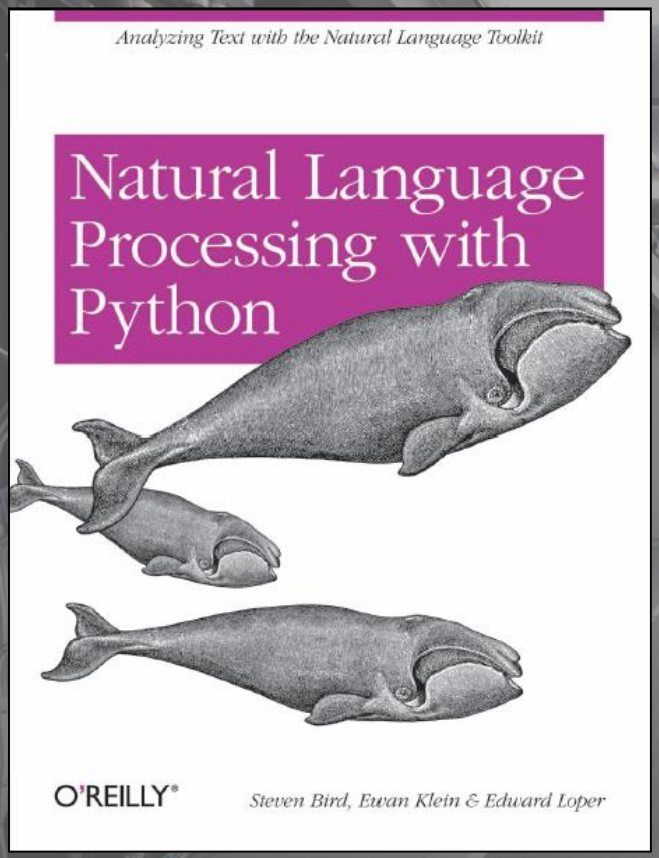
Nowcasting Events from the Social Web with Statistical Learning, V. Lampos and N. Cristianini, ACM TIST 2011.

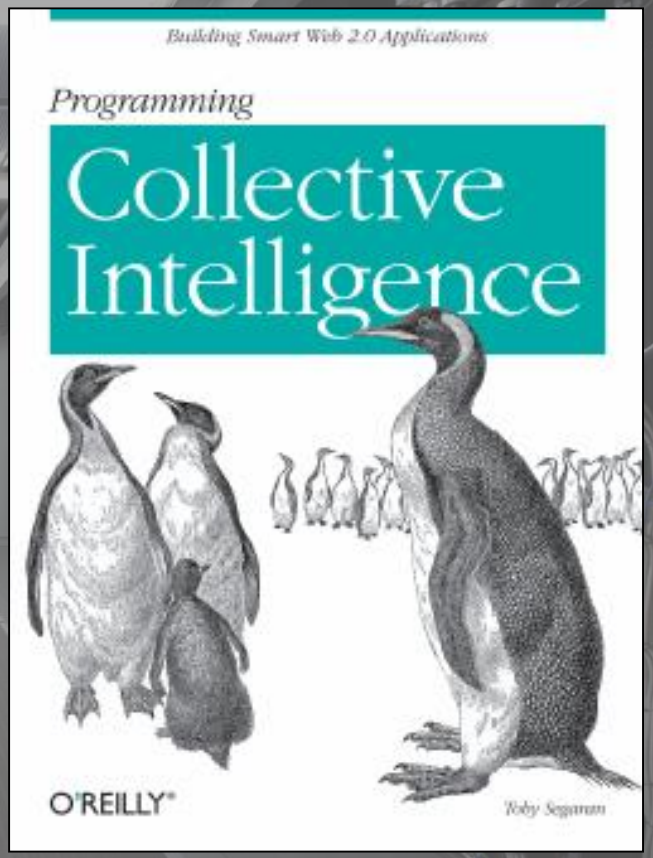
Tracking the flu pandemic by monitoring the Social Web, V. Lampos and N. Cristianini, CIP 2010.

Flu Detector - Tracking Epidemics on Twitter, V. Lampos, T. De Bie and N. Cristianini, ECML PKDD 2010.

http://geopatterns.enm.bris.ac.uk/epidemics/

Programming Text Mining for Prediction with Python





Text Mining for Prediction: Programming with Python & NLTK


```
# Utilizes “nltk”,  a Python “natural language toolkit”
# Step 1: Initialize modules, stopwords and stemmer

import simplejson
import urllib
import re
import nltk

from nltk.corpus import stopwords
stopwords = stopwords.words('english')
porter = nltk.PorterStemmer()

def remove_stopwords(text):
   stopwords = nltk.corpus.stopwords.words('english')
   content = [w for w in text if w.lower() not in stopwords]
   return content
```

Text Mining for Prediction: Programming with Python & NLTK


```
# Step 2: Utilize Twitter Search API to collect “flu” tweets  in JSON format.
# Then extract the “text” fields associated with each tweet.

itemsFlu = []

for i in range(0,14):
   urlFlu = 'http://search.twitter.com/search.json?q=flu&rpp=100&page=1'
   resultFlu = simplejson.load(urllib.urlopen(urlFlu))
   itemsFlu = itemsFlu + resultFlu['results']

tweetsFlu = [ item['text'] for item in itemsFlu]

# Step 3.  Eliminate Retweets

FluTweetsNoRTs= []

for tweetText in tweetsFlu:
   if not re.search('RT ',tweetText):
      tweetTextList = [tweetText]
      FluTweetsNoRTs = FluTweetsNoRTs + tweetTextList
```

Text Mining for Prediction: Programming with Python & NLTK


```
# Step 3:  Preparing to do Text analysis
noTweets = 0; Flu_tmp_word = []; Flu_tot_word = []
Flu_tmp_low = []; Flu_tot_low = []; Flu_tmp_alpha = []; Flu_tot_alpha = []
Flu_tmp_stop = []; Flu_tot_stop = []; Flu_tmp_stem = []; Flu_tot_stem = []
stems_dict = {}  # dictionary holding "text" broken into words for 1..N tweets

# Step 4: Text processing.  Produces lowercase, alpha stems devoid of stopwords
for tweet in FluTw eetsNoRTs :
   noTweets = noTweets + 1
   Flu_tmp_word = [ w for w in tweet.split()];     Flu_tot_word = Flu_tot_word + Flu_tmp_word
   Flu_tmp_low = [w.lower() for w in Flu_tmp_word] ;     Flu_tot_low = Flu_tot_low + Flu_tmp_low
   Flu_tmp_alpha = [cv for w in Flu_tmp_low for cv in re.findall(r'^[a-z]+[a-z]+$', w)]
   Flu_tot_alpha = Flu_tot_alpha + Flu_tmp_alpha
   Flu_tmp_stop = remove_stopwords(Flu_tmp_alpha);     Flu_tot_stop = Flu_tot_stop + Flu_tmp_stop
   Flu_tmp_stem = [porter.stem(t) for t in Flu_tmp_stop];     Flu_tot_stem = Flu_tot_stem + Flu_tmp_stem
   Flu_stems_dict[noTweets] = Flu_tmp_stem
```

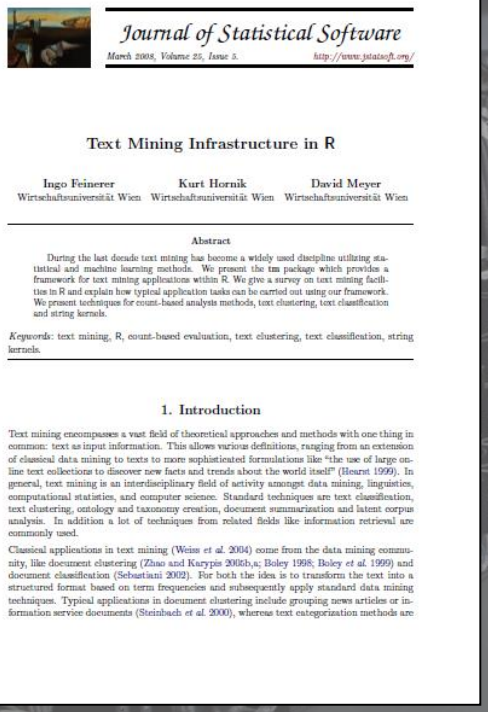
Text Mining for Prediction: Programming with Python & NLTK


```
# Step 5: Analyzing frequency distributions

fdist_Flu_stems = nltk.FreqDist(Flu_tot_stem)
fdist_Flu_stems.plot(25)
fdist_Flu_stems.items()[0:25]

# Step 6. Produce doc-matrix "wc" (a dictionary hold the counts associated
#  with each word).

for dCnt in range(1,len(Flu_stems_dict)):
   wlist = Flu_stems_dict[dCnt]
   for word in wlist:
      wc.setdefault(word,0)
      wc[word] += 1
```

Text Mining for Prediction: Programming with Python & NLTK

```
# Step 7. Eliminate all words that occur less than 3 times
apcount = {}
for word,wcnt in wc.items():
   apcount.setdefault(word,0)
   if wcnt > 3:  apcount[word] = wcnt

# Step 8. Write out doc-term matrix to a file
out = file('flu-doc-matrix.txt','w'); out.write('Tweets')
for word,count in apcount.items():
   out.write(",");     out.write(word) ;  out.write('\n')
for tweetno, twlist in Flu_stems_dict.items():
   tweetname = "Tweet" + str(tweetno);    out.write(tweetname)
   for word, count in apcount.items():
      if word in twlist:   out.write(",");   out.write("1")
      else: out.write(",");   out.write("0");  out.write("\n")
out.close()
```

Text Mining for Prediction: Programming with R, tm and RTextTools



Journal of Statistical Software
March 2008, Volume 25, Issue 5. http://www.jstatsoft.org/

Text Mining Infrastructure in R

Ingo Feinerer, Wirtschaftsuniversität Wien
Kurt Hornik, Wirtschaftsuniversität Wien
David Meyer, Wirtschaftsuniversität Wien

Abstract

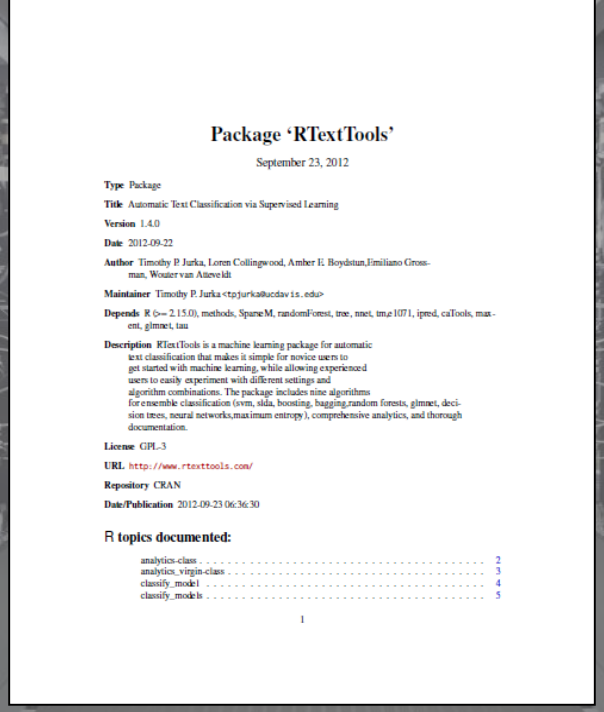
During the last decade text mining has become a widely used discipline utilizing statistical and machine learning methods. We present the tm package which provides a framework for text mining applications within R. We give a survey on text mining facilities in R and explain how typical application tasks can be carried out using our framework. We present techniques for count-based analysis methods, text clustering, text classification and string kernels.

Keywords: text mining, R, count-based evaluation, text clustering, text classification, string kernels.

1. Introduction

Text mining encompasses a vast field of theoretical approaches and methods with one thing in common: text as input information. This allows various definitions, ranging from an extension of classical data mining to texts to more sophisticated formulations like "the use of large online text collections to discover new facts and trends about the world itself" (Hearst 1999). In general, text mining is an interdisciplinary field of activity amongst data mining, linguistics, computational statistics, and computer science. Standard techniques are text classification, text clustering, ontology and taxonomy creation, document summarization and latent corpus analysis. In addition a lot of techniques from related fields like information retrieval are commonly used.

Classical applications in text mining (Weiss *et al.* 2004) come from the data mining community, like document clustering (Zhao and Karypis 2005b,a; Boley 1998; Boley *et al.* 1999) and document classification (Sebastiani 2002). For both the idea is to transform the text into a structured format based on term frequencies and subsequently apply standard data mining techniques. Typical applications in document clustering include grouping news articles or information service documents (Steinbach *et al.* 2000), whereas text categorization methods are



Package 'RTextTools'

September 23, 2012

Type Package

Title Automatic Text Classification via Supervised Learning

Version 1.4.0

Date 2012-09-22

Author Timothy P. Jurka, Loren Collingwood, Amber E. Boydstun,Emiliano Grossman, Wouter van Atteveldt

Maintainer Timothy P. Jurka <tpjurka@ucdavis.edu>

Depends R (>= 2.15.0), methods, SparseM, randomForest, tree, nnet, tm,e1071, ipred, caTools, maxent, glmnet, tau

Description RTextTools is a machine learning package for automatic text classification that makes it simple for novice users to get started with machine learning, while allowing experienced users to easily experiment with different settings and algorithm combinations. The package includes nine algorithms for ensemble classification (svm, slda, boosting, bagging,random forests, glmnet, decision trees, neural networks,maximum entropy), comprehensive analytics, and thorough documentation.

License GPL-3

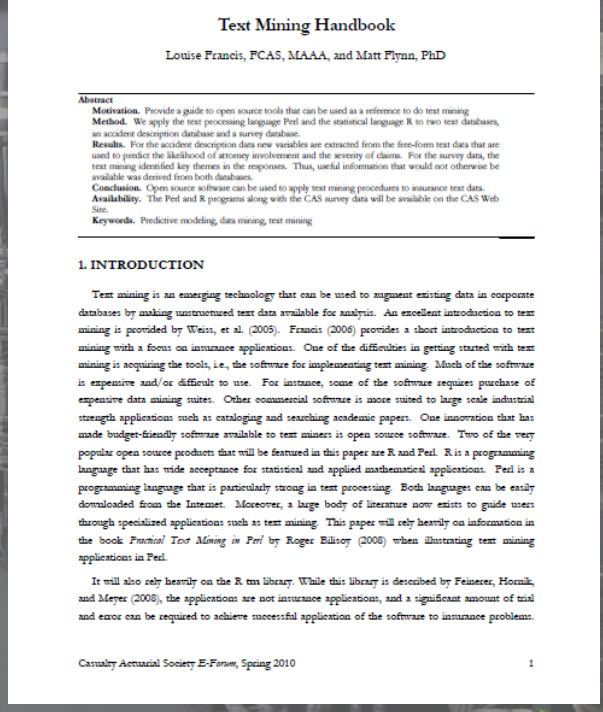
URL http://www.rtexttools.com/

Repository CRAN

Date/Publication 2012-09-23 06:36:30

R topics documented:

1



Text Mining Handbook

Louise Francis, FCAS, MAAA, and Matt Flynn, PhD

Abstract

Motivation. Provide a guide to open source tools that can be used as a reference to do text mining

Method. We apply the text processing language Perl and the statistical language R to two text databases, an accident description database and a survey database.

Results. For the accident description data new variables are extracted from the free-form text data that are used to predict the likelihood of attorney involvement and the severity of claims. For the survey data, the text mining identified key themes in the responses. Thus, useful information that would not otherwise be available was derived from both databases.

Conclusion. Open source software can be used to apply text mining procedures to insurance text data.

Availability. The Perl and R programs along with the CAS survey data will be available on the CAS Web Site.

Keywords. Predictive modeling, data mining, text mining

1. INTRODUCTION

Text mining is an emerging technology that can be used to augment existing data in corporate databases by making unstructured text data available for analysis. An excellent introduction to text mining is provided by Weiss, et al. (2005). Francis (2006) provides a short introduction to text mining with a focus on insurance applications. One of the difficulties in getting started with text mining is acquiring the tools, i.e., the software for implementing text mining. Much of the software is expensive and/or difficult to use. For instance, some of the software requires purchase of expensive data mining suites. Other commercial software is more suited to large scale industrial strength applications such as cataloging and searching academic papers. One innovation that has made budget-friendly software available to text miners is open source software. Two of the very popular open source products that will be featured in this paper are R and Perl. R is a programming language that has wide acceptance for statistical and applied mathematical applications. Perl is a programming language that is particularly strong in text processing. Both languages can be easily downloaded from the Internet. Moreover, a large body of literature now exists to guide users through specialized applications such as text mining. This paper will rely heavily on information in the book *Practical Text Mining in Perl* by Roger Bilisoly (2008) when illustrating text mining applications in Perl.

It will also rely heavily on the R tm library. While this library is described by Feinerer, Hornik, and Meyer (2008), the applications are not insurance applications, and a significant amount of trial and error can be required to achieve successful application of the software to insurance problems.

Text Mining for Prediction: Programming with R, tm and RTextTools

```
# initialize libraries
library(twitteR)
library(tm)
library(RTextTools)
library(plyr)
library(stringr)

# utilize Twitter Search API to collect "flu" Tweets
# in JSON format
fluTweets = searchTwitter('flu', n=1500, lang='en')

# extract text from JSON objects
fluTweetText = laply(fluTweets, function(t) t$getText())

# remove retweets
indxFluRetweets <- grep('RT',fluTweetText)
indxFluTweetText <- c(1:length(fluTweetText))
indxFluTweetNonRTs <- !(indxFluTweetText %in%
indxFluRetweets)
fluTweetTextNonRTs <-
fluTweetText[indxFluTweetNonRTs]
```

```
# text preprocessing function
toLowerCaseAlpha <- function(x){
removeHTTP <- gsub("http:[/a-zA-Z0-9._]+","",x)
removeNonAlpha <- gsub("[^a-zA-Z]"," ",removeHTTP)
removeMultipleSpaces <- gsub(" +"," ",removeNonAlpha)
changeToLowerCase <- tolower(removeMultipleSpaces)
return(changeToLowerCase) }

# clean tweet text, convert lower case, remove stop words
# create corpus for analysis
fluTweetTextCleaned <- toLowerCaseAlpha(fluTweetTextNonRTs)
corpusFluTweets <- Corpus(VectorSource(fluTweetTextCleaned))
corpusFluText <- tm_map(corpusFluTweets, removeWords,
corpusStopwords)
```

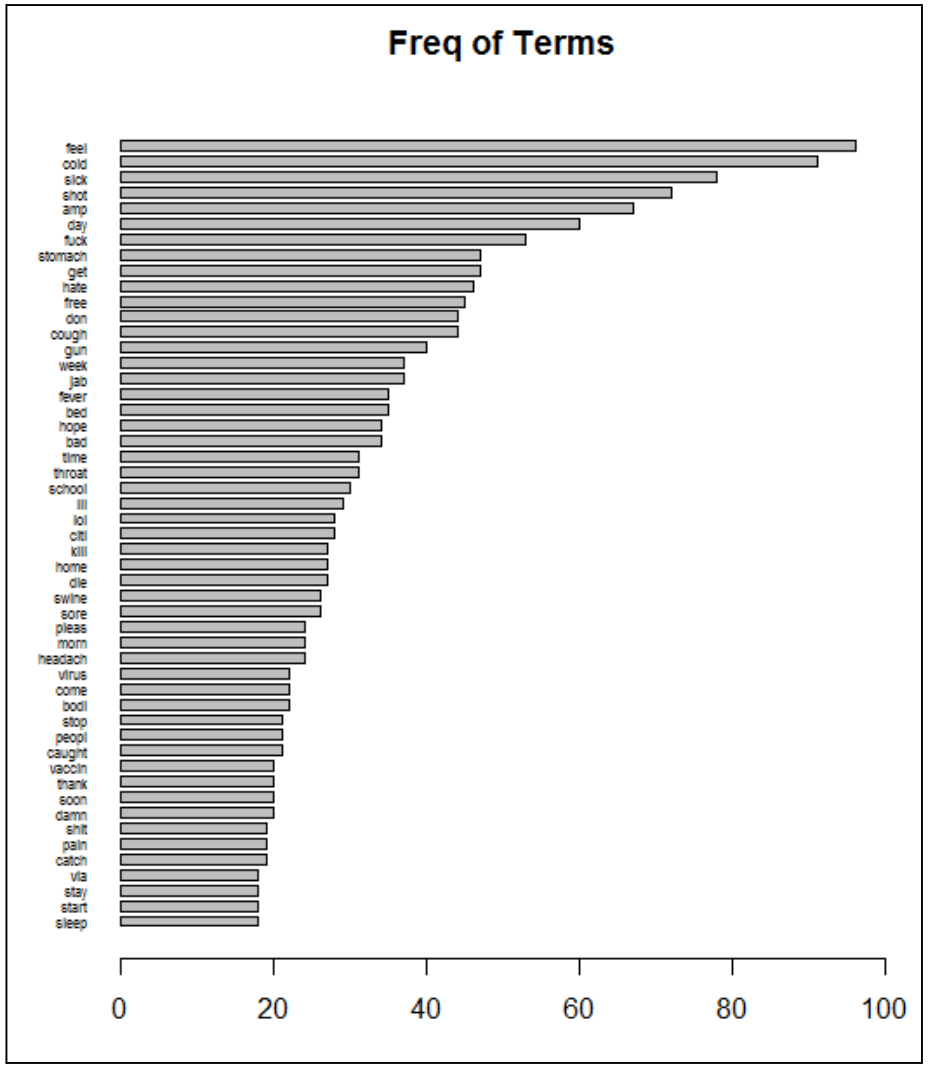
Text Mining for Prediction: Programming with R, tm and RTextTools

```
# function to convert tokens/words to stems
convertToStems <- function(x){
tokens <- strsplit(x,' +')
listToVect <- unlist(tokens)
stems <- wordStem(listToVect)
return(stems)}

# convert to stems and create new corpus
FluStems <- sapply(corpusFluText, convertToStems)
corpusFluStems <- Corpus(VectorSource(FluStems))
```

```
# create document –term matrix
corpusFluStemsDTM <-
  DocumentTermMatrix(corpusFluStems)

# create vector of individual stems and the associated
# frequencies with which they occur across all tweets
fluStemsDTMMat <- as.matrix(corpusFluStemsDTM)
fluStemsDTMMatSorted <-
  sort(colSums(fluStemsDTMMat))
```

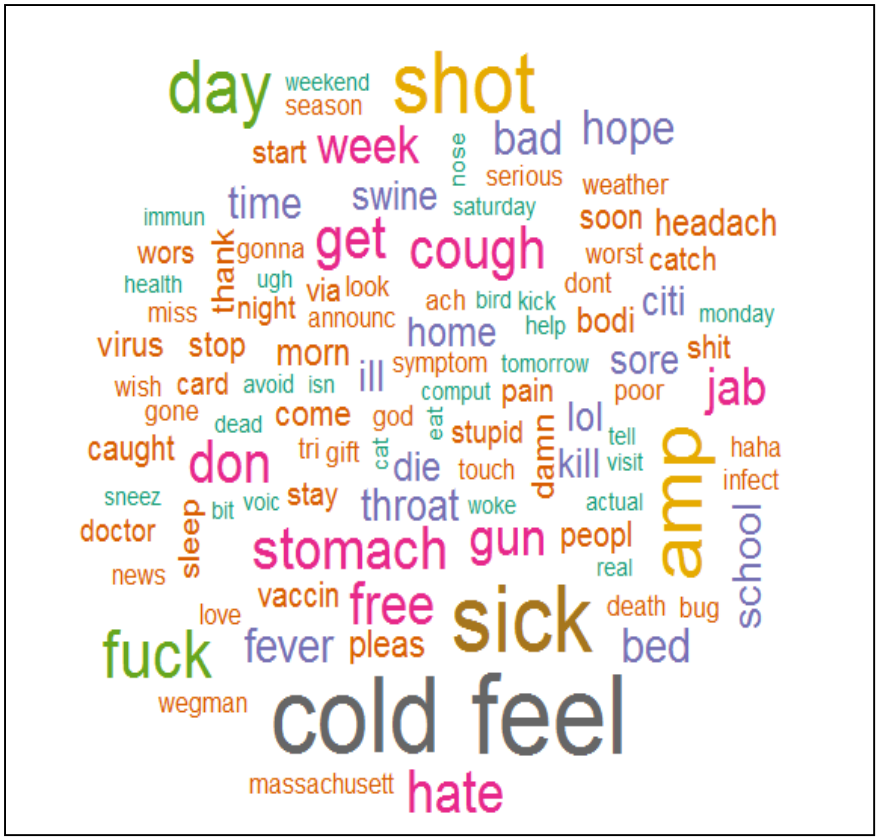
Text Mining for Prediction: Programming with R, tm and RTextTools

```
# plot frequency disttribution of  stems

noOfPlotPnts = 50

numOfStems =
   length(fluStemsDTMMatSorted)

plotStart = numOfStems - noOfPlotPnts +1

plotEnd = numOfStems

top50FluStems =
   fluStemsDTMMatSorted[plotStart:plotStart]

barplot(top50,horiz=TRUE,cex.names=0.5,
    space = .5,las=1, xlim=c(0,100),
    main="Freq of Terms")
```



Text Mining for Prediction: Programming with R, tm and RTextTools

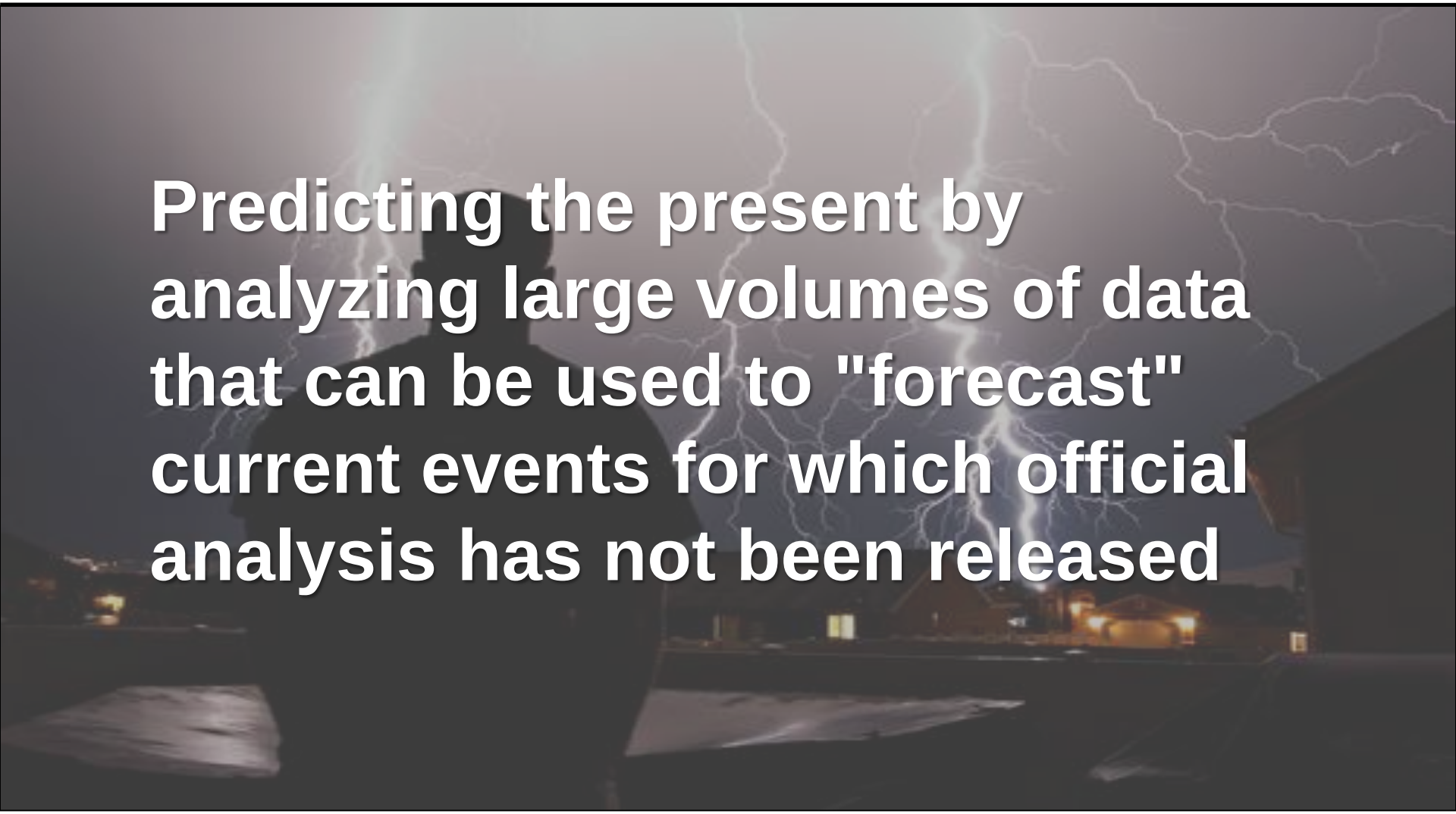
```
# create wordcloud of  stems based on frequency

library(wordcloud)
library(RColorBrewer)

fluStemNames <- names(fluStemsDTMMatSorted)

dfFluStemDTMMat <- data.frame(word=fluStemNames,
   freq=fluStemsDTMMatSorted)

pal2 <- brewer.pal(8,"Dark2")

wordcloud(dfFluStemDTMMat$word,
   dfFluStemDTMMat$freq, min.freq=10, colors=pal2)
```



Infodemiology is a form of NOWcasting

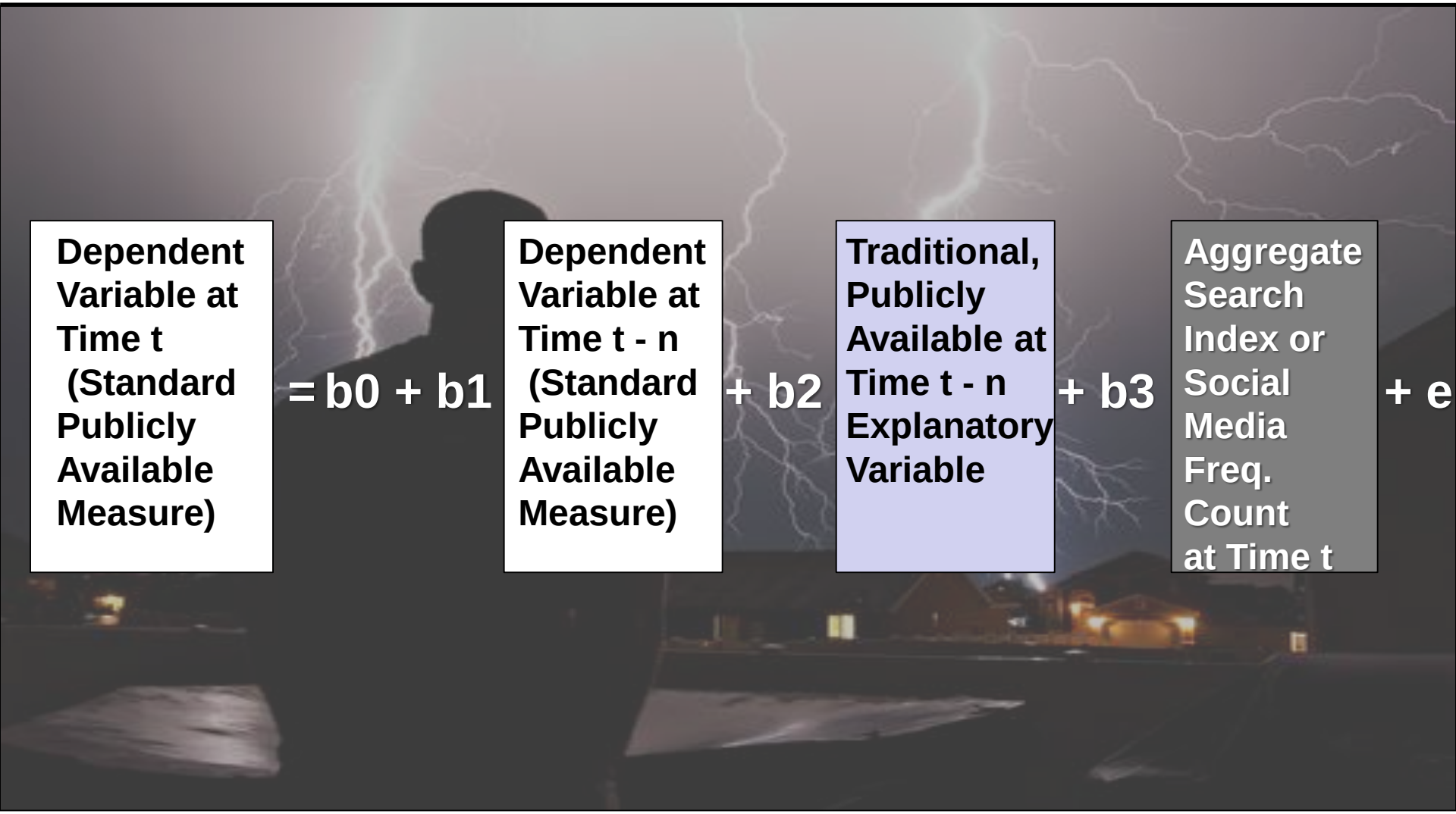
Nowcasting

Now+Forecasting



Nowcasting

Basic Regression Analysis



Nowcasting

Examples

Authors	Title	Date (M-Y)	Type	Data Source	Dependent Variables	Explanatory Variables	Model	Results
Kholodilin et al.	Do Google Searches Help in Nowcasting Private Consumption? A Real-Time Evidence for the US	Apr-10	Search	Fed reserve data on US private consumption and related Google search terms from Jan '05-Dec '09.	Year-on-Year Growth Rate of Monthly US Real Private Consumption, ALFRED db of Fed Rsrv of St. Louis	220 Google Trend/Insights Search terms related to Priv Consumption reduced to 10 principal components for montly periods from Jan 2005 to Dec 2009	Y-o-Y monthly URPC growth rates for 3 sets of regressors -- Sentiment (consumer sentiment and confidence); Financial (short term and long term interest rates and S&P 500); Query (combinations of principal components of query terms)	Query term principal components outperform standard Sentiment and Financial Indicators. A combination of two of the factors work best -- those related to mobility and health care consumption.
Sakaki et al.	Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors	Apr-10	Tweets	Earthquake occurrences/intensity and tweets containing the words "earthquake" or "shaking" in Japan from Aug '09-Sep '09	Occurrence, intensity and location of an earthquate in Japan	Tweets that contain the query words "earthquake" and/or "shaking"by location	Utilized a support vector machine (SVM) to determine whether tweet reports refers to an earthquake occurrence. The reports are then matched against actual occurrence at a particular location to see if it is detected within 1 min of occurrence.	Of those earthquakes occurring in the Aug-Sept time frame in Japan, 96% of 24 quakes above an intensity of 3 were reported in a tweet. Of the 24 quakes, 80% were reported with a minute of occurrence. This is much faster the reports issued by the Japan Meteorological Agency.
Carrière-Swallow & Labbé	Nowcasting With Google Trends in an Emerging Market	Jul-10	Search	Auto sales and Google search indices for specific automobiles in Chile from 2005 thru 2010.	% year-on-year change in auto sales in Chile	Google Search index of interest in automobile purchases in Chile	% change in y-on-y autosales in Chile regressed against a compositive auto Google Search index based on queries about 9 leading automobile manufactures in Chile.	Despite relatively low rates of Internet usage in Chile, models incorporating Google Trends Automotive Index outperform benchmark specifications in both in-sample and out-of- sample nowcasts of y-on-y % changes in autosales while providing substantial gains in information delivery times
Ciulla	Beating the news using social media: the case study of American Idol	May-12	Tweets	Tweets from US users that related to American Idol contestants and tweeted during the voting time window for each episode during 11th season from Jan '12 - May '12	Contestants who were eliminated or won final episode	Number of hashtags with Tweets with hastags signifying contestants	Contestants with the fewest mentions predicted to be the candidate eliminated.	In general the simple tweet frequencies are strong predictor of the contestant who will be eliminated.
Chadwick & SengulCiu	Nowcasting Unemployment Rate in Turkey: Let's Ask Google	Jun-12	Search	Monthly Turkey non-agricultural unemployment rate from Jan '05-Dec '11	Monthly Turkey non-agricultural unemployment rate	Google Search Index in Turkey for terms directly ("looking for job") or indirectly (job announcements) related to unemployment.	Linear auto regression models and Bayesian Model Averaging procedure to investigate whether Google search query data can improve	Models with Google Search Indicators perform better in nowcasting the 1 period, 2 periods and 3 periods ahead unemployment rate than the benchmark where we use only the lag values of the unemployment rate.
Song, Pan, Ng	Forecasting hotel room demand using search engine data	Sep-12	Search	Weekly data on hotel bookings in Charleston, SC and Google trend data for specific travel tourism search terms from Jan '08-Aug '09	Weekly Hotel Bookings in Charleston, SC	Indexed Search Volumes from Google Trends/Insights Jan 2008-Aug 2009	Log of Room Nights for Log of Search Volumes - Charleston, Travel Charleston, Charleston Hotels, Charleston Restaurants, Charleston Tourism	Test various statistical models; all gave reasonable forecasts. Best fit model was Autoregressive Distributed Lag (ADLM) with a lag period of 6 weeks.
McLaren, Shanbhogue	Using internet search data as economic indicators	Q2-11	Search	UK monthly unemployment data and housing price growth from June '04-Jan '11 associated with Google Trend query indices for job seeking and unemployment	Official monthly unemployment data and housing price growth in the UK from June 2004-Jan 2011	Google Trend/Insight query indexes for the term "Job Seekers Allowance (JSA)" for unemployment and "Estate Agents" for housing	For unemployment, linear AR model with query term, claimant count, and GfK consumer confid. as exp vars; for housing price growth with query term, Home Builders and Royal Instit. of Chartered Surveyors price growth balances as exp vars.	For unemployment forecasts, claimant count strongest followed by query term. For housing prices, the query term was much stronger than HBF and RICS data.

Nowcasting

Examples

Authors	Title	Date (M-y)	Type	Data Source	Dependent Variables	Explanatory Variables	Model	Results
Gruhl et al.	The Predictive Power of Online Chatter.	Aug-05	Blogs	Amazon Sales Rank for for best selling books from Jul '04-Aug '04 and number of mentions in blogs for same time period	Amazon Sales Rank for 2340 bestselling books in 4 month period (Jul 2004-Aug 2004) and spikes in these sales ranks	Number of mentions of the book/author in over 300K blogs whose postings that were maintained by IBM's WebFountain project (over 200K postings/day)	Cross correlation of time series for sales rank and mentions.	While sales rank is a poor predictor of the change in sales rankings, a prior spike in mentions predicts quite well a future spike in sales rank.
Choi, Varian	Predicting the Present with Google Trends	Apr-09	Search	Monthly data for search from Google Trends associated with various retail sales and travel data from Jan '04-Aug '08	US Census Bureau Advance Monthl Retail Sales (general and specific) and Travel (Visitor arrival in Hong Kong)	Google Trend/Insight query indices for categories and subcategories related to retail sales (general and specifix) and related to Travel	Google Trend indices for query subcategories related to (log values) of overall monthly retail trade (NAICS categories), automotive sales, home sales and travel.	Simple seasonal AR models and fixed-effects models that includes relevant Google Trend variables tend to outperform models that exclude these variables. In some cases small gains, in other substantial.
Suhoy	Query Indices and a 2008 Downturn: Israeli Data	Jul-09	Search	Monthly official economic growth data for the months and quarters from 2nd quarter 2004 to 2nd quarter 2009 along with various Google search indices during the same period	Monthly percent changes of various real values of industrial production, retail trade, revenue of trade and services, consumer imports, exports of services and the employment rate	30 Google Search Index categories related to consumption and employment	Bayesian probabilities of downturn calculated by Hamilton's two-sate Markov switching AR(0) model. Used to determine changes in query indices can predict changes in official economic variables.	Six leading query categories including HR (recruiting and staffing), home appliances, travel, real estate, food and drink and beauty and contain cyclical components which conform with cycles of economic growth. The strongest relationship was between HR and unemployment.
Sadikov et al.	Blogs as Predictors of Movie Success	Aug-09	Blogs	Weekly box movie sales, gross sales, and critic and user rankings from Nov '07-Nov '08 and counts for moving references and sentiment measures for same time period.	Movie critic ranking, user ranking, 2008 gross sales, weekly box office sales (weeks 1-5)	Analysis of spinn3r.com blog data set 11/07-11/08, counting movie references and sentiment within specified time window before and after movie release date.	Linear regression for weekly rankings and sales data by blog references and sentiment.	Minimal correlation between rankings and references and sentiment. Strong correlation between references and gross sales but week with sentiment. Strongest relationships with timing of references in weeks after release.
Zhang & Skiena	Improving Movie Gross Prediction Through News Analysis	Sep-09	Blogs & News	Online News stores & blogs 1960-2008 along with movie receipt and IMDB data. News stores analyzed for various pre-release time periods	Gross receipts from movies	Variety of IMDB variables (e.g. movie genre), movie budget, number of first week theaters, and number of stores and mentions of movie titles, director, top 3 & top 15 actors	Linear regression of receipts for various combinations of explanatory variables. Also, K-NN nearest neighbor analysis determining factors associated with gross receipts	Number of news articles mentioning moving 1 week prior have highest correlation (~.7)and predictive ability
Wu & Brynjolfsson	The Future of Prediction: How Google Searches Foreshadow Housing Prices and Sales	Dec-09	Search	Quarterly Housing Sales and Housing Price Index for 50 US states along with the Google Search Index for Real Estate, Real Estate Agencies, and Home Appliances from 4th Quarter 2007 to 2nd Quarter 2007	Housing Sales and Housing Price Index (HPI)	Google Search Index for Real Estate, Real Estate Agencies, and Home Appliances	Linear autoregression between Housing Sales and prior sales, the HPI, and Search Indices for Real Estate and Real Estate Agencies as well as the same regression for the HPI	Strong predictive relationships between Housing Sales and searches for Real Estate Agencies. Similarly relationships for HPI.
Asur, Huberman	Predicting the Future with Social Media	Mar-10	Tweets	3 million tweets mentioning 24 movies from Nov '09-Feb '10 along with associated box-office revenues	Box-office revenues for (24) movies	Promotion tweets-retweets for a particular movie, tweet rates for particular movie per hour, ratio of positive to negative sentiments for the movie	Regression of 1st weekend box office revenues by promotional tweets-retweets, by tweet rates vs. Hollywood Stock Exchange prices, and 2nd weekend revenues by tweet rates and the sentiment ratio.	Promotional tweets are weakly correlated 1st weekend revs. Tweet rates are very strongly correlated (min .9) and a stronger predictor than HSX. Finally, tweet rates are strongly correlated with 2nd weekend revenues and sentiments improve the forecasts slightly.