Using Sentiment

Lecture 10: Text Analytics for Big Data

Mark Keane, Insight/CSI, UCD

Selling Things

stock-markets

social media

science

news

polls

***

sentiment-id

sentiment-use

time-series

summaries

***

VSMs

Classifiers

Clustering

cosine

jaccard

dice

levenschtein

TF-IDF

LLR

PMI

Entropy

simple frequencies

pre-processed text items of some sort…

Basically It's…

REM

- Words express various sentiments, feelings, opinions, biases, reactions…

- Tracking sentiment words tells use what people feel about these things, their opinions…

- Opinion mining, sentiment analysis, subjectivity analysis, appraisal extraction…

Overview

REM

- Identifying Sentiment Terms (Lect-9)
- Using Sentiments to Do Things (Lect-10)
- Key Examples (both)
- Some Implementations (both)

Sentiment Use: The Problem

- Identifying sentiments is hardish; we have seen use of human ratings, affective lexicons, supervised and unsupervised classifiers…
- But, remember, identifying the sentiment terms is only the first step
- You must collate these individual identifications into an overall sentiment assessment and then use that to solve some problem (e.g. prediction)

Sentiment Use: We've Seen

- Turney (2002) averaging the PMI-IR scores for bi-grams in a review to say overall this doc recommends the movie or not
- Pang et al (2000) classifying a review as pos/neg to tell you that you may like film
- But, there are many more…

REM

Sentiment Use: Business

- Has become big business; analysing product reviews for market research, recommending based on user reviews (Amazon)
- Standard part of online market research
- Many companies providing brand tracking, watch lists for stocks, news tracking, and other applications…often using dodgy stuff

Using Sentiment

- Sentiment used in many different ways:
  - to reflect opinion of a population
  - to predict people's behaviour
  - to make recommendations to others

Using Sentiment:

Reflecting Opinion

Reflecting Opinion

- Traditional market surveys elicited population opinions using questionnaires

- Big data provides modern equivalent using available data on web/internet/info-footpath

- Major step change in availability of personal information has changed everything… traditional market research is dying

Problem

- Take some population textual sources
- Extract the relevant part of those text sources and pre-process that in some way
- Identify sentiments (see previous lecture)
- Aggregate those sentiments into some summary of the population's opinion or track changes over time

Pulse: Mining Customer Opinions from Free Text

Michael Gamon, Anthony Aue, Simon Corston-Oliver, and Eric Ringger

Natural Language Processing, Microsoft Research, Redmond, WA 98052, USA,
(mgamon|anthaue|simonco|ringger)@microsoft.com,
http://research.microsoft.com/nlp/

3 Examples

From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series

Brendan O'Connor[†] **Ramnath Balasubramanyan**[†] **Bryan R. Routledge**[§] **Noah A. Smith**[†]
brenocon@cs.cmu.edu rbalasub@cs.cmu.edu routledge@cmu.edu nasmith@cs.cmu.edu

[†]School of Computer Science
Carnegie Mellon University

[§]Tepper School of Business
Carnegie Mellon University

Mining the Web for the "Voice of the Herd" to Track Stock Market Bubbles

Aaron Gerow
Trinity College Dublin
Dublin, Ireland
gerowa@tcd.ie

Mark T. Keane
University College Dublin
Belfield, Ireland
mark.keane@ucd.ie

Eg1

Pulse: Mining Customer Opinions from Free Text

Michael Gamon, Anthony Aue, Simon Corston-Oliver, and Eric Ringger

Natural Language Processing, Microsoft Research, Redmond, WA 98052, USA,
(mgamon|anthaue|simonco|ringger)@microsoft.com,
http://research.microsoft.com/nlp/

- Problem
- Setup & Data
- Architecture & Techniques
- Results & Findings
- Lessons

Eg1: Problem - Pulse

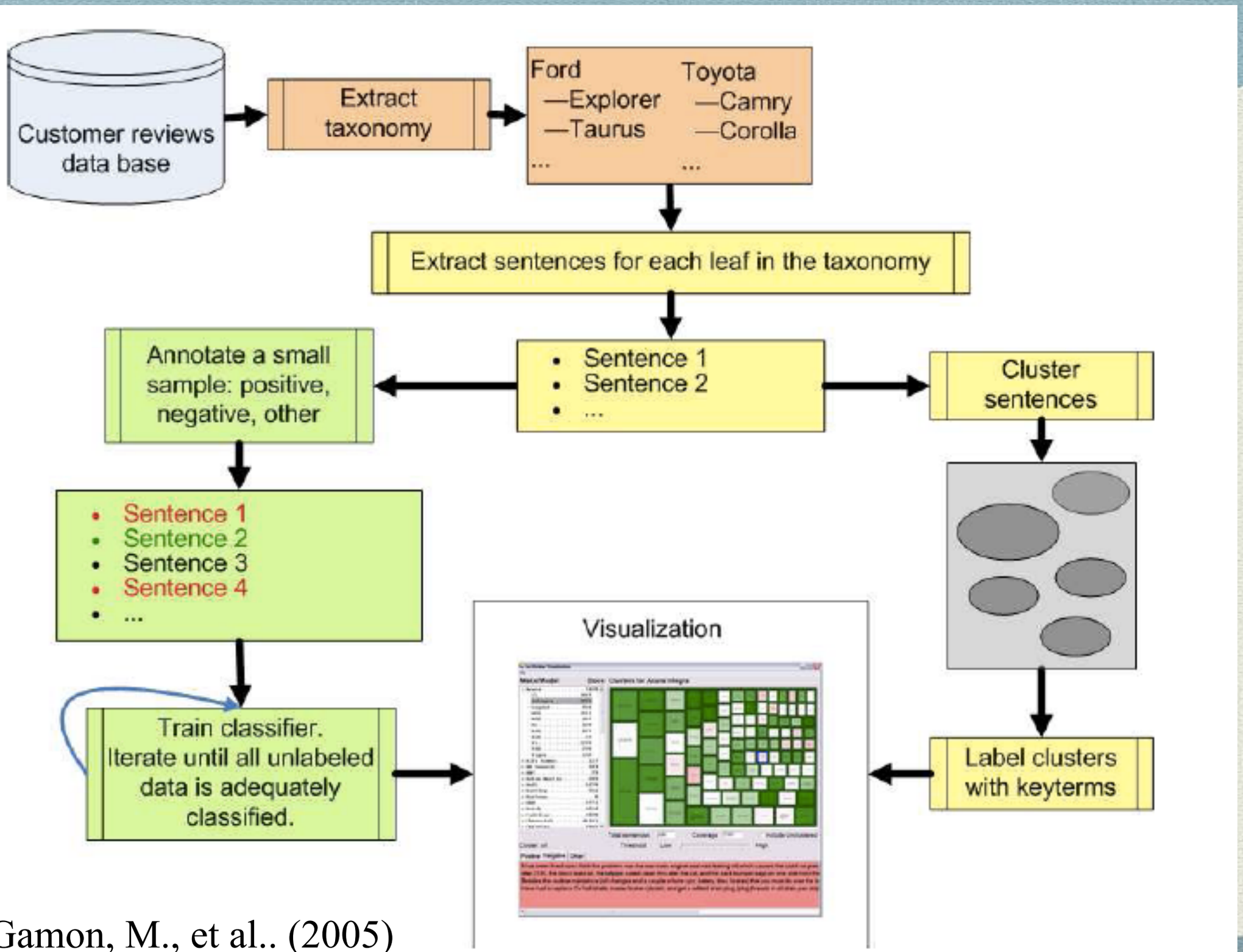
- Business intelligence on products, using customer comments and online reviews
  - identify sentiment
  - aggregate sentiment
  - visualise that aggregation
- Microsoft NLP Group

Gamon, M., Aue, A., Corston-Oliver, S., & Ringger, E. (2005). Pulse. In Advances in Intelligent Data Analysis VI (pp. 121-132). Springer Berlin Heidelberg.

Pulse: Set-up & Data

- 406,818 customer car reviews over 4 years, from website that also had ratings (1-10) of cars, average 8.3
- 1-50 sentences from each; 900,000 sentences in total
- Generally, positive reviews but they include negative sentences; better at sentence than document level
- Randomly select 3000 sentences, hand-rated as pos/neg/other; 2500 used for training, 500 for gold-standard testing (boost with unlabelled, unclear)

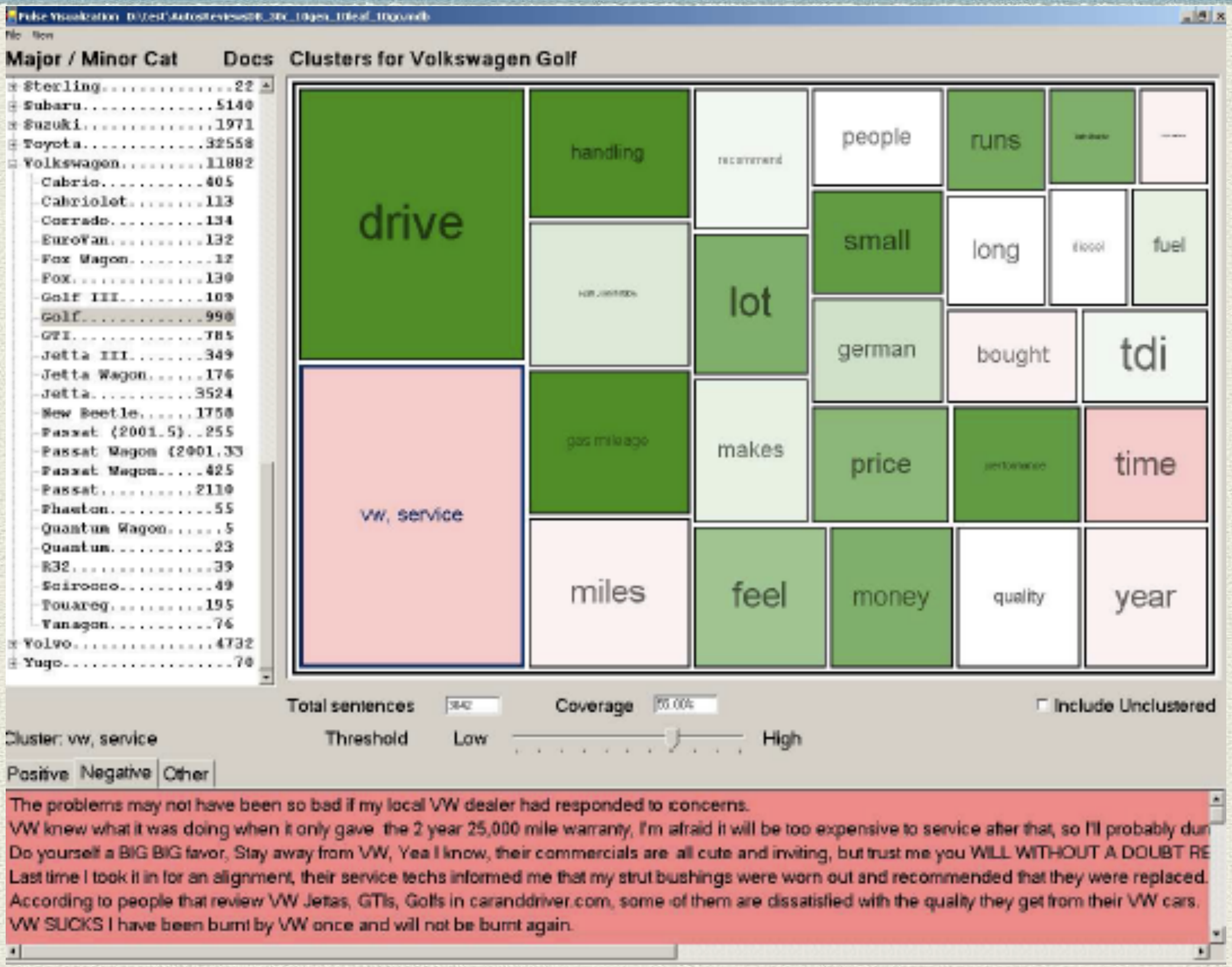
Pulse: Architecture



Gamon, M., et al.. (2005)

Pulse: Techniques

- Car name extracted, then sentences about that Car
- Car-sentences clustered into main topics/aspect-being-discussed using tf-idf clustering
- Classifier tags sentences as pos/neg/other
  - overall-box = whole car
  - car-topic box size = no of sentences
  - car-topic box colour = average pos/neg/neutral



Gamon, M., Aue, A., Corston-Oliver, S., & Ringger, E. (2005). Pulse: Mining customer opinions from free text. In Advances in Intelligent Data Analysis VI (pp. 121-132). Springer Berlin Heidelberg.

Techniques: Details

- Pre-processing for clustering: Special stop word exclusion (usual and sentiment-terms) means clusters orthogonal to sentiment
- Clustering: tried 4 but unsatisfactory; used Naive Bayes with Estimation Maximisation
- Do something complicated / unclear with labelled and unlabelled data

The input to the clustering algorithm is the set of sentences S for which clusters are to be extracted, a stop-list W_{Stop} of words around which clusters ought not to be created, and (optionally) a "go list" W_{Go} of words known to be salient in the domain.

1. The sentences, as well as the stop and go lists, are stemmed using the Porter stemmer. [9]
2. Occurence counts C_W are collected for each stem not appearing in W_{Stop}.
3. The total count for stems occuring in W_{Go} is multiplied by a configurable parameter λ_1.
4. The total count for stems with a high tf·idf (calculated over the whole data set) is multiplied by a configurable parameter λ_2.
5. The total count for stems with a high tf·idf (calculated over the data in the given leaf node of the taxonomy) is multiplied by a configurable parameter λ_3.
6. The list of counts is sorted by size.
7. To create a set of N clusters, one cluster is created for each of the most frequent N stems, with all of the sentences containing the stem forming the cluster. The clusters are labeled with the corresponding stem St [4] An optional additional constraint is to require a minimum number M of sentences in each cluster.
8. Two clusters C_1 and C_2 are merged if the overlap of sentences S_{C1C2} contained in both C_1 and C_2 exceeds 50% of the set of sentences in C_1 or C_2. If the labels of C_1 and C_2 form a phrase in the sentences in S_{C1C2}, the new cluster C_{12} is labeled with that phrase, otherwise it is labeled with both labels, separated by a comma.

An overview of the clustering approach is presented in Figure 3. The initial set of clusters is determined by term frequency alone. Go words and the two tf·idf weighting schemes each re-rank the clusters, and finally some of the clusters are merged and a fixed number of clusters is selected off the top of the ranked list for display.

Techniques: Details

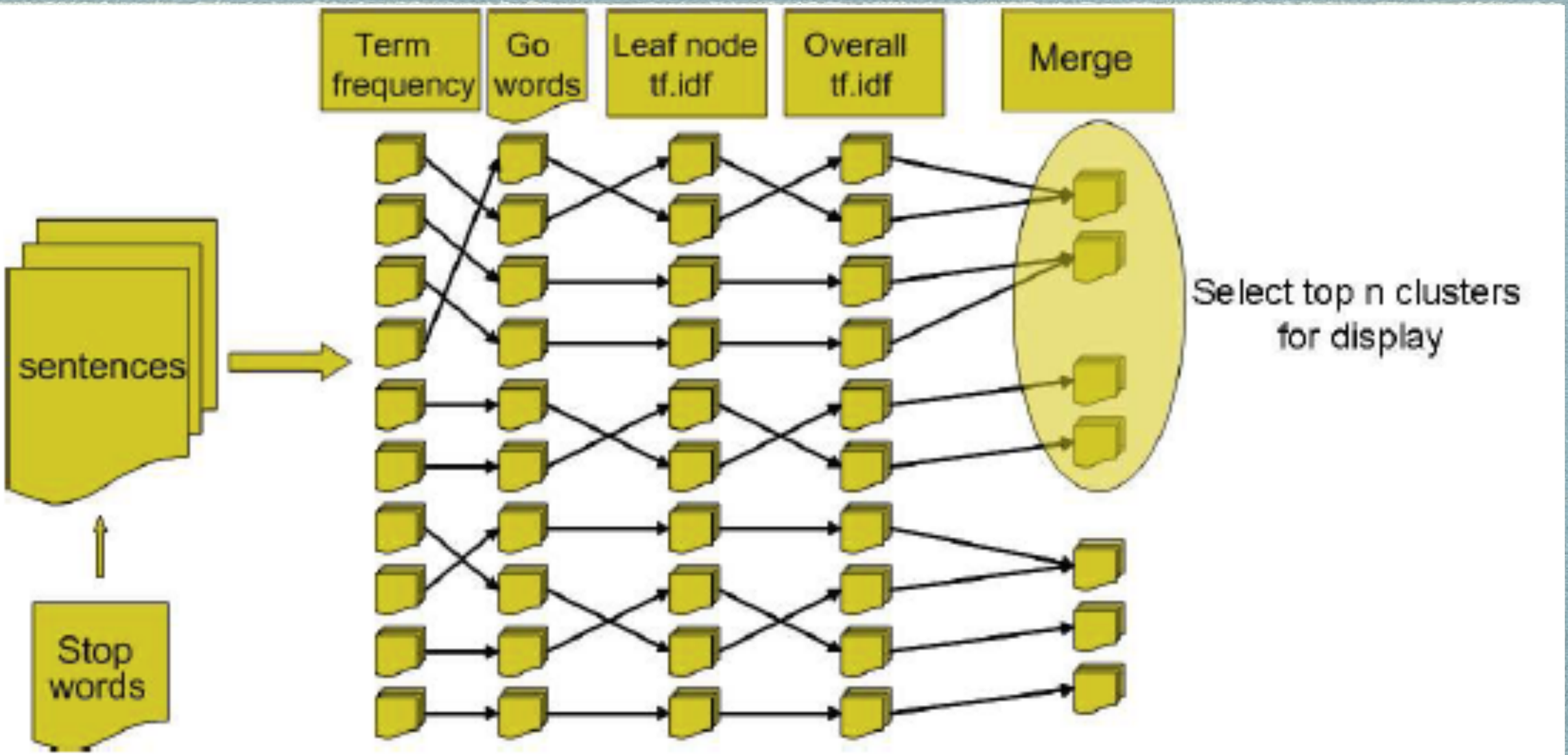


Fig. 3. Diagram of the clustering algorithm

Pulse: Lessons

- Nice complete system
- Tailored clustering and stop-words solutions
- Sentence-level classification, not document-level as in Pang et al.
- Perhaps better evaluation of usefulness

Eg2

- Problem
- Setup & Data
- Techniques
- Results & Findings
- Lessons


From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series

Brendan O'Connor† Ramnath Balasubramanyan† Bryan R. Routledge§ Noah A. Smith†

†School of Computer Science
Carnegie Mellon University

§Tepper School of Business
Carnegie Mellon University

O'Connor et al: Problem

- Does a population like / dislike X (Obama, jobs, ice-cream…)
- Can online text sources (eg blogs, facebook, tweets) replace opinion polls and surveys
- Identify sentiment, how it trends and changes over time (time-based aggregation)

OC: Set-up & Data

- 2-3 years of tweets day-by-day, retrieving on topic (Obama, jobs); 100s-1000s messages a day dep. news
- Simple deterministic model counting positive/ negative words using sentiment lexicon (MPQA)
- If tweet contains a positive count as positive, negative as negative, both count in both (?) (= counting +/- words as Tweets short)
- Track (noisy) trends over time (using smoothening)

O'C: Techniques

We define the sentiment score x_t on day t as the ratio of positive versus negative messages on the topic, counting from that day's messages:

$$
\begin{aligned}
x_t &= \frac{\text{count}_t(\text{pos. word} \wedge \text{topic word})}{\text{count}_t(\text{neg. word} \wedge \text{topic word})} \qquad (1)\\
&= \frac{p(\text{pos. word} \mid \text{topic word}, t)}{p(\text{neg. word} \mid \text{topic word}, t)}
\end{aligned}
$$

where the likelihoods are estimated as relative frequencies.

Moving Average Aggregate Sentiment

Day-to-day, the sentiment ratio is volatile, much more than most polls.[9] Just like in the topic volume plots (Figure 4), the sentiment ratio rapidly rises and falls each day. In order to derive a more consistent signal, and following the same methodology used in public opinion polling, we *smooth* the sentiment ratio with one of the simplest possible temporal smoothing techniques, a moving average over a window of the past k days:

$$MA_t = \frac{1}{k}\left(x_{t-k+1} + x_{t-k+2} + \ldots + x_t\right)$$

Smoothing is a critical issue. It causes the sentiment ratio to respond more slowly to recent changes, thus forcing consistent behavior to appear over longer periods of time. Too much smoothing, of course, makes it impossible to see fine-grained changes to aggregate sentiment. See Figure 5 for an illustration of different smoothing windows for the *jobs* topic.

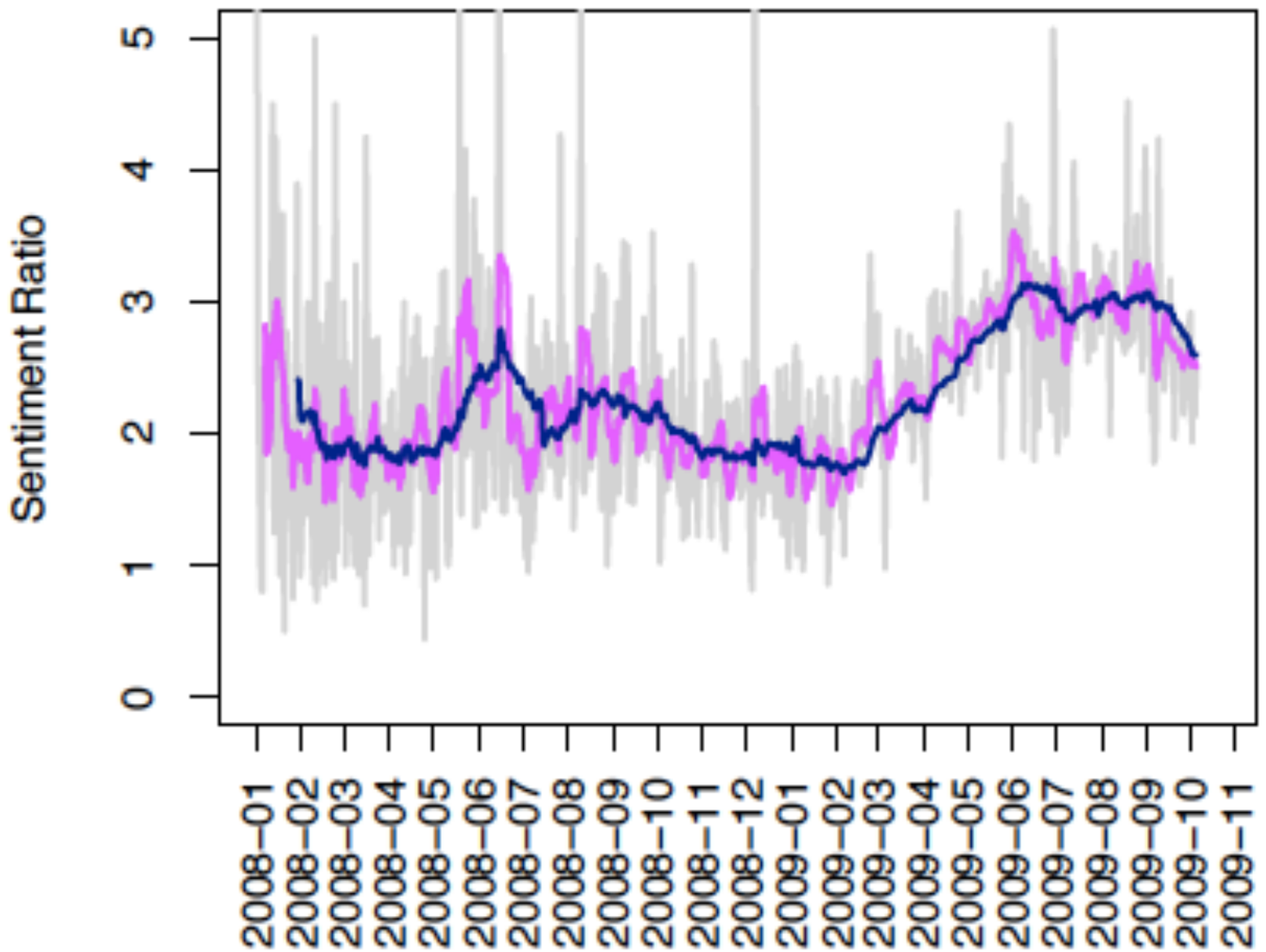


Figure 5: Moving average MA_t of sentiment ratio for *jobs*, under different windows $k \in \{1, 7, 30\}$: no smoothing (gray), past week (magenta), and past month (blue). The unsmoothed version spikes as high as 10, omitted for space.

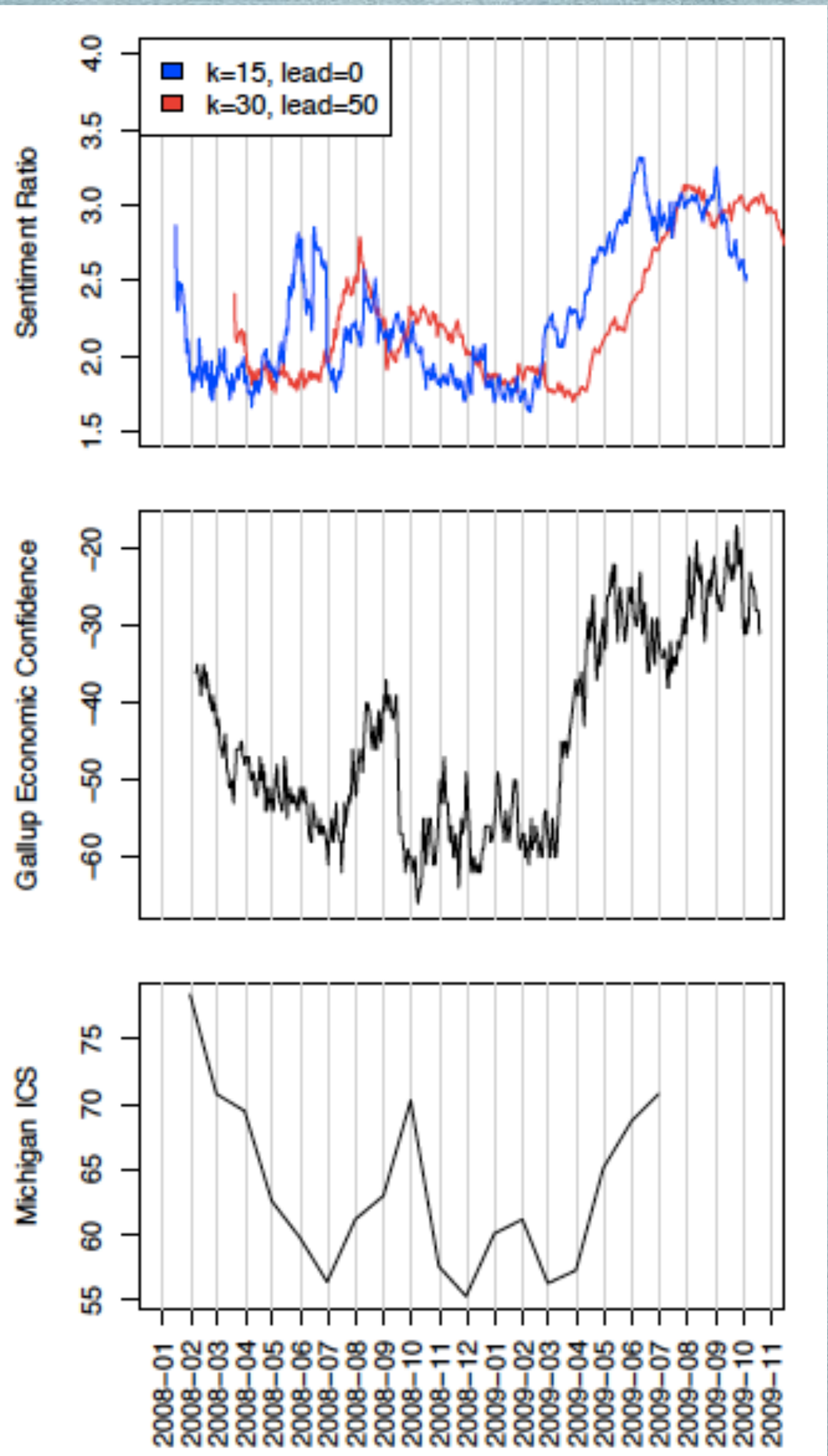
O'Connor, B., Balasubramanyan, R., Routledge, B. R., & Smith, N. A. (2010). From tweets to polls: Linking text sentiment to public opinion time series. ICWSM, 11, 122-129.

Techniques: Issues

- Lexicon-positives contains many NLP errors (verb will, id-ed as positive-noun will)
- Tweet mis-spellings, do not match lexicon
- But, doesn't matter for aggregation; just a noisy detector, with sufficiently large nos. errors cancel out (IR model wrong)
- But, smoothening becomes critical

O'C: Results

- Correlates r = 0.73, at 15 day smoothening with polls
- *r* is goodness-of-fit metric for a one variable linear least-squares model
- Can look at lag, which predicts first; poll/tweets
- Offset line by X days (**lead**) and check correlation
- *r* higher for text leading poll; or poll lagging text



O'C: Lessons

- Find text-measure forecasting accuracy changes over time (with news?/no-of-tweeters?)
- Simple models can work and getting the NLP perfect may not be important (noise cancelling constant)
- Time-series trend tracking is VIMP
- Topic frequency alone (without sentiment) can say a lot about attention (Obama topic-freq correlates with approval-ratings of Obama)

Realtime Sentiment During Presidential Debate

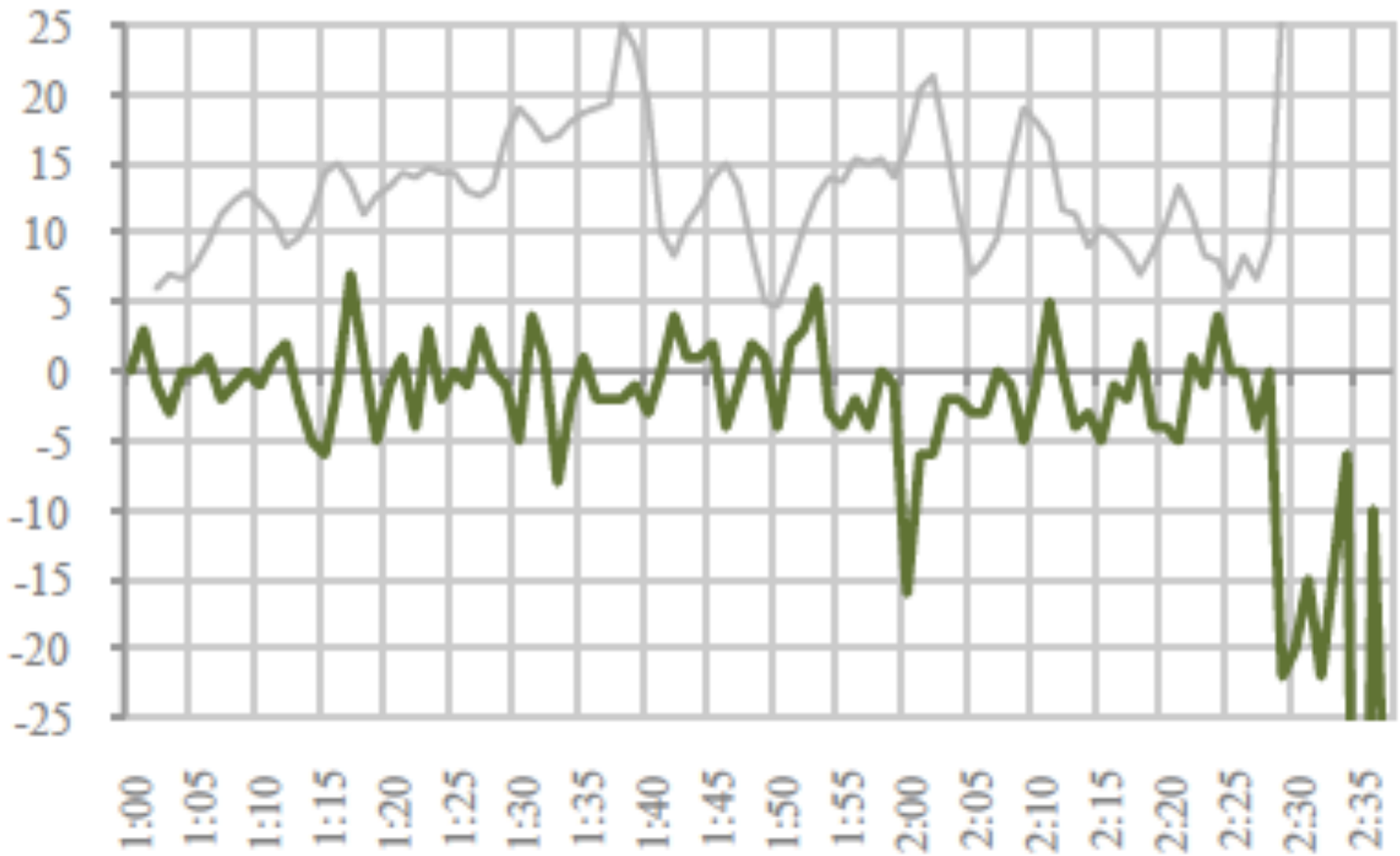


Figure 1. The number of positive minus the number of negative tweets per minute, with 3 minute moving average of total tweet volume in light gray. Times in GMT.

Diakopoulos, N. A., & Shamma, D. A. (2010). Characterizing debate performance via aggregated twitter sentiment. In SIGCHI Conference on Human Factors in Computing Systems, 1195-1198, ACM.

Eg3

- Problem
- Setup & Data
- Techniques
- Results & Findings
- Lessons

Mining the Web for the "Voice of the Herd" to Track Stock Market Bubbles

Aaron Gerow
Trinity College Dublin
Dublin, Ireland
gerowa@tcd.ie

Mark T. Keane
University College Dublin
Belfield, Ireland
mark.keane@ucd.ie

G&K: Problem

- Does language used by a population of journalists show more positivity in a bubble

- G&K has shown distributional change in verb-count power-law distributions (see L4)

- Would there be distributional changes in positive/negative language too

Gerow, A., & Keane, M. T. (2011). Mining the web for the voice of the herd to track stock market bubbles. *IJCAI-11*. AAAI Press.

G&K:

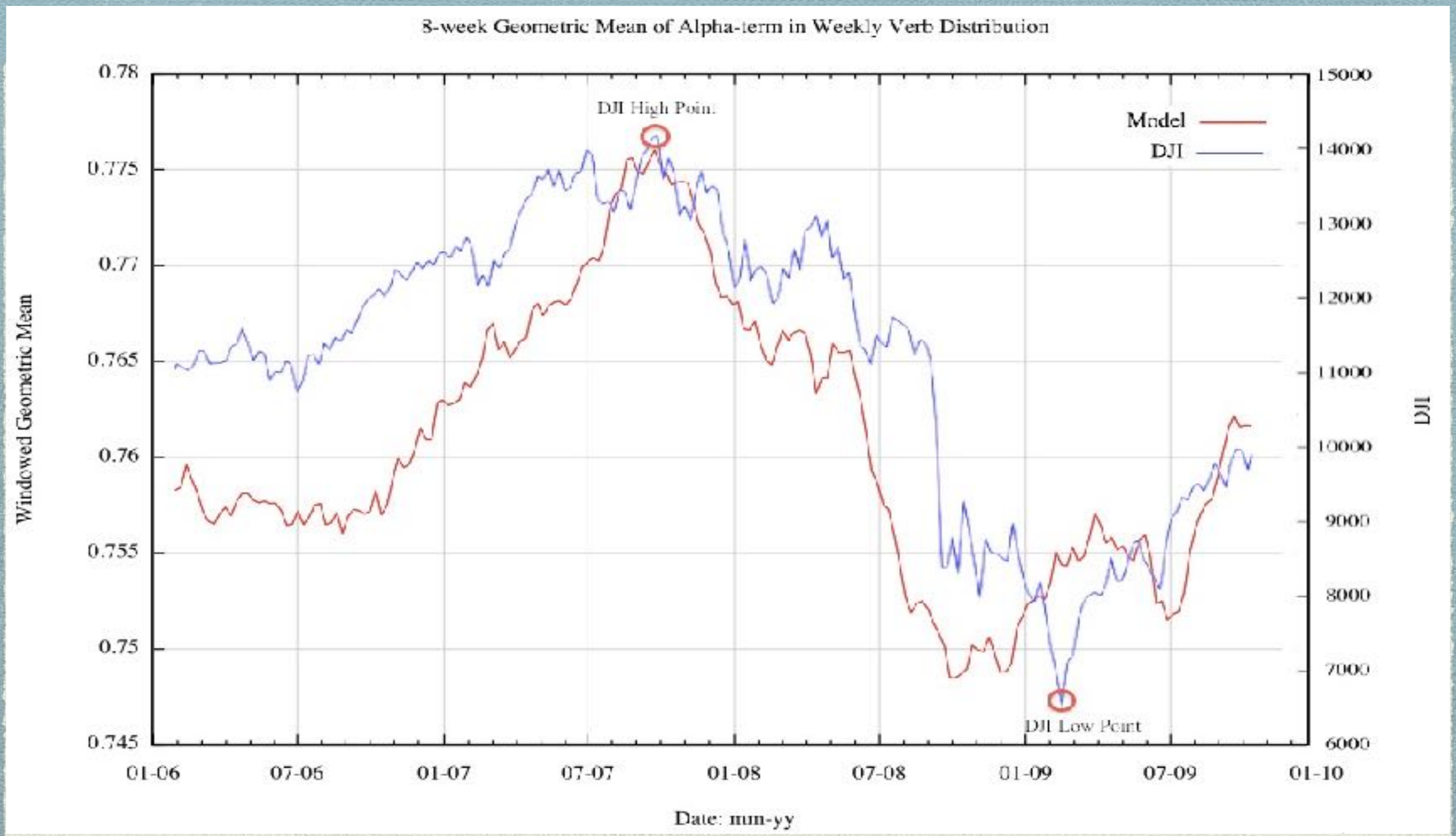
time. As early as the 1940s, George Zipf found that the frequency distribution of words in Moby Dick [Melville, 1851], and other corpora, follow a regular power-law with the generalized form:

$$y = Cx^{-a} \qquad (1)$$

with $C = e^{c}$ [Estoup, 1916; Newman, 2006; Zipf, 1949].

$$y = e^{10.8407}\, x^{-0.8137}$$

$$r = .79 \; ; \; p < .001$$



G&K: Set-up & Data

- Gerow & Keane (2011): 4 years of financial articles 17,713 (FT, NYT, BBC), 5.4M words
- Extracted Lemma-Object Pairs (LOPs): “ris* inflation”, “plung* stocks”, “fail* company”
- Raters judge +/-/neutral; both agreement

Gerow, A., & Keane, M. T. (2011). Mining the web for the voice of the herd to track stock market bubbles. *IJCAI-11*. AAAI Press.

G&K: Techniques

- 3,000 LOP judgements = 49,000 "similar"
- Identical cases and others; "fail* company" gets you "failing company", "failed company", "fails company"…
- But, 14% positive, 27% negative, 59% neutral/unsure (residual category)

Gerow, A., & Keane, M. T. (2011). Mining the web for the voice of the herd to track stock market bubbles. *IJCAI-11*. AAAI Press.

G&K: Techniques

- Get distribution of all words in sample of classified words for full period of graph

- Get distributions of positive and negative words for each week in graph-period

- See whether K-L divergence is greater for positive or negative dists. at different times

Gerow, A., & Keane, M. T. (2011). Mining the web for the voice of the herd to track stock market bubbles. *IJCAI-11*. AAAI Press.

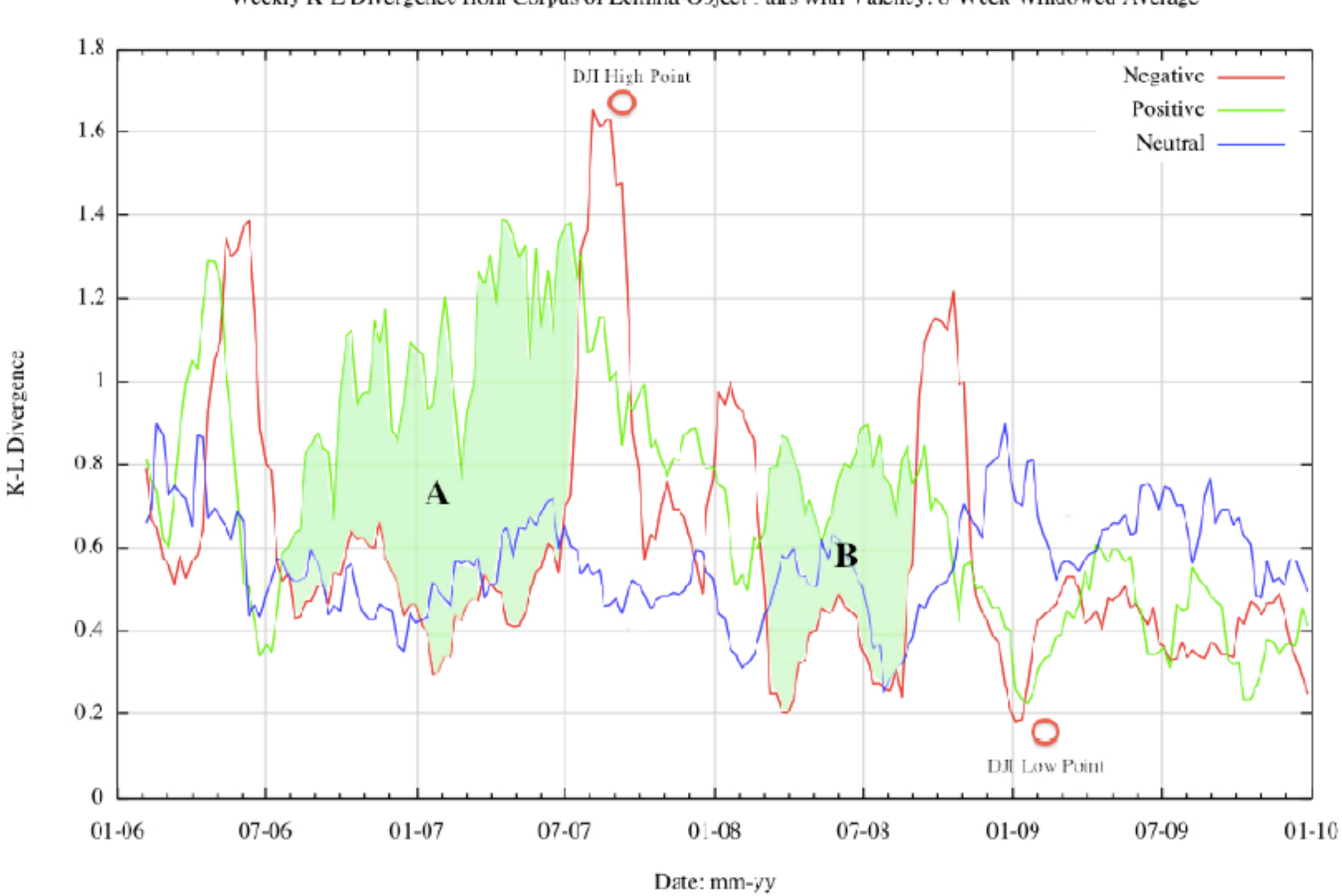


Figure 2: *Symmetric K-L divergence (8-week windowed mean) of positive, negative, and neutral lemma-object pairs. Note, the two regions, A and B, of distinct positive-negative divergence preceding the 2007 crash and subsequently the beginning of the recovery in 2009.*

Gerow, A., & Keane, M. T. (2011). Mining the web for the voice of the herd to track stock market bubbles. *IJCAI-11*. AAAI Press.

G&K: Results

- The relative distributions of positive and negative language change systematically in bubble periods
- These changes are reflected in changes in stock-market indices (like DJIA)
- Can talk of regions of positivity or negativity (see A and B)

G&K: Lessons

- K-L can be used to compare any 2 distributions (eg, to track topic changes)

- Here, we are looking at distributions of sentiment terms

- NB. This work is novel in comparing distributions, not examining just one distribution (as most do)

Using Sentiment:

Technique Tangent

Kullback–Leibler divergence

From Wikipedia, the free encyclopedia

Not to be confused with divergence in calculus.

In probability theory and information theory, the **Kullback–Leibler divergence**[1][2][3] (also **information divergence**, **information gain**, **relative entropy**, or **KLIC**; here abbreviated as KL divergence) is a non-symmetric measure of the difference between two probability distributions P and Q. Specifically, the Kullback–Leibler divergence of Q from P, denoted $D_{\mathrm{KL}}(P\|Q)$, is a measure of the information lost when Q is used to approximate P:[4] The KL divergence measures the expected number of extra bits required to code samples from P when using a code based on Q, rather than using a code based on P. Typically P represents the "true" distribution of data, observations, or a precisely calculated theoretical distribution. The measure Q typically represents a theory, model, description, or approximation of P.

Although it is often intuited as a metric or distance, the KL divergence is not a true metric — for example, it is not symmetric: the KL divergence from P to Q is generally not the same as that from Q to P. However, its infinitesimal form, specifically its Hessian, is a metric tensor: it is the Fisher information metric.

KL divergence is a special case of a broader class of divergences called *f*-divergences. It was originally introduced by Solomon Kullback and Richard Leibler in 1951 as the **directed divergence** between two distributions. It can be derived from a Bregman divergence.

Definition [edit]

For discrete probability distributions P and Q, the KL divergence of Q from P is defined to be

$$D_{\mathrm{KL}}(P\|Q) = \sum_i \ln\left(\frac{P(i)}{Q(i)}\right) P(i).$$

In words, it is the expectation of the logarithmic difference between the probabilities P and Q, where the expectation is taken using the probabilities P. The KL divergence is only defined if P and Q both sum to 1 and if $Q(i) = 0$ implies $P(i) = 0$ for all i (absolute continuity). If the quantity $0 \ln 0$ appears in the formula, it is interpreted as zero because $\lim_{x \to 0} x \ln(x) = 0$.

K-L Tangent

- K-L divergence, also called “relative entropy”, is not a metric; it can be asymmetric the divergence of P from Q may not the same as Q from P

- K-L distance/metric sums P-Q and Q-P KL-div measures for symmetry

- Back-off smoothening to handle absent words (Migge, 2003)

- http://staff.science.uva.nl/~tsagias/?p=185

- http://www.biology-online.org/biology-forum/about8270.html

Simple Roll of the Dice…

- Typically, K-L divergence compares a "true" , theoretical distribution to an observed distribution

- Imagine, an 6-sided dice

- "true" distribution is 1/6 chance of getting a given number (1-6)

- Observed distribution is what I get when I roll it, say, 1000 times

Simple Dice Eg

	P	Q	
Dice No	Observed	Theoretical	
1	30%	16.7%	0.176
2	15%	16.7%	-0.016
3	5%	16.7%	-0.060
4	5%	16.7%	-0.060
5	10%	16.7%	-0.051
6	35%	16.7%	0.259
Sum ->			0.247

$$D_{\mathrm{KL}}(P\|Q) = \sum_i \ln\left(\frac{P(i)}{Q(i)}\right) P(i).$$

P(1)		=	0.300
Q(1)		=	0.167
ln(P(1)/Q(1))		=	0.586
ln(P(1)/Q(1)) * P(1)		=	0.176
Using Natural Logs not Log base 2			

- P is "actual" distribution
- Q is "model" distribution
- how "good" is Q as a model of P
- nb. language model

Simple Dice Eg

Dodgy Dice ?

	P	Q	
Dice No	Observed	Theoretical	
1	30%	16.7%	0.176
2	15%	16.7%	-0.016
3	5%	16.7%	-0.060
4	5%	16.7%	-0.060
5	10%	16.7%	-0.051
6	35%	16.7%	0.259
Sum ->			0.247

$$D_{\mathrm{KL}}(P\|Q) = \sum_i \ln\left(\frac{P(i)}{Q(i)}\right) P(i).$$

P(1)		=	0.300
Q(1)		=	0.167
ln(P(1)/Q(1))		=	0.586
ln(P(1)/Q(1)) * P(1)		=	0.176

Using Natural Logs not Log base 2

Good Dice ?

	P	Q	
Dice No	Observed	Theoretical	
1	16%	16.7%	-0.007
2	16%	16.7%	-0.007
3	18%	16.7%	0.013
4	18%	16.7%	0.013
5	13%	16.7%	-0.033
6	19%	16.7%	0.025
Sum ->			0.005

$$D_{\mathrm{KL}}(P\|Q) = \sum_i \ln\left(\frac{P(i)}{Q(i)}\right) P(i).$$

P(1)		=	0.160
Q(1)		=	0.167
ln(P(1)/Q(1))		=	-0.043
ln(P(1)/Q(1)) * P(1)		=	-0.007

Using Natural Logs not Log base 2

Simple Word Eg

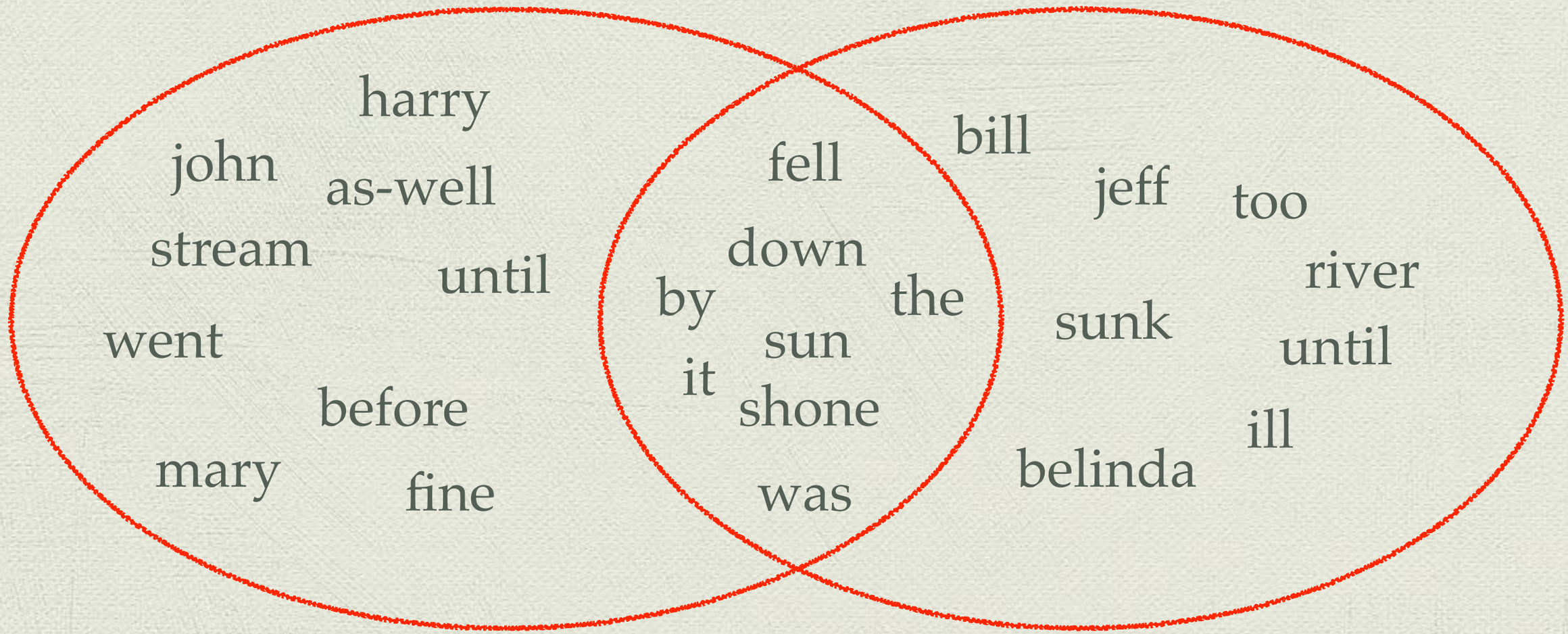
john fell down
harry fell as-well
down by the stream
the sun shone before
it went down
mary was fine

{john-1, fell-2,down-3,
harry-1, as-well-1…}

bill fell down
jeff fell too
down by the river
the sun shone until
it sunk down
belinda was ill

{bill-1, fell-2,down-3,
jeff-1, too-1…}

Simple Word Eg



{john-1, fell-2,down-3,
harry-1, as-well-1…}

{bill-1, fell-2,down-3,
jeff-1, too-1…}

Using Sentiment:

Predicting Behaviour

Using Sentiment

REM

- Sentiment used in many different ways:
  - to reflect opinion of a population
  - to predict people's behaviour
  - to make recommendations to others

Prediction Using…

- We have just looked at tracking opinion
- But, opinion should predict behaviour; things I like are the things I buy…
- So, how does opinion get used to predict !

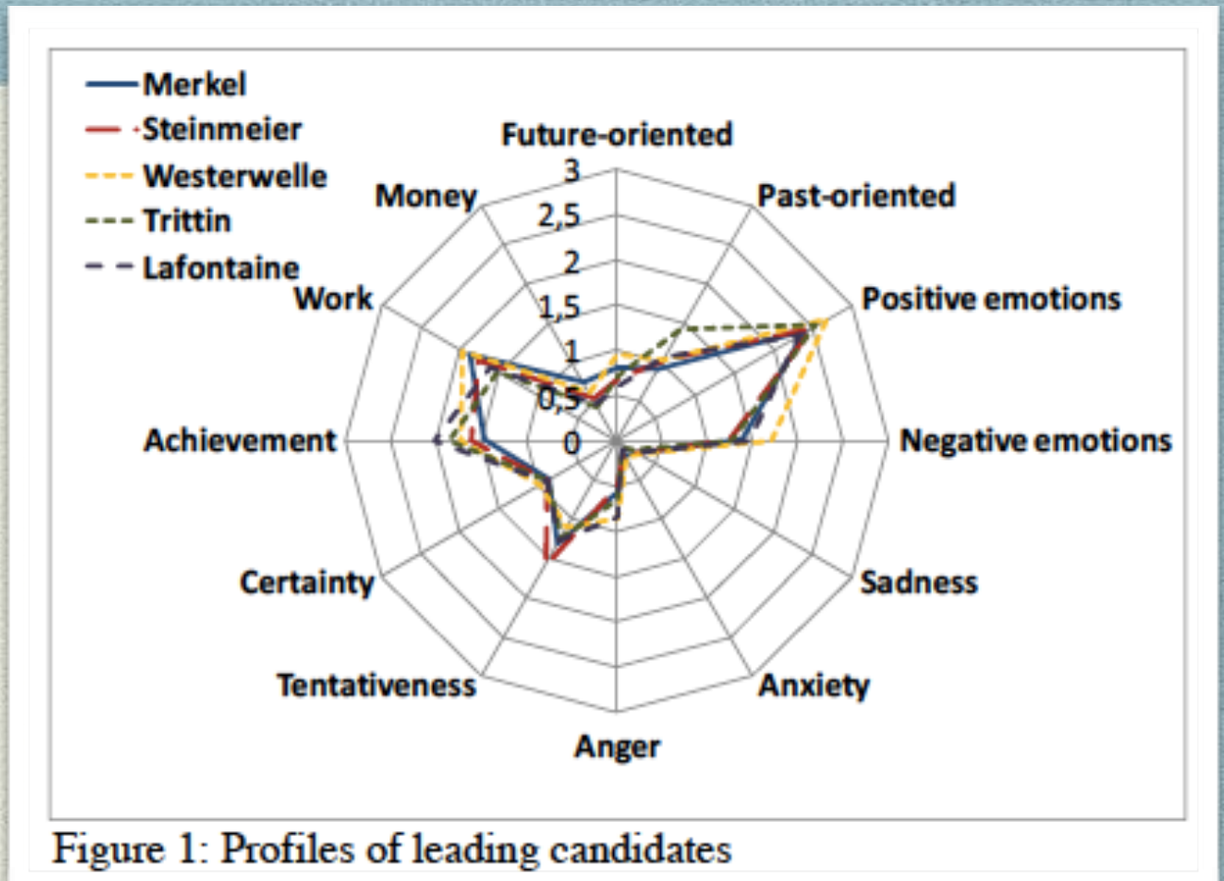


Figure 1: Profiles of leading candidates

Tumasjan, A., Sprenger, T. O., Sandner, P. G., & Welpe, I. M. (2010). Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment. ICWSM, 10, 178-185.

Two Examples

Predicting the Future With Social Media

Sitaram Asur
Social Computing Lab
HP Labs
Palo Alto, California
Email: sitaram.asur@hp.com

Bernardo A. Huberman
Social Computing Lab
HP Labs
Palo Alto, California
Email: bernardo.huberman@hp.com

Twitter mood predicts the stock market.

Johan Bollen[1,⋆],Huina Mao[1,⋆],Xiao-Jun Zeng[2].

⋆: authors made equal contributions.

Eg1

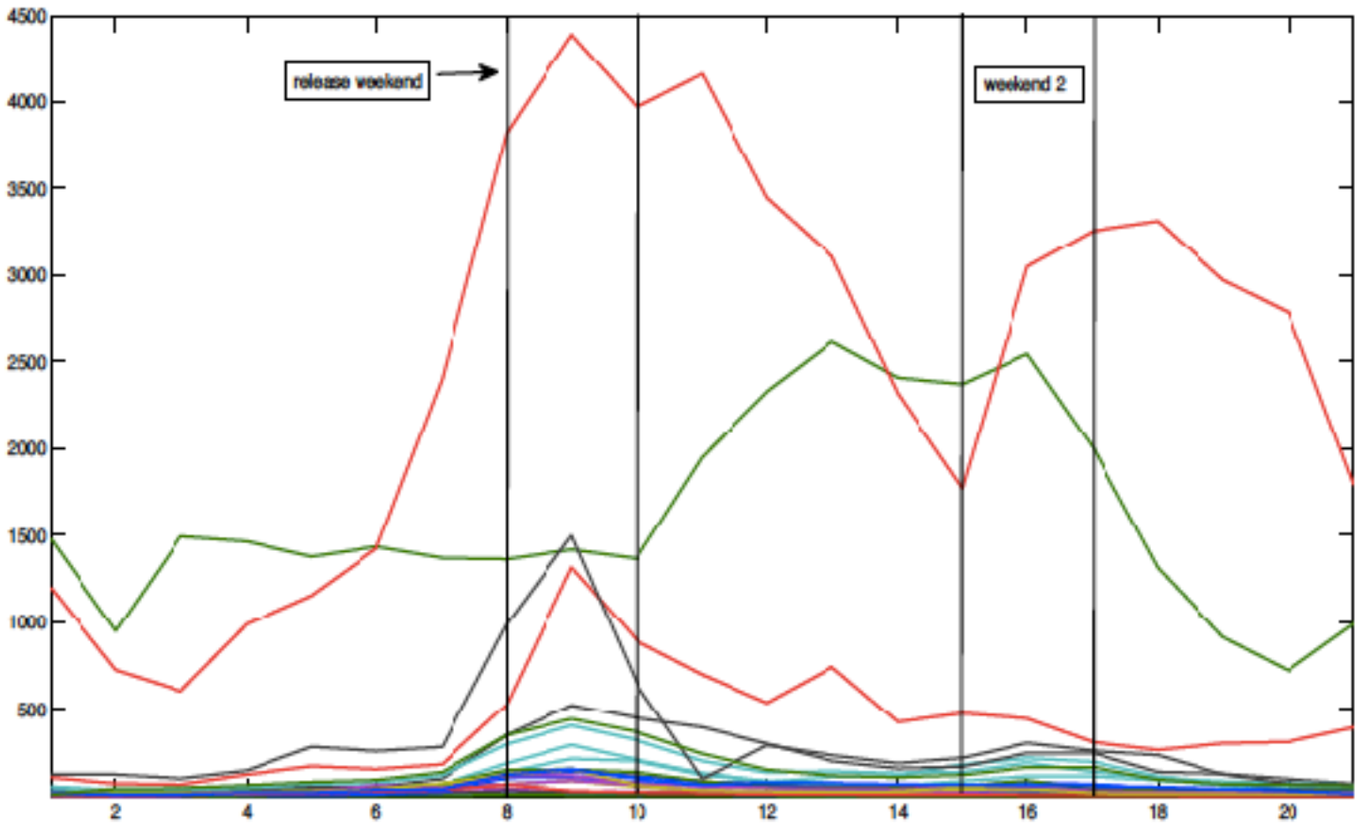
- Problem
- Setup & Data
- Architecture & Techniques
- Results & Findings
- Lessons

Eg1: Problem

- Predicting movie box-office is important; esp. in real-time; can prompt more promotion
  - Twitter is a good real-time measure
  - Often, sheer tweet volume is enough
  - Tweet pos/neg provides more info
- Doing prediction has extra steps…

Asur, S., & Huberman, B. A. (2010). Predicting the future with social media. In Web Intelligence and Intelligent Agent Technology (WI-IAT), 492-499. IEEE.

Set-up & Data



1. Time-series of tweets over the critical period for different movies.

Movie	Release Date
Armored	2009-12-04
Avatar	2009 12 18
The Blind Side	2009-11-20
The Book of Eli	2010 01 15
Daybreakers	2010 01 08
Dear John	2010 02 05
Did You Hear About The Morgans	2009 12 18
Edge Of Darkness	2010 01 29
Extraordinary Measures	2010 01 22
From Paris With Love	2010-02-05
The Imaginarium of Dr Parnassus	2010 01 08
Invictus	2009-12-11
Leap Year	2010-01-08
Legion	2010-01-22
Twilight : New Moon	2009-11-20
Pirate Radio	2009-11-13
Princess And The Frog	2009-12-11
Sherlock Holmes	2009-12-25
Spy Next Door	2010-01-15
The Crazies	2010-02-26
Tooth Fairy	2010 01 22
Transylmania	2009-12-04
When In Rome	2010-01-29
Youth In Revolt	2010-01-08

Sentiment: Setup

- Used Language Model classifier using n-grams(8); tweets crowdsourced labels (pos/neg/neutral) on AMT
- Pre-processed tweets; removing stops, urls, user-name, special characters and movie titles
- Classifier then indicates if tweet is pos/neg/neutral
- Computed these rates using *Subjectivity* and *Polarity* measures for each movie; did regression to predict movie box-office from twitter sentiments

A. Subjectivity

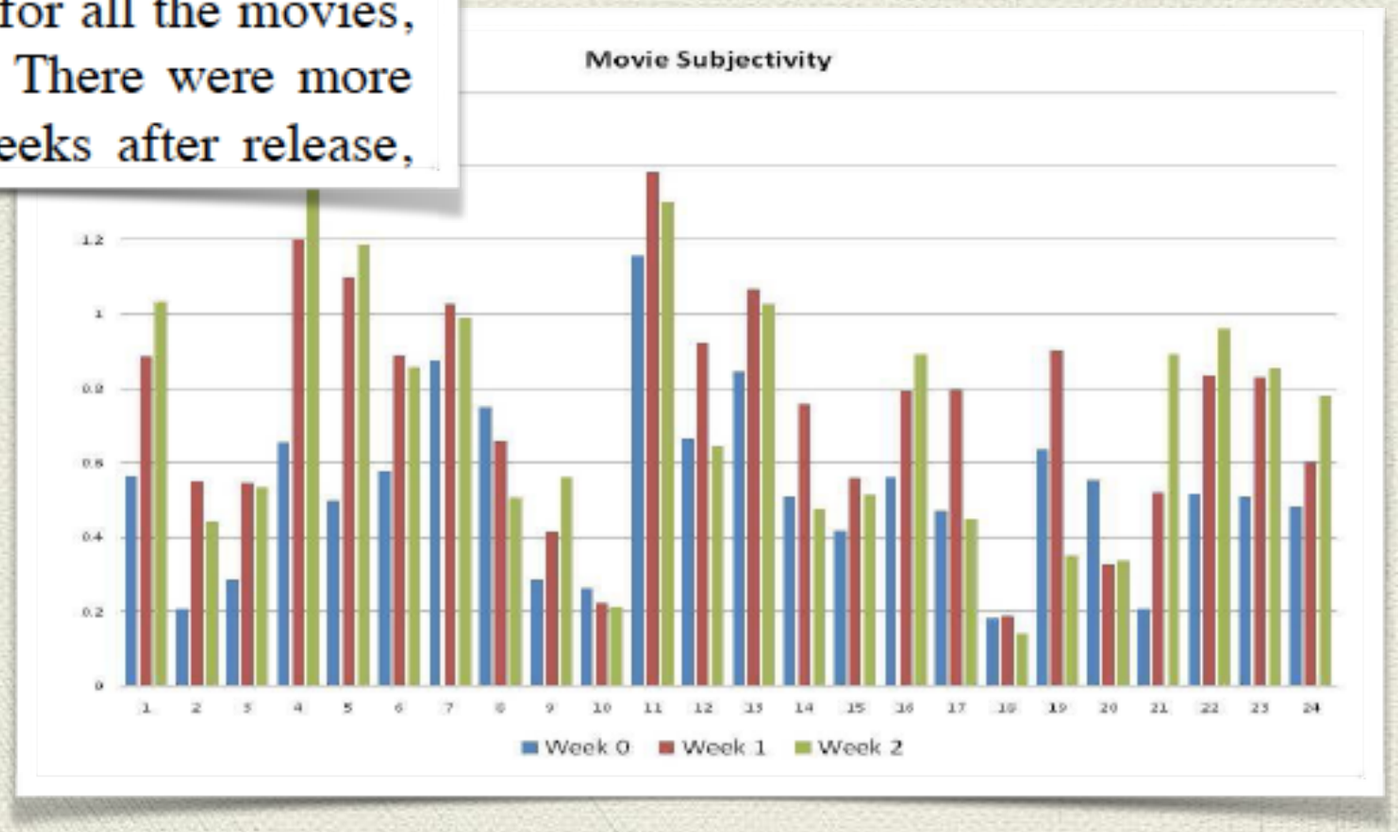
Our expectation is that there would be more value for sentiments after the movie has released, than before. We expect tweets prior to the release to be mostly anticipatory and stronger positive/negative tweets to be disseminated later following the release. Positive sentiments following the release can be considered as recommendations by people who have seen the movie, and are likely to influence others from watching the same movie. To capture the subjectivity, we defined a measure as follows.

$$Subjectivity = \frac{|Positive\ and\ Negative\ Tweets|}{|Neutral\ Tweets|} \quad (2)$$

When we computed the subjectivity values for all the movies, we observed that our hypothesis was true. There were more sentiments discovered in tweets for the weeks after release,

Asur & Huberman (2010)

Subjectivity



B. Polarity

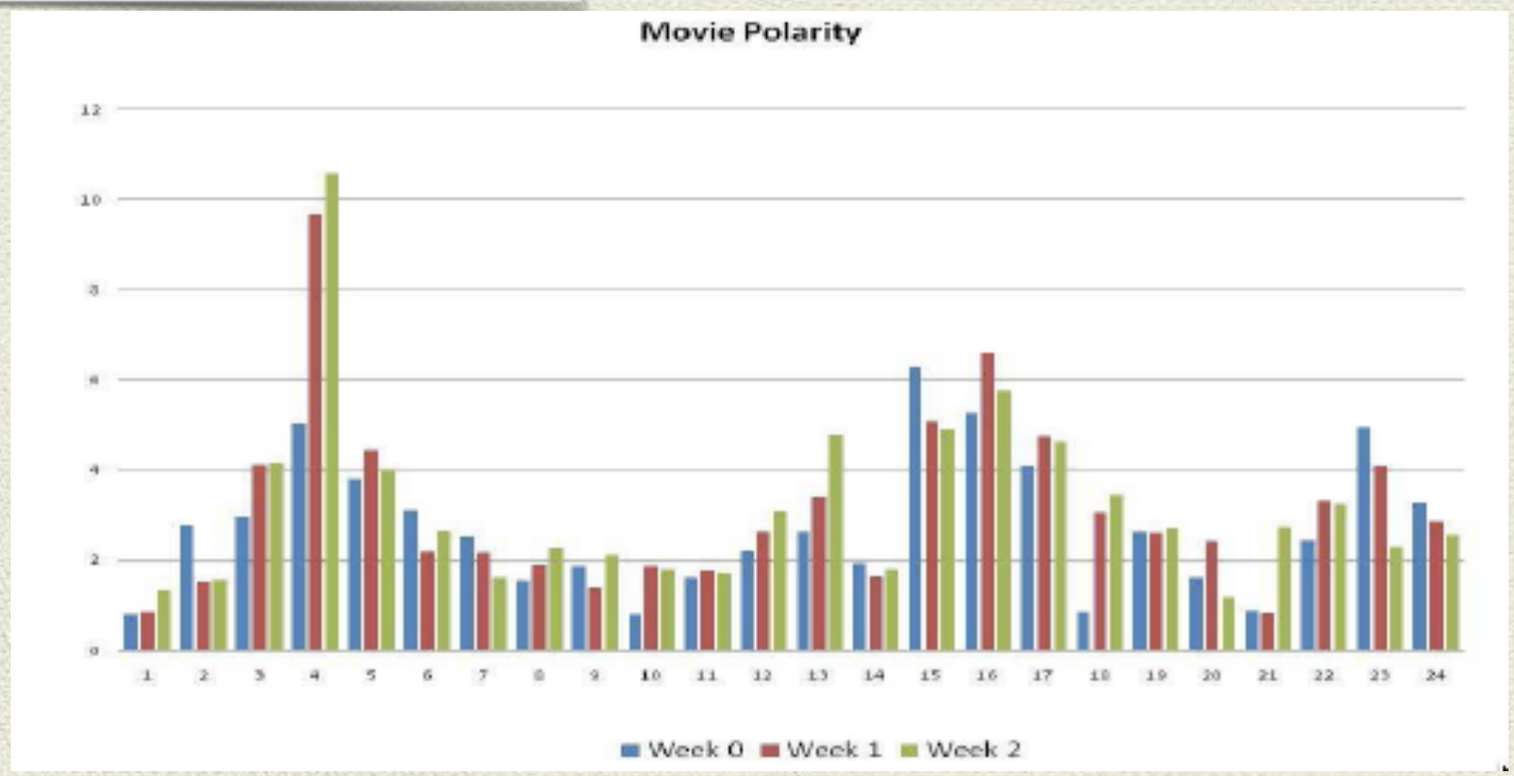
To quantify the sentiments for a movie, we measured the ratio of positive to negative tweets. A movie that has far more positive than negative tweets is likely to be successful.

$$PNratio = \frac{|Tweets\ with\ Positive\ Sentiment|}{|Tweets\ with\ Negative\ Sentiment|} \quad (3)$$

Fig 8 shows the polarity values for the movies considered in the critical period. We find that there are more positive sentiments than negative in the tweets for almost all the movies. The movie with the enormous increase in positive sentiment after release is *The Blind Side* (5.02 to 9.65). The movie had a lukewarm opening weekend sales (34M) but then boomed in the next week (40.1M), owing largely to positive

Asur & Huberman (2010)

Polarity



Prediction: Using Regression

Variable	$p - value$
(Intercept)	0.542
Avg Tweet-rate	2.05e-11 (***)
PNRatio	9.43e-06 (***)

TABLE IX
REGRESSION USING THE AVERAGE TWEET-RATE AND THE POLARITY (PNRATIO). THE SIGNIFICANCE LEVEL (*:0.05, **: 0.01, ***: 0.001) IS ALSO SHOWN.

Asur, S., & Huberman, B. A. (2010). Predicting the future with social media. In Web Intelligence and Intelligent Agent Technology (WI-IAT), 492-499. IEEE.

Movies: Lessons

- Tweet rate on its own can work quite well
- Though, prediction can be improved by adding sentiment; nb how it is aggregated in two different ways: subjectivity and polarity
- Then these measures in-turn can be used as predictive indices

Eg2

- Problem
- Setup & Data
- Techniques
- Results & Findings
- Lessons


Twitter mood predicts the stock market.

Johan Bollen[1,*], Huina Mao[1,*], Xiao-Jun Zeng[2].

*: authors made equal contributions.

Bollen et al: Problem

- Whether collective mood states in large-scale twitter feeds predict stock markets (DJIA)
- Mood tracking tools: OpinionFinder, Google Profile of Mood States (GPMS); 6 moods Calm, Alert, Sure, Vital, Kind, and Happy
- Use Self-Organising, Fuzzy NN as predictor

Pollen et al.: Set-up & Data

- 9.8M tweets from 2.7 M users (2008)
- Remove stop words, punctuation; group these by date (excl. SPAM as http, www)
- Different time series for each of 7 moods
- Verify correlation between mood measures and DJI changes using Grainger Causality for Presidential event

Bollen et al: check

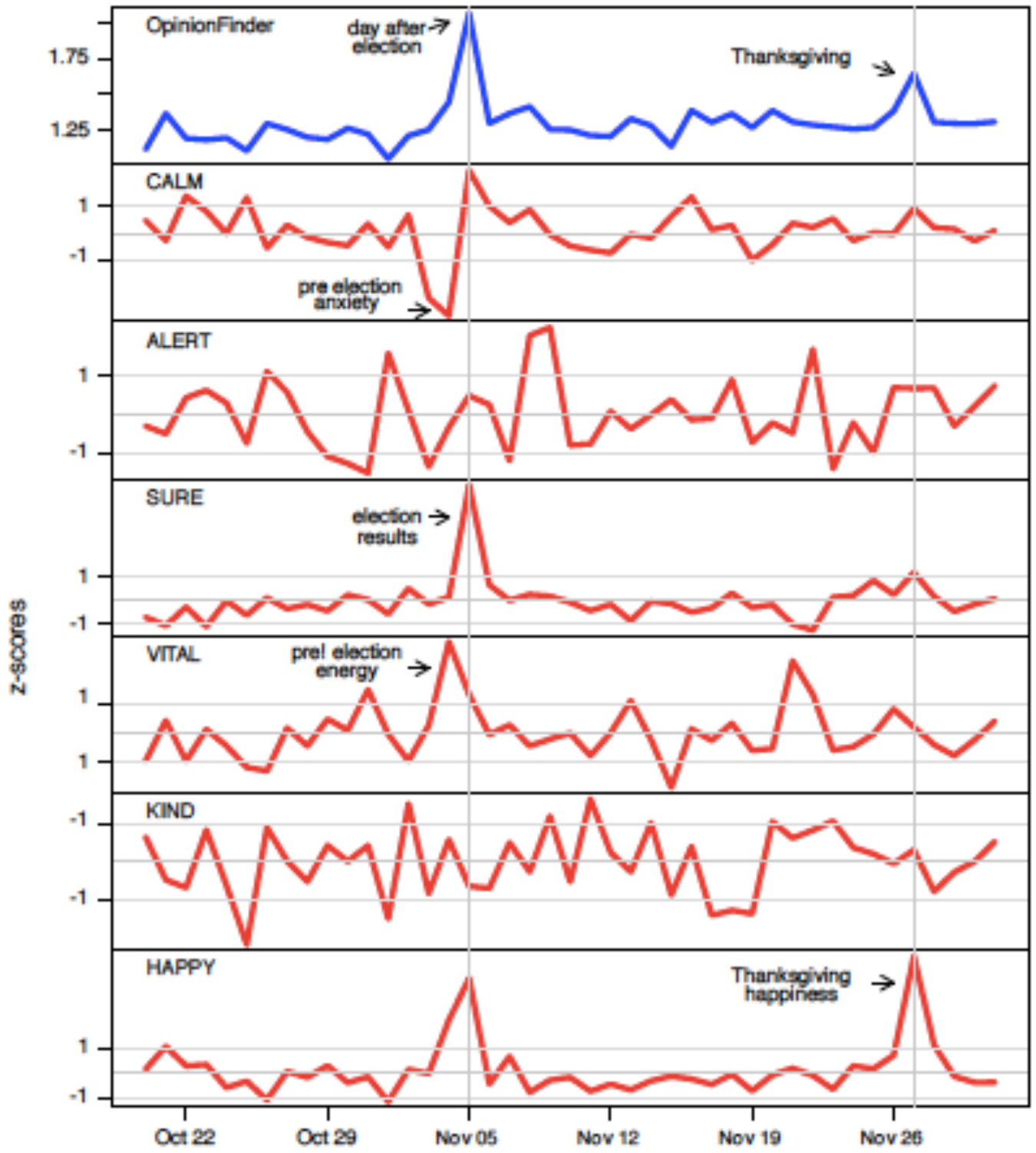


Fig. 2. Tracking public mood states from tweets posted between October 2008 to December 2008 shows public responses to presidential election and thanksgiving.

Normalising Measures

To enable the comparison of OF and GPOMS time series we normalize them to z-scores on the basis of a local mean and standard deviation within a sliding window of k days before and after the particular date. For example, the z-score of time series X_t, denoted $\mathbb{Z}_{X_t}$, is defined as:

$$\mathbb{Z}_{X_t} = \frac{X_t - \bar{x}(X_{t\pm k})}{\sigma(X_{t\pm k})} \tag{1}$$

where $\bar{x}(X_{t\pm k})$ and $\sigma(D_{t\pm k})$ represent the mean and standard deviation of the time series within the period $[t-k, t+k]$. This normalization causes all time series to fluctuate around a zero mean and be expressed on a scale of 1 standard deviation.

Bollen et al: System

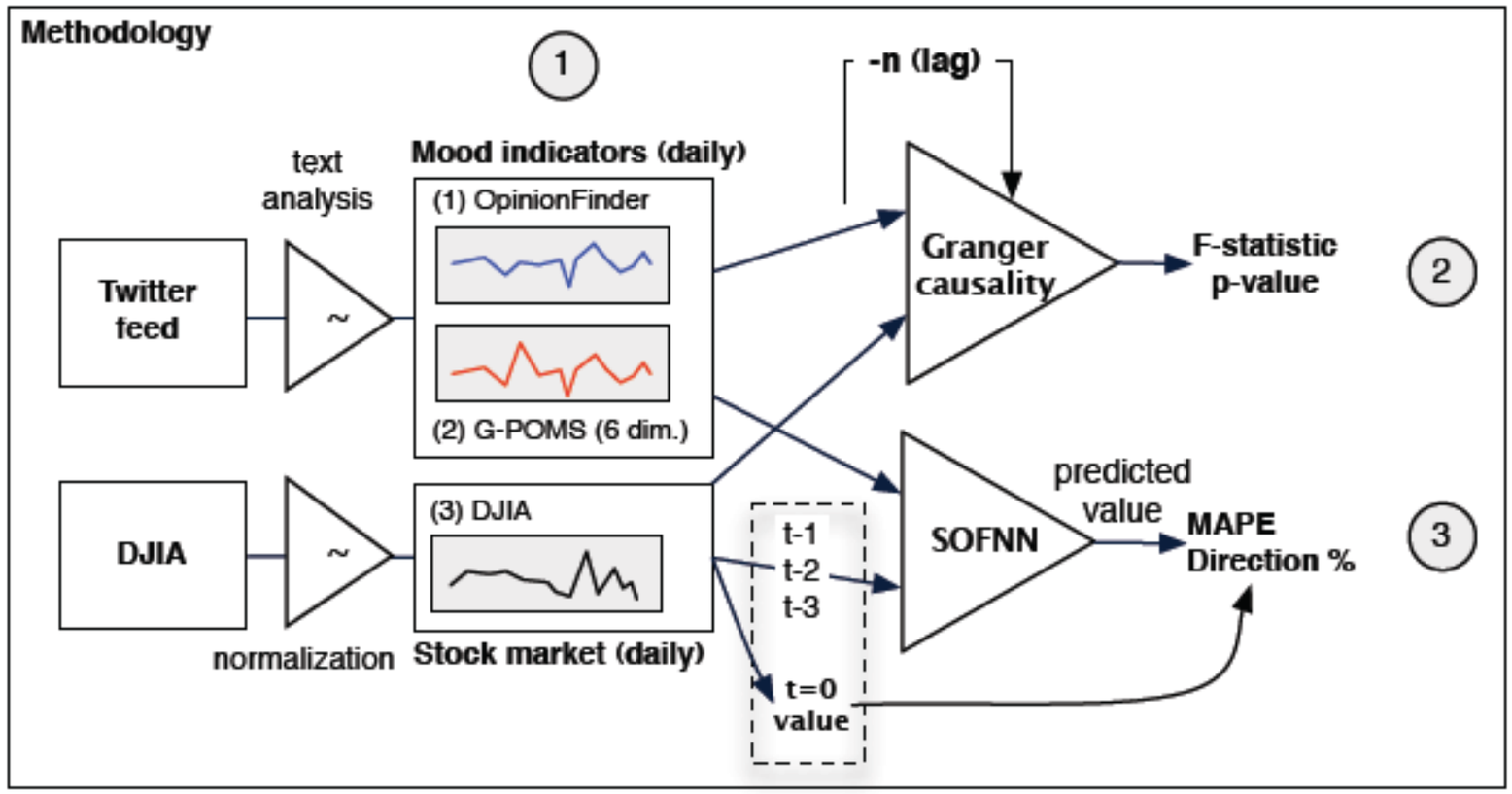


Fig. 1. Diagram outlining 3 phases of methodology and corresponding data sets: (1) creation and validation of OpinionFinder and GPOMS public mood time series from October 2008 to December 2008 (Presidential Election and Thanksgiving), (2) use of Granger causality analysis to determine correlation between DJIA, OpinionFinder and GPOMS public mood from August 2008 to December 2008, and (3) training of a Self-Organizing Fuzzy Neural Network to predict DJIA values on the basis of various combinations of past DJIA values and OF and GPOMS public mood data from March 2008 to December 2008.

Bollen et al.: Results

TABLE I
MULTIPLE REGRESSION RESULTS FOR OPINIONFINDER VS. 6 GPOMS MOOD DIMENSIONS.

Parameters	Coeff.	Std.Err.	t	p
Calm (X_1)	1.731	1.348	1.284	0.20460
Alert (X_2)	0.199	2.319	0.086	0.932
Sure (X_3)	3.897	0.613	6.356	**4.25e-08** $\star\star\star$
Vital (X_4)	1.763	0.595	2.965	0.004$\star\star$
Kind (X_5)	1.687	1.377	1.226	0.226
Happy (X_6)	2.770	0.578	4.790	**1.30e-05** $\star\star$
Summary	Residual Std.Err	Adj.R^2	$F_{6,55}$	p
	0.078	0.683	22.93	2.382e-13

(p-value < 0.001: $\star\star\star$, p-value < 0.05: $\star\star$, p-value < 0.1: $\star$)

Using Sentiment:

Recommending Things

Using Sentiment

REM

- Sentiment used in many different ways:
  - to reflect opinion of a population
  - to predict people's behaviour
  - to make recommendations to others

The Great Feedback Step

- Old-AI used to think you had to have knowledge and perfect model…
- But, a bad model with feedback is just as good (if not better); Google spelling, Google mobile voice recognition…
- Recommenders are another instance of this…

Recommending Things…

- Now, we look at recommending from opinions

- We are often guided by what other's think about something (esp. if they are like us, friends, people we admire)

- Recommender systems try to automate this process: collaborative, content, hybrid (mixed)

Collaborative Recommenders

- Analyse users' previous behaviour and similar decisions of other users (e.g., purchases, likes); use model to predict future items of interest
- Often, depends on explicit user feedback… ratings, thumbs-up without analysing items
- Used by last.fm, MySpace, Facebook & Amazon's famous people who bought this also bought…

Amazon Patent 1998 for Collaborative Recommendation

United States Patent **6,266,649**
Linden , et al. **July 24, 2001**

Collaborative recommendations using item-to-item similarity mappings

Abstract

A recommendations service recommends items to individual users based on a set of items that are known to be of interest to the user, such as a set of items previously purchased by the user. In the disclosed embodiments, the service is used to recommend products to users of a merchant's Web site. The service generates the recommendations using a previously-generated table which maps items to lists of "similar" items. The similarities reflected by the table are based on the collective interests of the community of users. For example, in one embodiment, the similarities are based on correlations between the purchases of items by users (e.g., items A and B are similar because a relatively large portion of the users that purchased item A also bought item B). The table also includes scores which indicate degrees of similarity between individual items. To generate personal recommendations, the service retrieves from the table the similar items lists corresponding to the items known to be of interest to the user. These similar items lists are appropriately combined into a single list, which is then sorted (based on combined similarity scores) and filtered to generate a list of recommended items. Also disclosed are various methods for using the current and/or past contents of a user's electronic shopping cart to generate recommendations. In one embodiment, the user can create multiple shopping carts, and can use the recommendation service to obtain recommendations that are specific to a designated shopping cart. In another embodiment, the recommendations are generated based on the current contents of a user's shopping cart, so that the recommendations tend to correspond to the current shopping task being performed by the user.

Inventors: **Linden; Gregory D.** (Seattle, WA), **Jacobi; Jennifer A.** (Seattle, WA), **Benson; Eric A.** (Seattle, WA)
Assignee: **Amazon.Com, Inc.** (Seattle, WA)
Family ID: **22562727**
Appl. No.: **09/157,198**
Filed: **September 18, 1998**

Content Recommenders

- Properties of items used to recommend similar items, from description of item (maybe user prefs)
- Recommends similar items to past purchases (extracted content features in product description)
- Depends on content-analysis to extract meta-data about items (eg genre, director, actors in a movie) with/without explicit ratings

Two Examples

Comparative Experiments on Sentiment Classification for Online Product Reviews

Hang Cui *
Department of Computer Science
School of Computing
National University of Singapore
cuihang@comp.nus.edu.sg

Vibhu Mittal
Google Inc.
vibhu@google.com

Mayur Datar
Google Inc.
mayur@google.com

Learning to Recommend Helpful Hotel Reviews

Michael P. O'Mahony
CLARITY: Centre for Sensor Web Technologies
School of Computer Science and Informatics
University College Dublin, Ireland
michael.p.omahony@ucd.ie

Barry Smyth
CLARITY: Centre for Sensor Web Technologies
School of Computer Science and Informatics
University College Dublin, Ireland
barry.smyth@ucd.ie

Eg1

- Problem
- Setup & Data
- Techniques
- Results & Findings
- Lessons

Comparative Experiments on Sentiment Classification for Online Product Reviews

Hang Cui *
Department of Computer Science
School of Computing
National University of Singapore
cuihang@comp.nus.edu.sg

Vibhu Mittal
Google Inc.
vibhu@google.com

Mayur Datar
Google Inc.
mayur@google.com

Cui et al: Problem

- Seminal on ML classification methods for sentiments in large data sets of reviews; 200k
- Using n-gram (3 or more); Pang et al (2002) showed poorer perf. beyond 2-grams but from small data sets (but uni-/bi- do not capture longer range dependencies); *excellent juicer…excellent maker of noise*
- Consider, Winnow (linear classifier), generative-language model (probability of generating a word sequence), SVM-type discriminative classifier

Cui, H., Mittal, V., & Datar, M. (2006, July). Comparative experiments on sentiment classification for online product reviews. In AAAI (Vol. 6, pp. 1265-1270).

Cui et al: Set-up & Data

- Over 320k online reviews, over 80k products, crawled from the Web; use user-ratings to identify pos/neg labels
- Higher-order n-grams (up to 6) should be more discriminative of content; better capture surface patterns; *highly recommend staying away from this*
- Produced 3.2M n-grams, removing those that appear in <20 reviews; do Chi2 analysis and pick top-M n-grams as features for classifiers (M = 50k/100k/200k)
- Track (noisy) trends over time (using smoothening)

Cui et al: Classifiers Used

Winnow classifier – We follow the configuration of the Winnow classifier as described in the previous section and employ n-grams ($n = 6$) as features.

LM classifier

- LM – N-gram language models are trained on the whole review articles in the training data. Similar to the configuration of the PA classifier, we vary the value of n ($n = 3, 4, 6$) in language modeling.
- LM-FILTER – This classifier uses the same language model as LM ($n = 6$), but it first tests on individual sentences in a review to filter out those possible objective sentences, *i.e.* the ratio between a sentence's positive and negative probabilities is below a threshold. Af-

PA

=

passive-aggressive

=

flavour of SVM

PA classifier

- Varying the number of n-gram features – Recall that we calculate χ^2 scores as the feature selection metric. We experiment with the top 50k, 100k and 200k n-grams with the highest χ^2 scores as features.
- Varying the n of n-grams – We fix the number of features to 100k while varying the n of n-grams ($n = 1 \ldots 6$). Note that we denote n-grams as all n-grams with orders less than or equal to n.

Results:

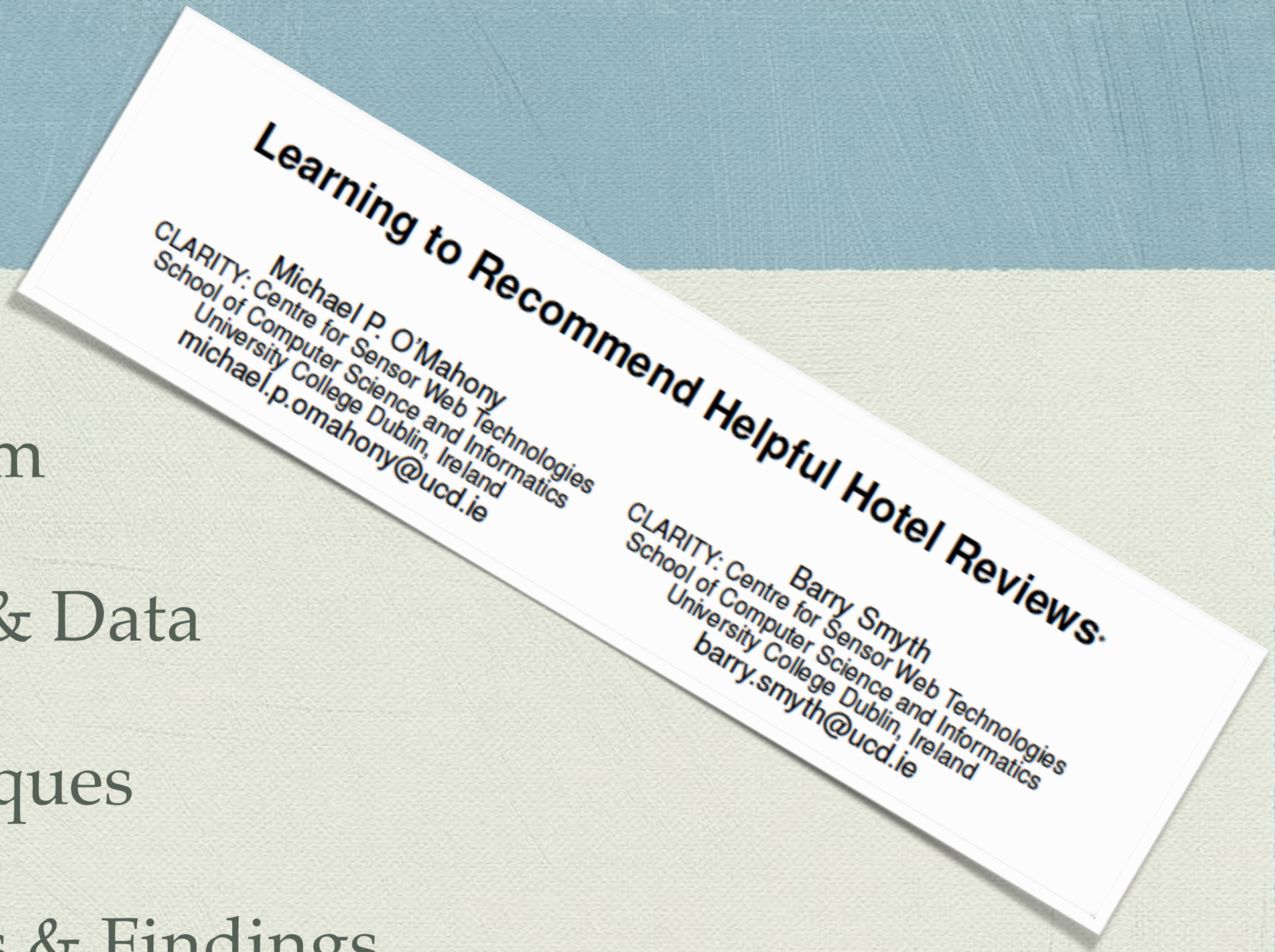
- Higher n-grams do better than uni-/bi- in most models
- Discriminative model (PA) does better than language models; latter confused by mixed sentiments in the same review
- Discriminative model (PA) has lower complexity; perf. does not vary with number of features
- Removal of “objective sentences” (ala Pang & Lee) not needed to make this work

Results: Details

Table 2: Performance comparison P denotes precision and R denotes recall. F_1 is defined as $F_1 = \frac{2PR}{P+R}$. *Pos* stands for positive instances and *Neg* represents negative instances.

	P(Pos)	R(Pos)	F_1(Pos)	P(Neg)	R(Neg)	F_1(Neg)	P	R	F_1
PA 50k ngrams($n = 6$)	0.9329	0.9480	0.9404	0.7437	0.6536	0.6958	0.9013	0.8988	0.9000
PA-200k ngrams($n = 6$)	0.9352	0.9452	0.9402	0.7354	0.6672	0.6996	0.9018	0.8987	0.9003
PA-100k ngrams($n - 6$)	0.9350	0.9460	**0.9405**	0.7386	0.6657	**0.7003**	0.9022	0.8991	**0.9007**
$n = 5$	0.9348	0.9460	0.9404	0.7383	0.6651	0.6998	0.9019	0.8990	0.9004
$n = 4$	0.9350	0.9455	0.9402	0.7367	0.6659	0.6995	0.9018	0.8987	0.9003
$n = 3$	0.9347	0.9460	0.9403	0.7383	0.6640	0.6992	0.9018	0.8988	0.9003
$n = 2$	0.9360	0.9424	0.9392	0.7263	0.6722	0.6982	0.9009	0.8972	0.8990
$n = 1$	0.9176	0.9464	0.9317	0.7112	0.5696	0.6325	0.8830	0.8833	0.8832
LM ($n = 6$)	0.8766	0.9749	0.9231	0.7166	0.3162	0.4388	0.8499	0.8648	0.8573
LM ($n = 4$)	0.8774	0.9739	0.9231	0.7090	0.3186	0.4396	0.8492	0.8643	0.8567
LM ($n = 3$)	0.8384	0.9243	0.8793	0.3516	0.1872	0.2443	0.7570	0.8010	0.7784
LM-FILTER ($n = 6$)	0.8744	0.9631	0.9166	0.6152	0.2988	0.4022	0.8311	0.8520	0.8414
Winnow	0.8688	0.8021	0.8341	0.2867	0.3966	0.3328	0.7715	0.7343	0.7524

Eg2



- Problem
- Setup & Data
- Techniques
- Results & Findings
- Lessons

O'Mahony & Smyth: Problem

- People rate products on-line in product reviews and provide free-text review (explanation); but not all text is "helpful" (biased, poor, unclear)
- TripAdvisor: many 100s/1000s reviews of same hotel; how to sift these appropriately to aid user decisions
- To re-rank on helpfulness; a *review-recommender* rather than *product-recommender;* using **sentiments** of opinions of review
- Use aspects of the reviewer and their reviews (sentiment); to develop classifier for "helpfulness"

OM&S: Set-up & Data

- All TripAdvisor reviews of hotels in Chicago & Las Vegas ('fore 2009); training on reviews with T>5 opinions (helpful >75% pos, unhelpful >75% neg)
- Classifier uses reputation, content, social and sentiment features; which are most predictive of helpfulness measure
- Training Set 50/50 on helpful/unhelpful; using *Weka's* JRip, J42 and Naive Bayes (JRip; propositional rule learner; best on 10-fold cross-validation)
- Las Vegas: 35,802 reviews by 18,849 reviewers of 10,782 hotels

OM&S: Features

- *Reputation*: Mean & SD helpfulness over all reviews by an author (3 features)
- *Content*:terms of text, upper/lowercase, completeness of reviews (6 features)
- *Social*: link structure of user-hotel review graphs (6 features)
- *Sentiment:* from hotel-score and 5-star ratings of hotel aspects (8 features); a *given* text-analysis

Best Features

Table 1: Features ranked by information gain (IG).

Rank	Feature ID	IG
1	R1	0.172
2	ST1	0.095
3	ST3	0.079
4	ST5	0.057
5	R2	0.040
6	Hotel ID	0.031
7	SL4	0.029
8	ST6	0.028
9	C1	0.023

- Four of top-9 were sentiment features
- *ST1*: no. of sub-scores by the user (eg. room, leisure)
- *ST3*: mean of sub-scores assigned by users
- *ST5:* mean of scores over all reviews by user
- *ST6:* SD of scores over all reviews by user

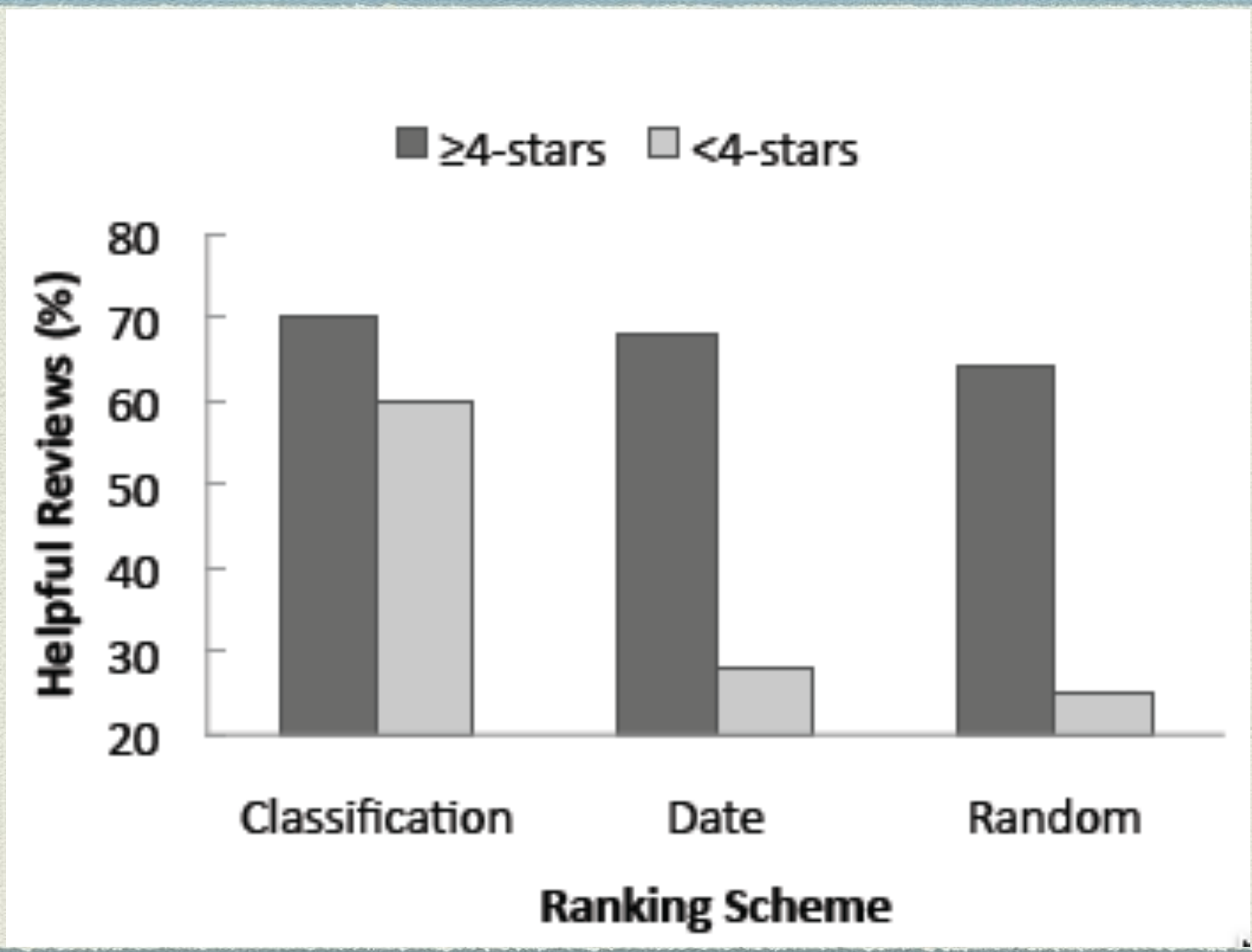
OM&S: Evaluation

- From a test set of reviews on "unseen hotels" for the two cities; selected a helpful review using *classifier*, *date* or *random*

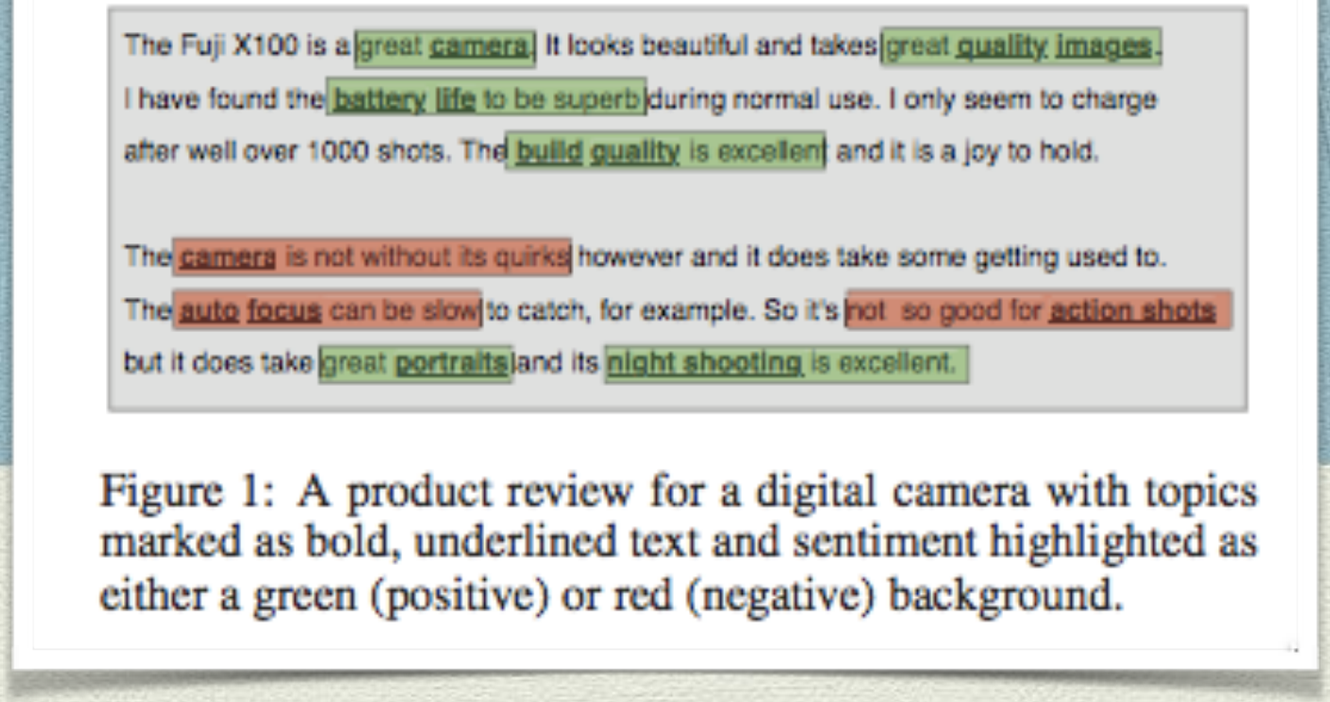
- Check which of these meet the "helpfulness" criterion; i.e. T = 5, >75% pos/neg

- Overall, *classifier* better than *date* and *random* (60% v 28% v 24%); but varies by hotel type

OM&S: Results of Re-Ranking



Points


The Fuji X100 is a great camera. It looks beautiful and takes great quality images. I have found the battery life to be superb during normal use. I only seem to charge after well over 1000 shots. The build quality is excellent and it is a joy to hold.

The camera is not without its quirks however and it does take some getting used to. The auto focus can be slow to catch, for example. So it's not so good for action shots but it does take great portraits and its night shooting is excellent.

Figure 1: A product review for a digital camera with topics marked as bold, underlined text and sentiment highlighted as either a green (positive) or red (negative) background.

- NB: this paper does not explicitly analyse sentiment terms; sentiment is pos/neg on a provided 5-star rating scale

- But, extension of the same helpfulness recommendation using word-sentiment terms in product reviews

Dong, R., Schaal, M., O'Mahony, M. P., & Smyth, B. (2013, August). Topic extraction from online reviews for classification and recommendation. In IJCAI-13 (pp. 1310-1316). AAAI..

Conclusion I: Using Sentiment

- In looking at the use of sentiment we have seen how it can:
  - reflect opinion of a population
  - predict people's behaviour
  - make recommendations
- These are some of most common uses, not exhaustive

Conclusion II: Options

- Note, there are standard ways to do this:
  - Build a classifier from sentiment features than use this to count data items; then use this to predict
  - Correlate sentiment features with some other measure, then do regression
  - Use some method to aggregate the sentiment counts and look at that aggregated measure