

Language-independent Bayesian sentiment mining of Twitter

Alex Davies
University of Cambridge
Cambridgeshire, United Kingdom
ad564@cam.ac.uk

Zoubin Ghahramani
University of Cambridge
Cambridgeshire, United Kingdom
zoubin@eng.cam.ac.uk

ABSTRACT

This paper outlines a new language-independent model for sentiment analysis of short, social-network statuses. We demonstrate this on data from Twitter, modelling happy vs sad sentiment, and show that in some circumstances this outperforms similar Naive Bayes models by more than 10%.

We also propose an extension to allow the modelling of different sentiment distributions in different geographic regions, while incorporating information from neighbouring regions.

We outline the considerations when creating a system analysing Twitter data and present a scalable system of data acquisition and prediction that can monitor the sentiment of tweets in real time.

Categories and Subject Descriptors

I.2.7 [Artificial Intelligence]: Natural language processing—*Language models, Text analysis*

General Terms

Theory

Keywords

Twitter, Sentiment analysis, Topic modelling, Geo mining

1. INTRODUCTION

People are increasingly posting public content on the internet that reveals their sentiments and opinions. By far the largest source of this content, in terms of numbers of users, is Twitter. Twitter, at the time of writing, has 200 million users, who generate over 65 million tweets per day. There is a vast amount of sentiment information contained in this data, but we need suitable techniques to effectively extract it.

Understanding the sentiment or opinions of people is a valuable resource. It is useful on both a macro scale, where

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

The 5th SNA-KDD Workshop '11 (SNA-KDD'11), August 21, 2011, San Diego CA USA . Copyright 2011 ACM 978-1-4503-0225-8...\$5.00.

we can evaluate aggregated sentiment to predict macro-scale human behaviour, and a micro scale, where individual's sentiments provide actionable information to personalize services and target particular users.

At the macro-scale, there has been work predicting a wide variety of trends based on sentiment information retrieved from Twitter. This has been in fields as varied as politics, marketing and finance. In [13], it was found that correlations as high as 80% could be found between political sentiment data on Twitter and traditional polling methods. Recent work has shown that Twitter sentiments are very strong predictors of movie box office performance [1] and even the closing value of the Dow Jones Index [5].

A useful feature of Twitter is that a large amount of its tweets are accompanied by geodata; information about where in the world the tweet was generated. This facilitates the creation of systems that can model differences in social network usage between different geographic areas. This data has been used to model language variation across the United States [8], as well as sentiment variation across the US by state and time of day.

2. TWITTER DATA, GEO MODELLING AND SENTIMENT ANALYSIS

Previously, sentiment analysis in Twitter has largely been concerned with English tweets. This has allowed the use of domain knowledge about English in models to substantially increase the effectiveness of sentiment prediction models. There are large lists of manually curated English words and their associated sentiments such as affective word lists ANEW [6]. Some methods also employ features obtained from Part-of-Speech taggers, though there are conflicting results as to their utility [16][2]. Both of these methods do not extend to other languages.

We approach the problem of modelling sentiment, while making no assumption on language. We propose a probabilistic model that learns a word distribution for each sentiment based on a set of key indicator words for each sentiment. To be language independent, throughout our examples we use emoticons as the key indicating words. Previous works have used emoticons as noisy labels for training sentiment classifiers. On this data, Naive Bayes tends to outperform other more sophisticated techniques, such as SVMs and CRFs[15], so we will use it as a baseline for comparison.

Secondly, we propose a simple method for modelling different sentiment distributions in different geographic regions, while incorporating information from neighbouring regions to improve estimates in areas of low tweet density.

3. SENTIMENT PREDICTION

Our model for sentiment prediction is as follows:

A set of tweets, T , are generated from a multinomial mixture model, where the hidden mixture component s is the sentiment and the multinomial $\vec{\theta}_s$ is the word distribution for sentiment s . $\vec{\theta}_s$ is a vector of length $|V|$, that sums to 1, where each element $\theta_{s,i}$ is the probability that word w_i is drawn on a given draw from sentiment s .

Our prior belief over each $\vec{\theta}_s$ is an asymmetric Dirichlet distribution $Dir(\vec{\alpha}_s)$. $\vec{\alpha}_s$ is based on a set of keywords for sentiment s , F_s , and will enforce that $\vec{\theta}_s$ is conceptually about sentiment s .

3.1 Likelihood

Our data consists of $|T|$ tweets $t_i = (w_{i,1}, \dots, w_{i,n_i})$ where n_i is the number of words in tweet i . Each word is a member of the set V , which is the vocabulary of words we consider in our model. For each tweet t_i the probability of the tweet given a set of word distributions, one for each sentiment s , is

$$p(t|\vec{\theta}) = \sum_{s \in S} p(t|\vec{\theta}_s)p(s) \quad (1)$$

$$= \sum_{s \in S} \prod_{w \in t} p(w|\vec{\theta}_s)p(s) \quad (2)$$

$$= \sum_{s \in S} p(s) \prod_{w \in t} p(w|\vec{\theta}_s) \quad (3)$$

The multinomial mixture-model is a good language model for tweets, as due to their short size (max 140 characters) they very rarely exhibit multiple sentiments. This is the justification for choosing this model over a more expressive model such as Latent Dirichlet Allocation[4].

3.2 Prior

We would like our word distribution to reflect our chosen sentiments, rather than for arbitrary components that separate the data well. To do this we use an asymmetric dirichlet prior $Dir(\vec{\alpha}_s)$. By setting a high value of $\vec{\alpha}_{happy,i}$, where i is the index for the word ':)', we encode the belief that happy smilies are likely to occur in happy tweets. Similarly, by setting a low value of $\vec{\alpha}_{sad,i}$, we encode the belief that happy smilies are unlikely to occur in sad tweets.

Formally, we set $\vec{\alpha}_s = k\vec{1} + \vec{\delta}_s$. All vectors are of length $|V|$, the size of the vocabulary. $\vec{1}$ is a vector with all values equal to 1, k is a scalar constant and $\vec{\delta}_s$ is defined as follows:

$$\delta_{s,w} = \begin{cases} \delta_+ & \text{if } w \in F_s \\ \delta_- & \text{if } w \in F_{s'}, \text{ where } s' \neq s \\ 0 & \text{Otherwise} \end{cases} \quad (4)$$

δ_- and δ_+ are tunable parameters that are the weights added to the prior on the values of the indicative words. $\delta_+ > 0$ adds mass to the indicative words for a sentiment, reflecting our belief that they are more likely, while $-\delta_- < 0$ removes mass from indicative words for other sentiments. The full form of the prior is thus:

$$p(\vec{\theta}_s) = Dir(k\vec{1} + \vec{\delta}_s) \quad (5)$$

Choice of δ_+

It is important to take some care in selecting the value of δ_+ . The value of δ_+ shouldn't be set too high, as if the values of $\alpha_{s,i}$ for F_s are too high relative to the other values of $\vec{\alpha}_s$, our model would expect there to be multiple elements of F_s in a given tweet of sentiment s . It would then assign lower probabilities to tweets that have too few indicating words.

As a rule of thumb, it is best to set $\delta_+ < \frac{k|V|}{n|T_s|} \forall T_s$, where n is the average number of words in a tweet.

3.3 Learning

Because of the latent variables s , exact inference in this model is not possible. However, we can perform Variational Bayes[9] to obtain an approximation to the posterior distribution $P(\vec{\theta}_s|\vec{\alpha}_s, T)$. Variational Bayes is a common method for efficiently finding an approximate distribution over model parameters in intractable Bayesian models. It finds this approximation by first assuming the distribution factorizes into a product of simpler distributions and iteratively optimizes each of the simpler distributions such that their product is closer to the true distribution we are trying to approximate.

The update equations for a Dirichlet-Multinomial model using the mean-field approximation are:

$$n_{s,w}^{k+1} = \langle n_{s,w}^k \rangle_{\vec{\theta}^k} \quad (6)$$

$$\theta_{s,w}^{k+1} = \frac{e^{\Psi(\alpha_{s,w} + n_{s,w}^{k+1})}}{e^{\Psi(\sum_{w' \in V} \alpha_{s,w'} + n_{s,w'}^{k+1})}} \quad (7)$$

Where $n_{s,w}^t$ is the number of times a word w appears in tweets with sentiment s at iteration k and Ψ is the digamma function $\Psi(x) = \frac{\Gamma'(x)}{\Gamma(x)}$. For the algorithm we also need a variable $\vec{\phi}^k$, where $\vec{\phi}_t^k$ is the sentiment of tweet t at iteration k .

The algorithm is executed as follows:

1. Both $\vec{\theta}^0$ and $\vec{\phi}^0$ are initialized randomly.
2. Update our estimates of the sentiment of each tweet $\vec{\phi}_t^{k+1}$, based on the current word distribution $\vec{\theta}^k$.
3. Count the number of times each word w occurs in a tweet with sentiment s and assign this value to $n_{s,w}^{k+1}$ (Equation 6).

4. Update our estimate of the word distributions $\tilde{\theta}^{k+1}$, based on our new values of \tilde{n}^{k+1} (Equation 7).
5. Repeat steps 2-4 until convergence.

At completion, our posterior belief distribution over θ_s is $\text{Dir}(\tilde{\alpha}'_s)$ where $\tilde{\alpha}'_s = \tilde{\alpha}_s + \tilde{n}_s$.

3.4 Prediction

Once we have trained the model, the log probability of a tweet being generated given a particular sentiment can be computed as:

$$\log(p(t|\tilde{\alpha}'_s)) = \sum_{w \in V} \sum_{i=0}^{n_{t,w}} \log(i + \alpha'_{s,w}) - \sum_{i=0}^{|t|} \log\left(i + \sum_{j=1}^{|V|} \alpha'_{s,j}\right) \quad (8)$$

Where $n_{t,w}$ is the number of occurrences of w in t . This is the log probability of an observation under a Dirichlet-Multinomial model. Using Bayes' rule and an empirical estimate of $p(s)$ this can be used to compute $p(s|t)$, which is our sentiment prediction.

$$p(s|t) = \frac{e^{\log(p(t|\tilde{\alpha}'_s))}}{\sum_{s' \in S} e^{\log(p(t|\tilde{\alpha}'_{s'}))}} p(s) \quad (9)$$

Importantly, this can be calculated efficiently and is suitable for prediction in an online setting.

4. MODELLING GEOGRAPHIC REGIONS

If we now want to model separate sentiment distributions for different geographic regions, a first option would be to merely consider tweets from each region independently. However, by doing this we are disregarding any potential information we can learn about regions' word distributions from neighbouring regions. Especially in the cases where a region has very few tweets, it makes sense to use data from neighbouring regions to improve our estimates of the word distributions. We propose a simple, tractable extension to separately model word distributions while maintaining comparable classification accuracy when data is sparse.

We consider a hierarchy of regions R . We have a root node, r_0 , which contains all tweets. Below this, the child nodes of r_0 partition the tweets into disjoint subsets, with one belonging to each child node. These new sub-regions can in turn be subdivided into smaller subregions. For example, one possible hierarchy is the geo-political regions of the world. In this case, the root node is "The World". This has one child for each country of the world. Each country then has a child for each of its states/provinces. We define $N(r)$ as the neighbours of region r (those that share the same parent region), $P(r)$ as the parent of region r and T_r as the tweets contained in region r .

We train our original model independently for each region to obtain a distribution over words, $p(\tilde{\theta}_{s,r}|T_r) \forall r$. This

means we have separate word distributions for each region, parametrized by $\tilde{\alpha}_{s,r}$. We now want to define a prior $p(\tilde{\theta}_{s,r}|T_{-r})$, which is our belief distribution over $\tilde{\theta}_s$ in region r , given tweets observed in other regions.

We design a prior that encodes a broad assumption that there will be similarities between close regions, that each region should have a limited effect on every other region and which is very simple to compute. We define this prior as follows:

$$P(\tilde{\theta}_{s,r}|T_{-r}) = \text{Dir}(\tilde{\beta}_{s,r}) \quad (10)$$

$$\tilde{\beta}_{s,r} = \omega_0 \frac{1}{|N(r)|} \sum_{r' \in N(r)} \frac{f(\sum_{i=1}^{|V|} \alpha'_{s,r',i})}{\sum_{i=1}^{|V|} \alpha'_{s,r',i}} \tilde{\alpha}'_{s,r'} + (1 - \omega_0) \tilde{\beta}_{s,P(r)} \quad (11)$$

$$f(x) = A \left(1 - e^{-x(\log(A) - (\log(A-1)))}\right) \quad (12)$$

$f(x)$ is a thresholding function which is monotonically increasing and asymptotes at A . This limits how much a given region can influence the prior of the region we are considering. ω_0 is a tunable parameter that indicates how much more influence closer regions have than ones further away in the hierarchy.

Now we can train a new model that is the same as the model that considered the regions independently, only now we replace the priors with those $p(\tilde{\theta}_{s,r}|T_{-r})$ that we just calculated.

5. DATA PIPELINE

Data acquisition is a non-trivial step, involving filtering of users, filtering of tweets, parsing of tweets and determining the region of tweets.

5.1 Obtaining data

Twitter provides two APIs to access information about tweets; the Search API and the Streaming API. When selecting an API to use, there are several important characteristics to consider.

APIs

The Search API is intended to provide the ability to perform specific, low throughput queries. One can refine by user, content, geographic location (defined by a GPS co-ordinate and radius) and date. Importantly, only tweets from the preceding 5 days can be searched and queries are limited to approximately 10 per minute at the time of writing.

The Streaming API is designed to allow access to a live stream of tweets, as they occur. One can refine by user, content and area (defined by a rectangular bounding box). However, the user/content filter and the geographic filter are performed with a logical "OR". This means that a search for

tweets “Charlie” in San Francisco will return all tweets from San Francisco and all tweets containing “Charlie”. We are interested in predicting on the streaming API.

To collect a dataset of substantial size quickly we use the Streaming API, though the Search API is also used to compile testing datasets.

Data format

When querying either API, we are returned a list of status updates. Each status contains fields describing the tweet. The fields we are concerned with are shown in Table 1.

Some other information is returned, mainly relating to whether a tweet is in response to another tweet or user. Through another API call, more information can be requested about a particular user. However, the search API’s rate limit is too low to query every user we see in the stream. Therefore for prediction on the stream, we cannot rely on having access to this extra data.

5.2 Filtering Tweets

Spam accounts

While there are millions of legitimate users on Twitter, there are also a large number of spam accounts, corporate accounts and bots (most notably weather bots), that have a very negative effect on standard language models. This is due both to the volume of tweets they generate, which is generally much higher than legitimate users, and the fact that many of the tweets are automatically generated or templated. This leads to many words artificially appearing together many times, which breaks the assumptions of language models based on co-occurrences such as ours.

Identification of spam accounts is a separate research question in itself [7][17][18]. However, as noted before, in the online setting we are severely limited in the number of features we have for a given user. We don’t have access to their full feed or social graph. The only informative features are the number of followers that a user has, as well as the number that they are following. Research has shown that users with more than 1000 followers or who follow more than 1000 users have a drastically different usage pattern to normal users [11]. This may indicate that the account is spam, a celebrity, a corporation or other accounts we are not interested in. Therefore as a simple rule we filter out all these users.

Duplicate tweets

An important aspect of Twitter that breaks the assumption of most language models are retweets. Retweets are when a user rebroadcasts a tweet that was tweeted by another user. This is usually of the form: “RT @[Original User] [Original Text]”. While the fact that a user has retweeted another tweet is likely to contain meaningful sentiment information, the fact that the words appear verbatim strongly breaks the iid assumption of common language models. While retweeting is an important part of Twitter culture, we choose simply to remove the retweets at sampling time.

5.3 Tweet Processing

Once we have retrieved our tweets from the API, we convert them from the string representation to a feature representation. For our representation we choose a bag-of-words model.

Tokenization

Tweets are not written like other text documents. The 140 character limit and informal setting result in tweets that heavily use slang, emoticons, spelling and punctuation that would not be found in traditional text documents. Because of this, we need to perform custom pre-processing and tokenization of the tweets. We use a tokenizer specifically designed for Twitter [14]. This much better captures the use of punctuation, emoticons and words on Twitter. Additional steps were to conflate more than three consecutive occurrences of the same letter down to two letters. There are very few meaningful terms that contain more than three of the same character in a row, but it is very common on Twitter to repeat letters for emphasis.

5.4 Determining location in geo-hierarchy

As already mentioned in the previous section, the geographic information may come as either the GPS co-ordinates that the tweets was tweeted from, or the name of a location with a bounding polygon for that region. When creating a system to model regional variation, we would like to use the geopolitical region of the tweet. Fortunately, there is an API provided by GeoNames.org that converts GPS co-ordinates to a code defined in ISO 3166-2. ISO 3166-2 are two region codes separated by a dash that are the country code and a sub region (state/province) code.

This service is also rate-limited such that we cannot query at the rate of the streaming API. However, if we receive a tweet’s location information as a name and bounding region pair, we can calculate a representative GPS co-ordinate within the region, query the GeoNames API and cache the result against the location name. As a representative point we use the centroid. While this is not assured to be inside the bounding polygon, as the polygons can be non-convex, in practice this works for almost all points.

6. RESULTS

To evaluate our model we firstly compare the performance of our model without geo-information against Naive Bayes and then test the effect of including geo-information in our model. In our comparison against Naive Bayes we test using a sample of 50,000 tweets with a happy or sad emoticon. 60% of these contain a happy emoticon, while 40% contain a sad emoticon. No tweet contains both a happy and sad emoticon. In our geo model comparison, we test on a different sample of 20,000 tweets with happy or sad emoticon that also include geo information. For illustrative purposes, we also include an example list of high probability words for different regions and sentiments (Table 2) and maps (Figures 3,4) showing relative rates of happy and sad tweets for different regions of the world.

6.1 Comparison with Naive Bayes

Firstly, we test the accuracy of the proposed tweet model without geographic information. As a baseline comparison we use a Naive Bayes classifier trained on the same features.

text	The body text of the tweet
id	The unique id of the tweet
author	The author of the tweet
retweeted	A boolean variable indicating if this is a retweet
coordinates	Co-ordinates the tweet originated from (if available)
followers	The number of followers the user has
following	The number of users that the user follows
place	Geo information, (place name, bounding box)
created at	Timestamp the tweet was created at

Table 1: Fields returned in Twitter API

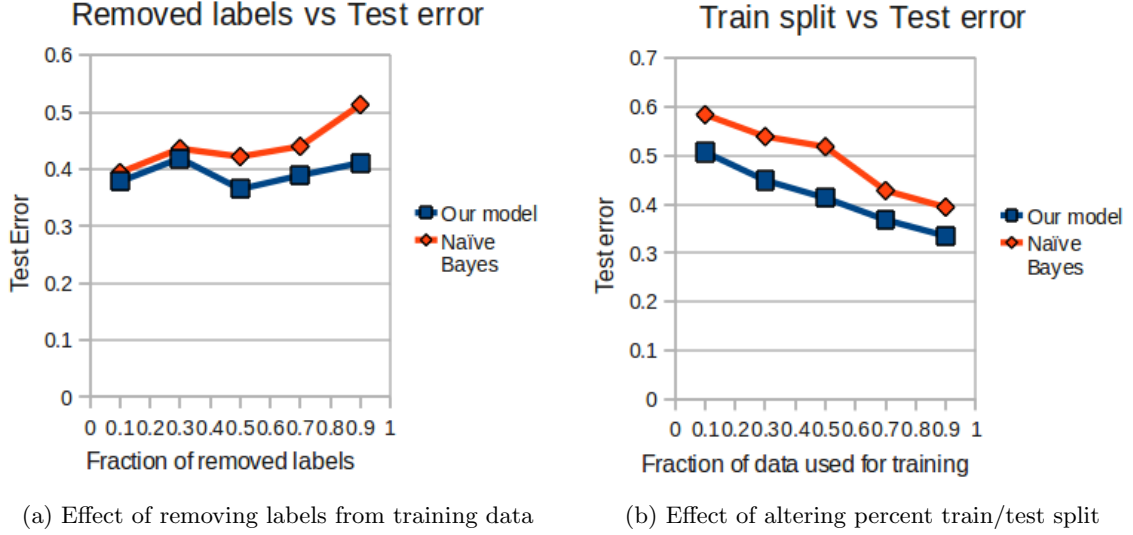


Figure 1: Comparison of our model with Naive Bayes

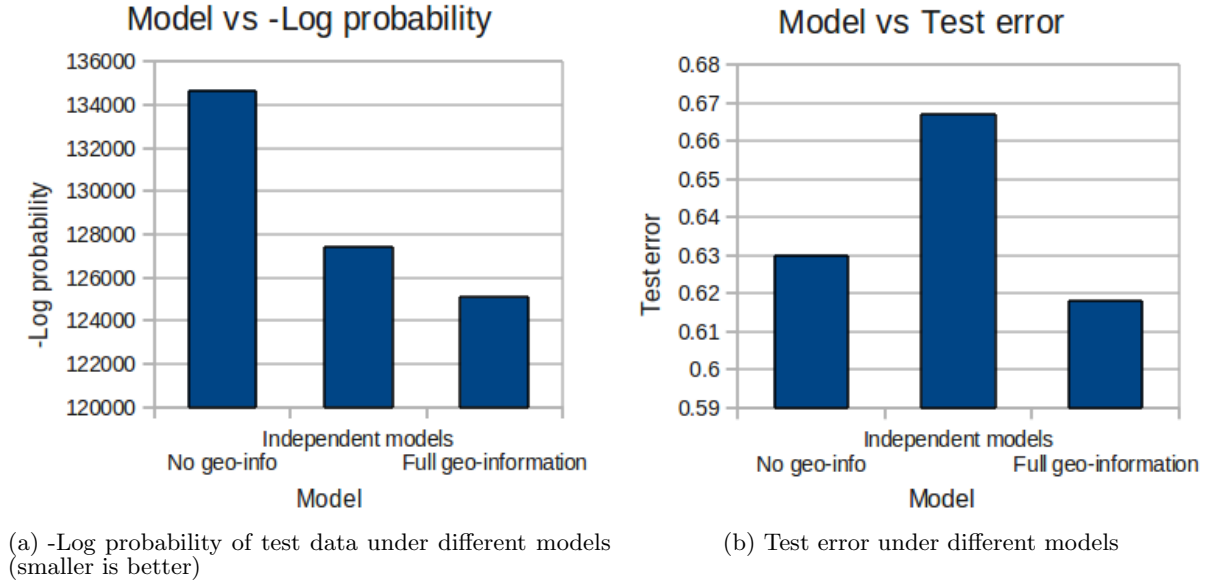


Figure 2: Comparison of our model without geo information, with independent partitioning and using neighbouring geo-information

Region	Sentiment	Top words
World	Happy	amore amour liebe happy :-) kasih amores bahagia love feliz :) makasih follback seguindo oooi good follow lovely xxx selamat hey s/o terima cheers thx
	Sad	sedih :(sad triste :(dor saudade droga fome cade doendo aaa saudades garganta mimimi aff morreu odeio gripe perdi merda aaah :'(
UK	Happy	amour feliz kasih :-) happy love :) thanks birthday thank follow good welcome luck great nice morning amazing hello lovely xxx best hey awesome cheers
	Sad	:(sad triste :(nooo booo </3 poor miss *cries* gutted afford :'(poorly hate dean ugh fml urghhh stressed headache *sigh* canada prayforkatie richards rip whyyy horrible revision *hugs*

Table 2: List of highest probability words for each sentiment at different regions

For the first test we take 50% of the data for training and from a fraction of these tweets, remove the emoticons. These represent emotive tweets that we would encounter on Twitter that do not have an emoticon. The models are trained on this data and then the classification accuracy is assessed on the remaining 50% of the data, with all emoticons removed. For the second test we remove emoticons from 90% of the training data, but alter the fraction of data used for training.

In Figure 1 we see that as less labelled training examples are supplied, our model further outperforms Naive Bayes. At very low label density this is an improvement in classification accuracy of over 10%. We see that while increased test data results in increased model performance in both cases, there is little difference between our model and Naive Bayes.

6.2 Effect of incorporating geo-information

Secondly, we look at the effect of incorporating geo-information into our model. The three models we test are our model without geo-information, independent instances of our model for each region and incorporating neighbouring region information as outlined in Section 4. We compare them based on classification error and the negative log probability of the test dataset under the model. Both of these are averaged over 10 different test splits. The negative log probability gives us an idea of how well the data is modelled, as a lower value of this means that the model was less “surprised” by the data.

We can see in Figure 2 that by training independent models for different regions, we lose classification accuracy, even though we are modelling the data better, as shown by the lower negative log probability. However, by incorporating neighbouring region information, we recover our classification accuracy and achieve an even further reduction in negative log probability.

7. CONCLUSIONS

In conclusion, we propose a probabilistic model for sentiment classification of short social network statuses. We show that this outperforms Naive Bayes on simple word features when there are additional unlabelled training examples. We also propose a simple extension for modelling distributions across multiple geographic regions, without sacrificing classification accuracy. Finally, using this extension, we have described an implemented a data pipeline and efficient algorithm for performing online geographic sentiment prediction of Twitter statuses.

8. REFERENCES

- [1] S. Asur and B. Huberman. Predicting the future with social media. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pages 492–499. IEEE, 2010.
- [2] L. Barbosa and J. Feng. Robust sentiment detection on Twitter from biased and noisy data. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 36–44. Association for Computational Linguistics, August 2010.
- [3] M. Beal and Z. Ghahramani. The variational Bayesian EM algorithm for incomplete data: with application to scoring graphical model structures. *Bayesian statistics 7*, 2003.
- [4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3(4-5):993–1022, May 2003.
- [5] J. Bollen, H. Mao, and X.-j. Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, pages 1–8, 2011.
- [6] M. M. Bradley and P. J. Lang. Affective norms for English words (ANEW): Stimuli, instruction manual and affective ratings. Technical report, The Center for Research in Psychophysiology, University of Florida.
- [7] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia. Who is tweeting on Twitter: human, bot, or cyborg? In *Proceedings of the 26th Annual Computer Security Applications Conference*, pages 21–30. ACM, 2010.
- [8] J. Eisenstein, B. O’Connor, N. Smith, and E. Xing. A latent variable model for geographic lexical variation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1277–1287. Association for Computational Linguistics, 2010.
- [9] Z. Ghahramani and M. Beal. Variational inference for Bayesian mixtures of factor analysers. *Advances in neural information processing systems*, 12:449–455, 2000.
- [10] A. Go, R. Bhayani, and L. Huang. Twitter Sentiment Classification using Distant Supervision. Technical report, 2009.
- [11] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a Social Network or a News Media? In *WWW ’10: Proceedings of the 19th international conference on World wide web*, pages 591–600, 2010.
- [12] T. Lake. Twitter Sentiment Analysis. Technical report, Western Michigan University, Apr. 2011.
- [13] B. O’Connor, R. Balasubramanyan, B. Routledge, and

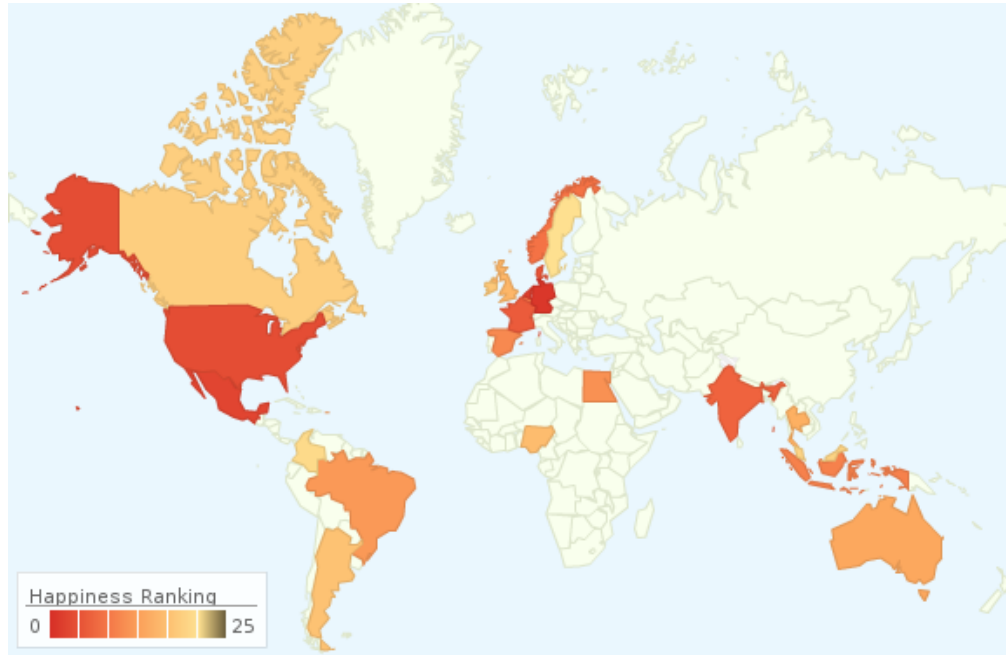


Figure 3: Map of the world with countries ordered by relative rate of happy to sad tweets

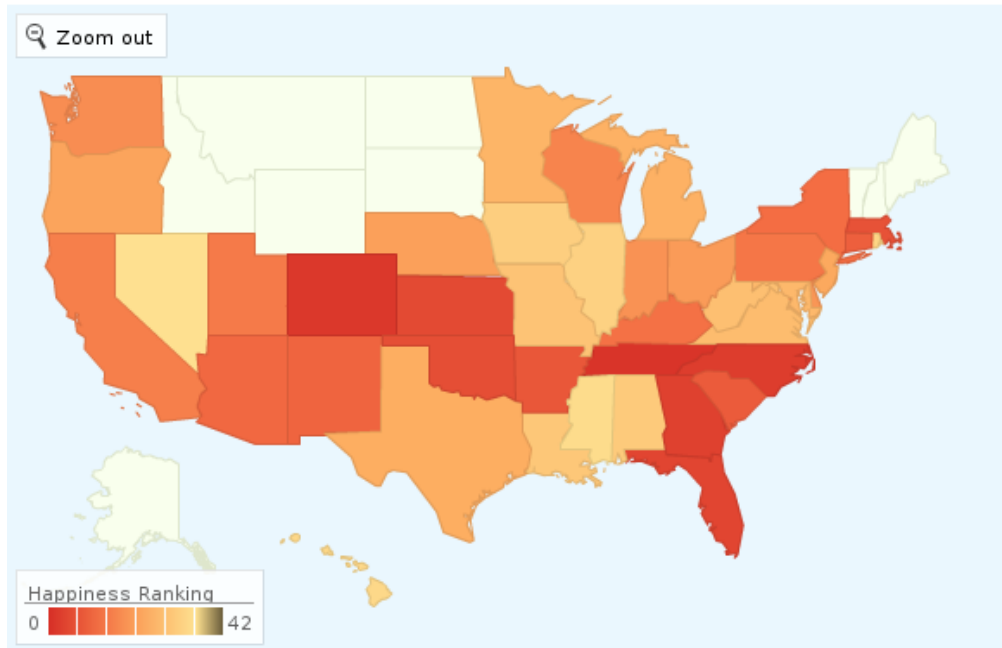


Figure 4: Map of the US with countries ordered by relative rate of happy to sad tweets

- N. A. Smith. From tweets to polls: Linking text sentiment to public opinion time series. In *Fourth International AAAI Conference on Weblogs and Social Media*, May 2010.
- [14] B. O'Connor, M. Krieger, and D. Ahn. TweetMotif: Exploratory search and topic summarization for twitter. *Proceedings of ICWSM*, pages 2–3, May 2010.
 - [15] A. Pak and P. Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. *Proceedings of LREC 2010*, pages 1320–1326, 2010.
 - [16] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.
 - [17] G. Stringhini, C. Kruegel, and G. Vigna. Detecting spammers on social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference*, pages 1–9. ACM, 2010.
 - [18] A. Wang. Don't follow me: Spam detection in Twitter. In *Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on*, pages 1–10. IEEE, 2010.
 - [19] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing Contextual Polarity: An Exploration of Features for Phrase-Level Sentiment Analysis. *Computational Linguistics*, 35(3):399–433, Sept. 2009.