Brand-related Events Detection, Classification and Summarization on Twitter

Eric Medvet, Alberto Bartoli
DI^3 - Industrial and Information Engineering Dept.
University of Trieste, Italy
Email: emedvet@units.it, bartoli.alberto@units.it

***Abstract*—The huge and ever increasing amount of text generated by Twitter users everyday embeds a wealth of information, in particular, about themes that become suddenly relevant to many users as well as about the sentiment polarity that users tend to associate with these themes. In this paper, we exploit both these opportunities and propose a method for: (i) *detecting* novel popular themes, i.e. *events*, (ii) *summarizing* these events by means of a concise yet meaningful representation, and (iii) assessing the prevalent *sentiment polarity* associated with each event, i.e., positive vs. negative.**

Our method is fully unsupervised and requires only a precompiled *topic* description in the form of set of potentially relevant keywords that might appear in the events of interest.

We validate our proposal on a real corpus of about 8,000,000 tweets, by detecting, classifying and summarizing events related to three wide topics associated with tech-related brands.

***Keywords*-event detection; sentiment analysis; summarization**

I. Introduction

Microblogging platforms produce a huge and ever increasing amount of user-generated content. Millions of Twitter users, for example, output hundreds of millions *tweets* every day—i.e., texts up to 140 characters which may contains tags, URLs, and mentions to other users. The purposes of this wealth of short texts include chatter, sharing of first- or second-hand news, opinions about things, facts or persons and so on. This variety of content types suggests, on one hand, to use Twitter as a source of breaking news or events; on the other hand, Twitter can indicate promptly the general mood or even a sense of sentiment about a specific topic. In this paper we propose a method for merging these two opportunities.

Our method is fully unsupervised and requires only a precompiled *topic* description in the form of set of potentially relevant keywords that might appear in the events of interest. We focus on topics represented by keywords related to three important tech-related brands and show how to: (i) discover the current popular brand-related events and assess qualitatively their popularity, (ii) infer the users' sentiment about the events and (iii) extract those tweets which best summarize this sentiment.

II. Related work

Two research topics are particularly relevant to our work: sentiment analysis and event detection.

A number of works has appeared concerning Twitter sentiment analysis. In many cases, the analysis aims to support high-level socio-economical studies. In [1], the authors find that public mood expressed in Twitter is correlated and weakly predictive of the Dow Jones Industrial Average index. A similar analysis is also carried out in [2]: the authors discover a weak correlation between the properties of an interaction graph induced from brand-related tweets and the traded volumes of corresponding stocks; indeed, the authors also state that they were unable to find any graph property significantly correlated with stock prices. Consumer confidence and political opinion are analyzed in [3] and correlated to the sentiment ratio for some statically chosen topics: the study shows that the latter captures the broad trends in the survey data. A similar analysis is performed also in [4].

Concerning the technique itself used to infer the sentiment of tweets, several approaches have been proposed. The authors of [5] propose to use Twitter specific features, such as emoticons and abbreviations, to improve sentiment classification effectiveness and compare them to more consolidated NLP features such as Part-of-Speech (POS) tagging: they conclude that Twitter specific features may be somewhat useful. POS tagging together with other high-level features are used in [6] to classify tweets sentiment using a tree-kernel. In our work, we borrow several ideas related to text preprocessing from the latter two approaches and apply them in context much broader than mere sentiment analysis. The problem of obtaining a useful, yet noisy, source for sentiment labels for tweets is considered in [7], where the authors improve POS features effectiveness by applying a two stage classification: first they distinguish between subjective and not subjective tweets; then they try to infer the sentiment polarity only for subjective tweets.

Many works have dealt recently also with event detection in Twitter. The problem of detecting and summarizing events in the context of football matches is considered in [8]. In our work we consider this problem from a broader point of view, in terms of general topics. Set of words, as in our work, rather than a single bursty keyword, are used in [9] and [10] to detect emerging topics expressed by the community. The clustering of wavelet-based signals of words is instead the technique proposed in [11], which is validated on netizens' online discussion about Singapore General Election 2011.

978-0-7695-4880-7/12 $26.00 © 2012 IEEE
DOI 10.1109/WI-IAT.2012.36

The authors of [12] focus on performance of their event detection method which applies to a stream of Twitter posts: they present an algorithm based on locality-sensitive hashing and show that it is able to provide performance similar to other state-of-the-art systems for first story detection while achieving significantly shorter processing times.

A system which is very close to our proposed method is presented in [13]: a web application monitors Twitter for a given keyword and presents the user a navigable timeline of related events, together with a rough estimate of the general sentiment ratio about each single event and a couple of summarizing tweets. The authors focus on the computer-human interaction with the system and investigate about its use as a tool for journalists.

Finally, the huge amount of user-generated content available in Twitter recently suggested its usage to allow an effective dialogue between citizens and governments. In this scenario, the authors of [14] show how Latent Semantic Analysis (LSA) can help processing large amounts of small, unstructured texts (such as tweets) in order to provide meaningful summaries of emerging themes.

III. Our approach

We aim at detecting and characterizing popular *events* related to a given general *topic*. A topic is a precompiled and statically defined set of keywords. In this work we focus on brand-related topics. For example, for the topic Apple we precompiled the following set of keywords: `apple`, `iphone`, `ipad`, `ipod`, `mac`, `macintosh`, `macbook`, `ios`.

An event is a theme of conversation that becomes suddenly popular amongst tweets of the same topic. In addition to detecting events, we want to *summarize* these events, i.e., provide a synthetic description of them, as well as provide a qualitative assessment of popularity and sentiment polarity (i.e., positive, neutral or negative) of the event.

We say that a tweet output by the Twitter stream *refers* to a certain topic, if the tweet contains at least one of the keywords for that topic (a tweet may refer to zero or more topics).

We propose an approach which consists of four steps: (i) select the recent tweets which refer the given topic; (ii) identify the event keywords—i.e., a set of words which exhibit a higher than normal usage—and select the related tweets; (iii) classify those tweets in one on the three sentiment classes (positive, neutral or negative); (iv) for each sentiment class, select few representative tweets to summarize the event. The next sections describe more in detail the last three steps. The first step can be achieved easily achieved using existing tools, e.g., the Twitter Streaming API.

A. Preprocessing

Before performing the next steps, we preprocess all the tweets as follows.

Let T be the set of all *recent* tweets t related to the given topic, i.e., the tweets which were output by the stream in the last 14 days. For each t, we: (i) convert to lowercase; (ii) replace each URL with the token `T_URL`; (iii) replace each positive emoticon with the token `T_POS_EMOT` and each negative emoticon with the token `T_NEG_EMOT`; (iv) replace all numbers with the token `T_NUM`; (v) expand acronyms and abbreviations—to this end, we previously compiled a list of 236 popular Internet and Twitter slang acronyms; (vi) truncate each in-word sequence of three or more repeated characters to three (e.g., `cooooool` becomes `coool`); (vii) remove all non-word characters; (viii) remove common English stop words; (ix) perform a stemming. We choose to include emoticons because previous studies show that such features can boost the performance of sentiment analysis on microblogging text corpus [6], [5].

Finally, we transform each t in a feature vector obtained by using unigrams, i.e., we count for each word its number of occurrences. We consider only the 2000 words which occur most in the corpus T, hence obtaining 2001 features, where the last one is the number of occurrences of all other less frequent words. We call the set W_M of these 2000 words the *monitored words* set.

B. Event keywords identification

We call *event keywords* a set $W^* \subset W_M$ of 3 words which occur with an unusually high rate in the recent tweets: we assume that W^* words are closely related to an event.

As described in full detail below, in order to construct W^*, we proceed as follows: (i) we first identify those words which exhibit an unusually high frequency; (ii) we then use each one of such words as a seed to form a word set which includes related words (i.e., synonyms) and possibly excludes noisy and frequent words; (iii) we finally check that the set itself exhibits an unusually high frequency. The rationale of considering sets of words instead of single words is two-fold.

On one hand, we want to broaden the event coverage. As stated below, we require just two on three words to occur in a tweet in order to consider it as related to the event. Hence, we can also detect those events which are associated with words which do not occur singularly with an unusually high rate: e.g., an event to which users refer using synonyms.

On the other hand, we aim at not considering recurring words as real events. For example, Twitter users often use the `#musicmonday` and `#ff` hashtags, respectively on Monday and on Friday; these words hence periodically occur with a higher rate, yet they do not represent real events. By requiring that at least one other word occurs together with the recurring word with an unusually high rate, we are possibly filtering out trivial recurring words usage.

The details of the sketched procedure follow.

Let $w \in W_M$ be a monitored word, $N_{w,i}$ the number of T tweets containing w output within the i-th past day (i.e.,

$N_{w,0}$ counts the number of T tweets containing w during the last 24 hours) and N_i the number of all tweets of T output within the i-th past day. For each w, (i) we compute the current frequency f_w^{now} as the relative frequency of w in the last 3 days:

$$f_w^{\text{now}} = \frac{\sum_{i=0}^{2} N_{w,i}}{\sum_{i=0}^{2} N_i}, \tag{1}$$

(ii) we compute the historical frequency f_w^{hist} as the relative frequency of w in the 13 days before the last day:

$$f_w^{\text{hist}} = \frac{\sum_{i=1}^{13} N_{w,i}}{\sum_{i=1}^{13} N_i} \tag{2}$$

(iii) we compute the *burstiness index*:

$$b_w = f_w^{\text{now}} - 3 f_w^{\text{hist}} \tag{3}$$

We form a sorted set $W_P \subset W_M$ by excluding those W_M words such that $f_w^{\text{now}} < 0.001$ or $b_w < 0$, and sorting the remaining ones by their decreasing burstiness index.

Then, we perform the following iterative procedure over W_P words. Let v be the first word in W_P, for each word $w \neq v$ in W_M, (i) we compute the current co-frequency $f_{v \wedge w}^{\text{now}}$ as:

$$f_{v \wedge w}^{\text{now}} = \frac{\sum_{i=0}^{2} N_{v \wedge w,i}}{\sum_{i=0}^{2} N_{v,i}} \tag{4}$$

where $N_{v \wedge w,i}$ is the number of T tweets containing both the word v and w output within the i-th past day; (ii) we compute the *co-burstiness index* $b_{v \wedge w} = f_{v \wedge w}^{\text{now}} - f_w^{\text{hist}}$. Next, we construct a candidate event keywords set W composed of v and the two other words with the highest co-burstiness index. We also remove from W_P the words in W. Finally, we compute a *set burstiness index* for W as:

$$b_W = \frac{\sum_{i=0}^{2} N_{W,i}}{\sum_{i=0}^{2} N_i} - 3 \frac{\sum_{i=1}^{13} N_{W,i}}{\sum_{i=1}^{13} N_i} \tag{5}$$

where N_W is the number of T tweets containing at least two W words within the i-th past day. The iterative procedure stops when the set burstiness index computed at the last iteration is lower than the one computed at the previous iteration: the previously obtained W is the event keywords W^*. The coefficient for historical frequency used in Eqn. 3 and 5 has been chosen after preliminary experiments on a small fraction of our dataset and never changed in the remaining experimental evaluation (see Section IV).

Note that in this work we aim at identifying one single event a day, i.e., the top event keywords set W^*. Yet, in a real-world deployment, one could select more events at will, simply by varying the stop criterion of the above described iterative procedure.

C. Popularity qualitative assessment

A qualitative evaluation $Q_{\text{pop}} \in \{\text{low}, \text{medium}, \text{high}\}$ of the event popularity is computed as follows.

We compose an *event corpus* T^* by selecting all T tweets output in the last 3 days which contain at least two W^* words. We compute the T^* corpus *relative size* R_0—i.e., the ratio between the number of event-related tweets within the last 3 days and all topic tweets in the same 3 days—as:

$$R_0 = \frac{\sum_{i=0}^{2} N_{W^*,i}}{\sum_{i=0}^{2} N_i}$$

We then compose a set $\mathcal{R} = \{R_0, R_1, \ldots, R_{13}\}$ of the relative sizes of the event corpora of the last 14 days. Finally, we set Q_{pop} to high, medium or low if R_0 is in the $\mathcal{R}$ subset composed of its 1%, 25% or 100% greatest elements respectively.

D. Sentiment classification

Having constructed the corpus T^* of tweets that refer the identified event, we aim at classifying each tweet t of T^* as positive, neutral or negative, with respect to the sentiment of the tweet author.

To this end, we use a sentiment classifier which is built on a previously available, static, labeled, balanced training set T_L which is independent from T^*, T and the related topic. T_L is hence composed of tweets, each one has a label in {positive, neutral, negative}, which has been manually assigned to represent the probable sentiment of the corresponding author. The T_L is statically preprocessed as described in Section III-A, with the exception of the feature extraction procedure which is instead performed basing on T, i.e., on the words actually used recently for the given topic.

The classifier is built as follows. We choose the first 500 monitored words extracted from T and build the corresponding feature vectors over T_L by using unigrams. We generate an output numerical variable y on T_L by setting $y = 0$, $y = 0.5$ and $y = 1$ respectively for negative, neutral and positive labeled tweets. Then, we fit a linear model for y using all the T_L selected features: the 50 features which have the lowest p-value in the linear model are then selected, intercept excluded. Finally, the sentiment classifier is set to a SVM classifier trained on those selected 50 features to estimate the sentiment label, i.e., one value in {positive, neutral, negative}.

The sentiment classifier is rebuilt every time the monitored words set changes: in terms of machine learning, the training set instances remain the same, while the attributes may change. In practice, we found that the set of the first 500 most occurring W_M words changes rarely; moreover, the set of 50 features with the lowest p-value changes even more rarely. Hence, an actual retraining of the SVM classifier is needed, for a given topic T, only in exceptional cases.

We apply the sentiment classifier to the tweets of the event corpus T^*, i.e., we apply the SVM classifier on each feature vector obtained from the corresponding tweet in T^* by taking into account only the 50 features selected as described above. The T^* corpus is then partitioned in three *sentiment corpora* T^*_{pos}, T^*_{neut} and T^*_{neg}, possibly empty, according to the label assigned by the sentiment classifier. A qualitative evaluation Q_{sent} of the event sentiment polarity is set to $s = \text{positive}$, neutral or negative if the corresponding sentiment corpus T^*_s is the largest one.

E. Event summarization

We want to summarize the event represented by W^* words and covered by T^* corpus by selecting the top k representative tweets for each sentiment label.

To this end, for each sentiment corpus T^*_{sent} we proceed as follows. We (i) compute the centroid of the feature vectors of T^*_{sent}, then (ii) we select the k tweets whose corresponding feature vectors have the lowest Manhattan distance from the centroid.

In a preliminary study, we explored the use of the sum of cosine similarities as an estimate for the summarization ability of a tweet with respect to a corpus. We found results similar to the centroid-based technique described above, but longer computation times.

IV. Experimental evaluation

A. Data

We experimentally evaluated our approach by validating it against real-world events concerning three popular tech-related brands: Apple, Google and Microsoft. We used two different corpora of tweets, also known as Twitter status updates.

We used a subset of the corpus of [15] in order to simulate the Twitter text stream, which allowed us to perform an offline validation of our approach. With respect to the original version, we discarded the tweets generated before September 1st, 2009 because they are too sparse in time for our purpose. The remaining portion of the corpus contains 8,154,406 tweets output by users within the United States—mainly using a smartphone: the majority of them is thus written in English. In the resulting period (September 1st, 2009–March 15th, 2010), the corpus contains an average of 41,000 tweets per day. Figure 1 shows the number of tweets per day in this corpus.

We also used another freely available corpus for sentiment classification, as described in Section III-D. This second corpus was specifically designed by Sanders Analytics LLC[1] for training and testing Twitter sentiment analysis algorithms. It contains about 5,000 tweets obtained by searching Twitter in October, 2011 for the following search terms: `@apple`, `#google`, `#microsoft`, `#twitter`. Each tweet has been labeled by an English fluent operator with a label among {positive, negative, neutral, irrelevant}. For the purpose of this work, (i) we changed all irrelevant labeled tweets to neutral and (ii) balanced the dataset with respect to the label, hence obtaining a 3 labels balanced corpus T_L, composed of 1,611 tweets, 537 tweets per label.

We call the two corpora the Stream Corpus and Sentiment Corpus respectively.

[1] http://www.sananalytics.com

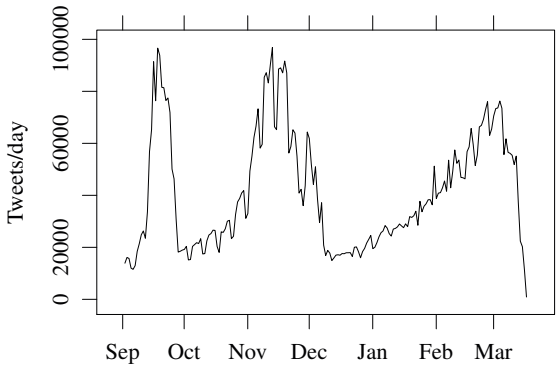

Figure 1. The number of tweets per day in the Stream Corpus.

Brand	Keywords	Tweets	T./day
Apple	`apple`, `iphone`, `ipad`, `ipod`, `ios`, `mac`, `macintosh`, `macbook`	53,680	272
Google	`google`, `android`, `chrome`	29,073	148
Microsoft	`microsoft`, `explorer`, `msn`, `windows`, `winxp`, `vista`, `xbox`	20,897	106

Table I
Topic matching keywords for each considered brand and summary of corresponding number of tweets in the Stream Corpus.

B. Results

We used the Stream Corpus to simulate execution of our approach once a day.

First, we compiled a static list of keywords for each considered brand (first two columns in Table I). Next, for each topic, we extracted the potentially relevant tweets by selecting those tweets which contain at least one of the keywords (see Section III). The number of tweets extracted for each topic is shown in the last column of Table I and, on a daily basis, in Figure 2. The figure shows a spike for the Apple brand, on January 28th, 2010, corresponding to the day after the announcement of the iPad by Steve Jobs. It is important to point out the extreme volatility of the tweets/day count, as this behavior makes event detection very difficult.

Finally, we applied the remaining key steps of our approach (preprocessing, event keywords identification, popularity qualitative assessment and event summarization) once a day, for each of the three sub-streams obtained above.

Brand	Date	Event keywords	Q_{pop}	Q_{sent}	Summarizing tweet for sentiment corpus given by Q_{sent}
Apple	09/10/21	`imac apple mouse`	high	pos.	`the new Apple Magic Mouse looks glorious! http://****`
Apple	10/3/14	`pre ipad order`	high	neut.	`@**** did you pre order your iPad?`
Google	10/3/12	`direction map bike`	high	neut.	`RT @****: (BikeRadar) Cycling directions added to Google Maps in US http://**** #procycling`
Microsoft	09/10/1	`security essentials microsoft`	high	pos.	`Microsoft Security Essentials is now available - free AntiVirus/Malware protection - http://****`

Table II
A FEW SIGNIFICANT EVENTS FOUND BY OUR APPROACH. USERNAMES AND URLS ARE MASKED BY $\star$.

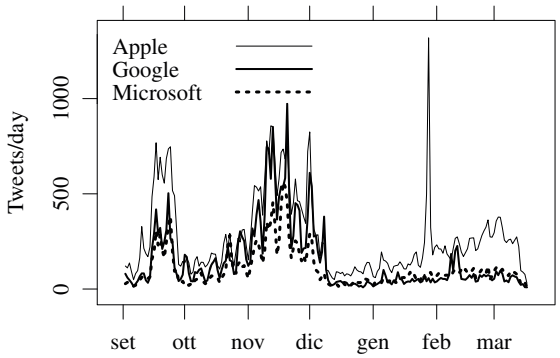

Figure 2. The number of tweets per day for the three brands.

	Q_{pop}			Q_{sent}		
Brand	low	med.	high	neg.	neut.	pos.
Apple	112	53	19	26	138	20
Google	117	41	26	12	134	38
Microsoft	115	53	16	41	119	24

Table III
Q_{pop} AND Q_{sent} VALUES SUMMARY.

Table III summarizes the values of the daily popularity assessment Q_{pop} and sentiment polarity assessment Q_{sent} over the 184 days of the considered period for the three brands. It can be seen that, in general, public sentiment seems slightly more positive towards Google-related events than Apple and Microsoft ones. Another finding is that, as expected, the number of highly popular events is low.

We manually explored the daily qualitative evaluations and summarizing tweets in order to validate them against real-world events. It is not feasible to express synthetically the outcome of the exploration. Instead, we briefly list in Table II some significant events that were indeed detected automatically by our approach. All shown event dates are close to the real event—either the same day or day after. This result suggests that our approach can provide a good precision.

V. CONCLUSIONS

Millions of Twitter users output every day a wealth of short texts, which may include opinions about things, facts, brands and so on. The ability to automatically extract suddenly popular themes, i.e., events, infer the general sentiment polarity on each event and finally summarize people corresponding opinions could be crucial for companies interested in monitoring their brand on Twitter.

In this paper, we propose a set of methods that address these needs and evaluate them experimentally on a real corpus of 8,000,000 tweets. We found that our methods can provide concise yet meaningful qualitative assessment of popular events related to a given topic, previously specified by means of a static set of keywords.

REFERENCES

[1] J. Bollen, H. Mao, and X. Zeng, "Twitter mood predicts the stock market," *Journal of Computational Science*, vol. 2, no. 1, pp. 1–8, Mar. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.jocs.2010.12.007

[2] E. J. Ruiz, V. Hristidis, C. Castillo, A. Gionis, and A. Jaimes, "Correlating financial time series with micro-blogging activity," in *Proceedings of the fifth ACM international conference on Web search and data mining*, ser. WSDM '12. New York, NY, USA: ACM, 2012, pp. 513–522. [Online]. Available: http://dx.doi.org/10.1145/2124295.2124358

[3] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith, "From tweets to polls: Linking text sentiment to public opinion time series," in *Proceedings of the International AAAI Conference on Weblogs and Social Media*, 2010, pp. 122–129.

[4] J. Bollen, A. Pepe, and H. Mao, "Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena," *CoRR, abs/0911.1583*, pp. 1–10, 2009.

[5] E. Kouloumpis, T. Wilson, and J. Moore, "Twitter Sentiment Analysis: The Good the Bad and the OMG!" in *Fifth International AAAI Conference on Weblogs and Social Media*, 2011.

[6] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, "Sentiment analysis of Twitter data," in *Proceedings of the Workshop on Languages in Social Media*, ser. LSM '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 30–38. [Online]. Available: http://portal.acm.org/citation.cfm?id=2021114

[7] 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*, ser. COLING '10. Stroudsburg, PA,

USA: Association for Computational Linguistics, 2010, pp. 36–44. [Online]. Available: http://portal.acm.org/citation.cfm?id=1944571

[8] D. Chakrabarti and K. Punera, "Event Summarization using Tweets," 2011.

[9] M. Cataldi, L. Di Caro, and C. Schifanella, "Emerging topic detection on Twitter based on temporal and social terms evaluation," in *Proceedings of the Tenth International Workshop on Multimedia Data Mining*, ser. MDMKDD '10. New York, NY, USA: ACM, 2010. [Online]. Available: http://dx.doi.org/10.1145/1814245.1814249

[10] M. Mathioudakis and N. Koudas, "TwitterMonitor: trend detection over the twitter stream," in *Proceedings of the 2010 international conference on Management of data*, ser. SIGMOD '10. New York, NY, USA: ACM, 2010, pp. 1155–1158. [Online]. Available: http://dx.doi.org/10.1145/1807167.1807306

[11] J. Weng, Y. Yao, E. Leonardi, and F. Lee, "Event detection in twitter," in *Fifth International AAAI Conference on Weblogs and Social Media*, 2011.

[12] S. Petrović, M. Osborne, and V. Lavrenko, "Streaming first story detection with application to Twitter," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, ser. HLT '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 181–189. [Online]. Available: http://portal.acm.org/citation.cfm?id=1858020

[13] A. Marcus, M. S. Bernstein, O. Badar, D. R. Karger, S. Madden, and R. C. Miller, "Twitinfo: aggregating and visualizing microblogs for event exploration," in *Proceedings of the 2011 annual conference on Human factors in computing systems*, ser. CHI '11. New York, NY, USA: ACM, 2011, pp. 227–236. [Online]. Available: http://dx.doi.org/10.1145/1978942.1978975

[14] N. Evangelopoulos and L. Visinescu, "Text-mining the voice of the people," *Commun. ACM*, vol. 55, no. 2, pp. 62–69, Feb. 2012. [Online]. Available: http://dx.doi.org/10.1145/2076450.2076467

[15] Z. Cheng, J. Caverlee, and K. Lee, "You are where you tweet: a content-based approach to geo-locating twitter users," in *Proceedings of the 19th ACM international conference on Information and knowledge management*, ser. CIKM '10. New York, NY, USA: ACM, 2010, pp. 759–768. [Online]. Available: http://dx.doi.org/10.1145/1871437.1871535