Exploiting Temporal Locality to Determine User Bias in Microblogging Platforms

Pedro H. Calais Guerra, Loïc Cerf, Thiago C. Porto
Adriano Veloso, Wagner Meira Jr. and Virgilio Almeida

Universidade Federal de Minas Gerais, Brazil
Computer Science Department
{pcalais,lcerf,tcp,adrianov,meira,virgilio}@dcc.ufmg.br

Abstract. Bias is the human tendency to favor one side of a discussion in argumentation, lacking neutrality and balance. Determining user biases is key to applications that process, interpret, and recommend content generated by those users in social media platforms. This paper addresses the problem of determining (in a supervised way) biases of microbloggers from the stream of messages. In this paper, we evaluate the use of a new criterion to identify user bias in social media systems: the temporal locality among users that have similar bias, i. e., the fact that people having similar biases express at about the same time. We show that this remarkable property indeed holds in some domains discussed in Twitter and may be explained mainly by the real-time use of the microblogging platform, i. e., users with similar biases react altogether to the outcome of events that are in accordance with their opinion (e. g., their favorite soccer teams scores a goal). Besides the precision of the computed biases, our proposal presents two major advantages that are consequences of not considering content at all (only temporal information is used). First, it is very efficient, i. e., a modest hardware can process on the fly the whole stream of messages about a popular topic commented in Twitter. Second, we believe that it may be applied to a wide range of domains regardless the language in which the messages are written. The experimental section of this paper reports the efficient learning of precise biases in both sportive and political contexts where the numerous messages are either written in English or in Portuguese.

Categories and Subject Descriptors: H. Information Systems [**H.m. Miscellaneous**]: Data Mining

Keywords: microblogs, opinion mining, social media, supervised learning, user bias, user profiling

1. INTRODUCTION

Microblogging platforms such as Twitter (or the similar features of social networks such as Facebook) grant the ability to express one's opinions in a real-time fashion. Web users now may comment political speeches, sports competitions, or any "buzz", *while* they occur [Kamath and Caverlee 2011; Guerra et al. 2011]. Taking into account this instantaneous aspect of the Web tells a lot about its users. For example, it allows to better answer Web search queries [Mustafaraj and Metaxas 2010] or to provide better recommendations [Phelan et al. 2009].

In this paper, we show that the time at which messages are sent in social media platforms (and, more specifically, in microblogs) can ease the automatic understanding of the opinions they express. Usually, algorithms rely on the content of the messages to infer the sentiment the author has w.r.t. an entity, e. g., a product, a service or a political person [Pang and Lee 2008; Turney 2002; Liu 2010]. In this paper, we argue that, in some scenarios, we can rely on *when* the messages were sent to predict the user views regarding a given topic. Those scenarios are the ones in which people comments are driven by real-life events that motivate them to react or not depending on their preferences. In the context of

This work was partially supported by CNPq, CAPES, FAPEMIG, FINEP and UOL (www.uol.com.br), through its Research Scholarship Program, Proc. Number 20110215235100.

microblogging, this paradigm shift may greatly enhance the quality of the results. Indeed, the ability to learn from the posted content not only suffers from the difficulty to process natural languages (that can be, e. g., ironic) but also from the shortness of the messages. For instance, temporal information is essential to learn anything about a supporter writing "Goal!" during a soccer match.

This paper deals with determining (or learning) the *bias* of microbloggers w.r.t. a given topic based on their *temporal behavior*. Social theories define bias as the lack of appropriate balance, neutrality and critical doubt in argumentation [Walton 1991]. Bias is inherent to humans, which tend to support one side too strongly or too often [Kienpointner and Kindt 1997]. This behavior is particularly noticeable with sport or political supporters and our proposal makes the assumption that it does not occur at random times but, usually, in reaction to particular events. As a consequence, people having the same bias would often express it at the same time on a microblogging platform. For instance, soccer supporters will write their joy after a goal for their team, political partisans will be eager to comment a new scandal involving the opposing party as soon as it is revealed, etc. Furthermore, most microblogging platforms make it easy to repeat the message of another user. The use of such mechanism amplifies the link between temporal locality and similar bias. Indeed, repeating a message indicates an endorsement of its content (hence the similar bias [Guerra et al. 2011]) and usually happens within a short period of time because it only takes one click to do so and the carried information quickly is considered outdated.

We have designed, implemented and evaluated a novel and simple algorithm that learns the bias of microbloggers from the timestamps of their messages. It may be applied to any topic (e. g., the Brazilian soccer championship or an electoral campaign) and regardless of the language (e. g., Portuguese or English). In fact, the actual content of the messages is not used at all by our approach and that explains another advantage of our algorithm: its efficiency. Even being executed on a modest hardware, the whole stream of messages about a given topic may be processed on the fly. In comparison, a real-time analysis based on the written content may become intractable when considering the rate of the message flow about a popular topic (such as soccer) on a popular platform (such as Twitter).

This temporal "pattern" in the usage of microblogging platforms is remarkable and opens-up great perspectives. Of course, the bias is, by itself, a useful piece of information. Nevertheless, other interesting measures may be (at least partially) derived from it. For instance, it may help in identifying the *reliability* of a content generator (e. g., a political commentator). It can also be key to provide personalized services such as recommendations [Hofgesang 2007] because, unlike the posted content that fluctuates w.r.t. external perturbations (e. g., along a discussion with an opponent), the bias is consistent and robust [Guerra et al. 2011]. It is valuable as well for the light it sheds on this subject.

Our main contributions are:

—we show that microbloggers do not express their opinions at random times but follow a temporal "pattern" that reveals their bias;

—we propose an efficient linear time algorithm that solely relies on temporal information to learn these biases;

—we apply and validate our approach using messages from three topics that drive live reactions in the Twitter microblogging platform: the Brazilian 2010 Soccer League, the 2010-2011 National Football League (NFL) season and the 2010 Brazilian Presidential Elections. Our results show that temporal information greatly helps in discovering user preferred sides. In fact, our proposal correctly identifies the dominant opinion of between 70% and 80% of the users who frequently post content on Twitter.

The remainder of this paper is organized as follows. Section 2 discusses the related work. Section 3 presents our algorithm. Section 4 empirically studies its behavior. Section 5 concludes and presents future research directions.

2. RELATED WORK

Bias has been defined by sociologists as a trace of people or argumentations that exhibit one or more of the following characteristics [Walton 1991]: 1) lack of appropriate balance and neutrality in argumentation, 2) lack of appropriate critical doubt, 3) the arguer has a particular position over a subject, and 4) the arguer has a personal interest in the outcome of an argument or discussion.

Most studies focus on the biases of news media. In this context, [Earl et al. 2004] distinguishes the *selection bias*, which guides the decision of which events are covered (a biased media outlet will favour some types of news), from the *description bias* — the veracity of the media coverage. A way to measure media biases is to count the number of times a particular media outlet cites various think tanks and policy institutes. By comparing those numbers to the citation rates of the same organizations by congressmen, a strong liberal bias has been unveiled in the US news media [Milyo and Groseclose 2005]. Our proposal focuses on the selection bias. It assumes that sending a message using the microblogging platform indicates the selection of the event it is about. Although the selection bias has been originally defined for news media (and some users of Twitter are indeed media [Kwak et al. 2010]), our proposal successfully applies to "regular" users as well.

The study of biases in social media has mainly focused on bloggers. The main approach to map the opinions of the blogosphere relies on the link structure that connect the blogs. It assumes that blogs with similar views are likely to cite each other [Gamon et al. 2008]. In this way, US political blogs can be clustered into two main groups (liberal/conservative) [Adamic and Glance 2005]. A more recent work uses the graph induced by endorsements (e. g., "retweets" in Twitter or the "Like" button in Facebook), which are assumed to indicate similarities of views, to predict user biases [Guerra et al. 2011]. Those approaches, although highly accurate, may be computationally prohibitive, i. e., unable to process on the fly the stream of messages about a popular topic.

In this paper, we propose to observe the *temporal behavior* of microbloggers to discover their biases. This idea is motivated by previous works that have shown temporal patterns in social media systems. For example, the popularity of textual phrases and hashtags along time follows six different patterns [Yang and Leskovec 2011]; the communications among social media users form transient communities (i. e., clusters that discuss a topic in specific moments) [Kamath and Caverlee 2011]; among other works. Thus, taking into consideration the temporal aspects of the microblog analysis looks promising. Indeed, one of the main appeals of microblogging is the ability to comment live events. A selection bias should make it more likely that users express themselves right after (or, maybe, before if the event is planned) an event supporting their bias. For example, a soccer supporter is more likely to react to a goal scored by her favorite team than by another team (she may react to goals made by the opponent, but over all data, she will more often comment her favorite teams' events). A political partisan is eager to immediately comment a faulty statement made by and adversary candidate. On the contrary, partisans of the candidate prefer to ignore her/his declaration.

Even without any information about those external events and without considering the written content or the relations between the users, our work shows that it is possible to learn user biases from those of known users who express themselves at about the same time (supervised learning). Notice that this temporal locality has nothing to do with concept drift [Widmer and Kubat 1996]. Supervised classifiers taking the latter effect into consideration [Rocha et al. 2008; Salles et al. 2010] assume that the relationship between attributes and classes changes over time.

3. LEARNING USER BIAS BY TEMPORAL LOCALITY

In this section we present our proposal for addressing the problem of determining the user bias based on temporal information.

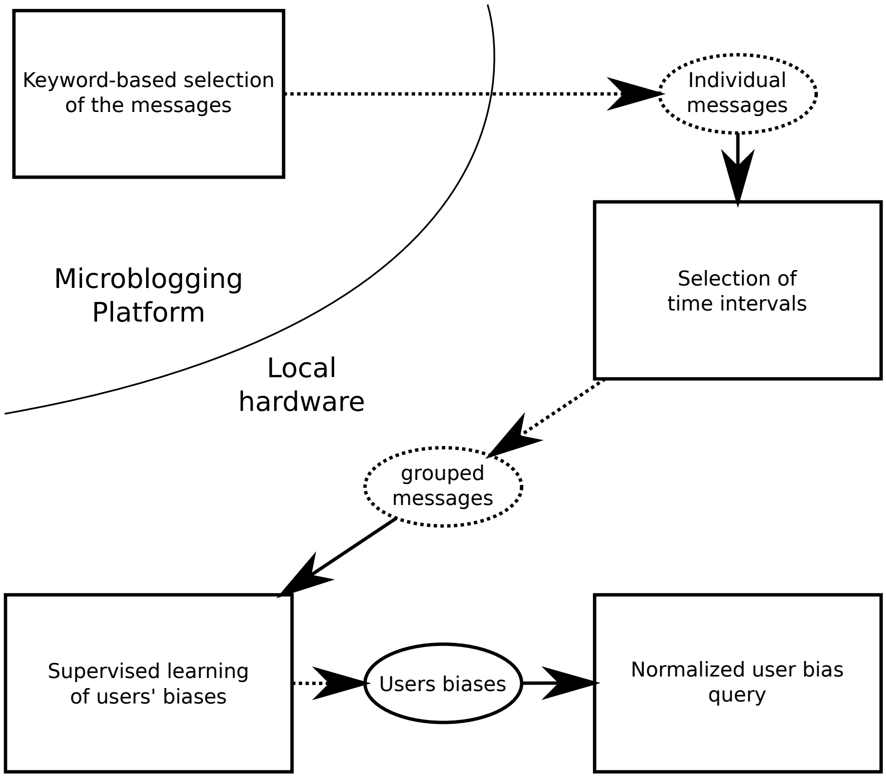


Fig. 1: The stream of messages, from the microblogging platform, is divided into time intervals. In each time interval, the users with known biases are used to learn more about the biases of all users in the same interval. A separate process takes care of the needed normalization that remains to be done when a specific user bias is queried.

3.1 Architecture of the Approach

Figure 1 depicts the architecture of our approach. Microblogging platforms such as Twitter have an API that allows to collect the messages based on the presence of keywords in their content. However, once outside the microblogging platform, our approach completely disregards the content of the messages. That is why, in the remainder of this paper, the word "message" only stands for the time it was sent and its author. In Figure 1, every rectangle represents a separate process. Those processes read their input (plain arrow from the source to the reader) and write their output (dotted arrow from the writer to the destination) in either a buffer (dotted ellipse) or a database (plain ellipse).

The remainder of this section presents the successive processes:

—the selection of time intervals that leads to grouping the messages;
—the supervised learning of users' biases, which is the core of our contribution;
—the normalization of a bias performed when that of a user is queried.

3.2 Selection of Time Intervals

Our approach assumes that users having similar biases are prone to send messages at about the same time. To clearly show that this assumption holds, the experimental section of this paper relies on a very simple method to temporally group the messages: a fixed duration is chosen and the whole stream is binned accordingly (without any overlap of the time intervals). Nevertheless, it is interesting to point out that more sophisticated methods may lead to better results. For example, since the synchronicity of the users' expression is explained by events that trigger them, one could try to detect these events.

Intuitively, they should correspond to "peaks" of activity. Detecting those peaks in the stream of messages is an interesting topic and known algorithms (e. g., [Azzini et al. 2004]) that analyze time-series data can perform this task. The temporal basis of the peaks would then define intervals in which messages are grouped. Weighting the subsequent learning step with the salience of these peaks may also improve the results. However, the main goal of this paper is to stress the mere existence of a temporal "pattern" that allows to efficiently learn the biases of the users. This goal would be weakened by the many parameters required to configure a peak detection algorithm. That is why the investigation of more sophisticated definitions of the time intervals is left as future work. Notice however that Section 4.3 shows that better precisions are achieved by discarding the time intervals with too few messages.

3.3 Supervised Learning of Users' Biases

3.3.1 *Defining Known Biases.* Let $\mathcal{C}$ be a set of classes. For example, $\mathcal{C} = \{$democrat, republican, green$\}$ could be used to learn political biases of American users of a microblogging platform. We represent bias as a vector of $|\mathcal{C}|$ positive values. Those values are higher when they relate to classes the user is biased toward. The considered setting is that of supervised learning, i. e., an associative array, B_{known}, is defined beforehand and relates the users' IDs to their known biases. These biases may be characteristic vectors, i. e., they may contain one value that equals to 1 while all remaining values are 0. For instance, known politicians' biases could be at 1 for their party and 0 for the other parties. However, these vectors must not necessarily be so. For instance, a democrat senator who is known to be close to the ideas of the green party may be considered known and her bias defined (in B_{known}) as $(0.6, 0, 0.4)$, where the first and third values respectively relate to the democrat and the green class. The norm of the known biases must not necessarily be the same either. In this way, the analyst can encode the certainty of the known biases.

A way to automatically define the known biases is to look at the content of a fixed subset of messages (by opposition to working on the stream). For example, each class can be associated with some terms that its members frequently use (e. g., the mere name of the class). By counting, for each user, the number of occurrences of the words associated with each class, one can assume that the users writing a lot of such words (threshold to fix) is known and her bias is the vector of the counts of words. This step is computationally demanding since the content must be read. However, it does not need to be performed on a huge quantity of messages and is separated from the actual learning algorithm, which processes the stream of messages without reading them.

The biases of the most vocal users are expected to be durable measures. As a consequence, B_{known} can be defined once for all. However, and depending on the application, this durability may have some limits. For example, a political partisan may, after some years, move to another party. That is why it makes sense to update B_{known} from time to time. This update does not require to stop the learning process presented in the next section. Furthermore the learning process also learns the bias of the known users (but does *not* modify B_{known}). As a consequence, an increasing dissimilarity between the learned vector and that in B_{known} could be used to indicate when a supposedly known bias becomes inaccurate.

3.3.2 *Algorithm.* Algorithm 1 details how (absolute) biases are learned. It reads that from a stream of messages and group them in intervals (if the buffer is empty, it waits for one). The multiset U contains all users which posted a message during that interval. U is a multiset because users can post more than once in a same interval. For instance, $\{U_1, U_1, U_2, U_3\}$ is a multiset in which the user U_1 posted twice. For every group U, Algorithm 1 processes the messages in two steps. First of all, it learns, from the presence of known users, what is the bias V of the group of messages. Then, every user u in the group has its bias $B[u]$ updated w.r.t. the group bias V.

For each interval, the biases *learned* for all k known users is computed. The idea is here to quantify

Algorithm 1 Learning biases by temporal locality.

$S \leftarrow (0, \ldots, 0)$
while true do
  $U \leftarrow$ GETUSERSINAGROUP()
  $k \leftarrow 0$
  $V \leftarrow (0, \ldots, 0)$
  for all $u \in U$ **do**
    if u is a key of B_{known} **then**
      $V \leftarrow V + B_{\text{known}}[u]$
      $k \leftarrow k + 1$
    end if
  end for
  $S \leftarrow S + k.V$
  for all $u \in U$ **do**
    $B[u] \leftarrow B[u] + V$
  end for
end while

the bias of the whole microblogging system (restricted to the known users as always with supervised learning) during the interval. Summing these vectors, interval after interval, in the global variable S, therefore gives the bias of the microblogging system for the whole stream of messages until the current interval. It is useful to compute a normalized bias when queried. Section 3.4.1 details this process and clarifies why this computation for S makes sense.

B is a global variable too. It is an associative array (initially empty) that associates each user with her absolute bias. Again, it will be explained later why the desired biases are not directly those absolute vectors but a relative version taking into account the overall bias S. To expose the pseudo-code in the simplest possible way, $B[u]$ is assumed to be the null vector when u's first message is processed.

3.3.3 *Complexity Analysis.* During the iterative computation of the vector V (the bias of the processed group of users), the associative array B_{known} is accessed $|U|$ times. Each of these accesses has a constant time complexity. Indeed, B_{known} being unaltered after its creation, any of its values can be accessed through a perfect hash function. As a consequence, in the worst case (U only contains users with known biases), this step has a $O(|U|.|\mathcal{C}|)$ time complexity.

During the update of users' biases, the associative array B is also accessed $|U|$ times but, this time, new users are inserted. As a consequence, by implementing B as a hash map, access and insertion operations may suffer from possible hash collisions [Skiena 1998]. In the worst case (all entries in the same bucket), the complexity of this step would be $O(|U|.(|B| + |\mathcal{C}|))$. However the amortized complexity only is $O(|U|.|\mathcal{C}|)$. In a real-time context (where only the worst case scenario matters), it is preferable to access and insert the content of B through a self-balancing binary tree. The cost of one of these operations becomes $O(\log(|B|))$, hence an overall time complexity of $O(|U|.(\log(|B|) + |\mathcal{C}|))$ for processing a given group of messages (the complexity of the first step being negligible). This computational cost is far smaller than those of the approaches based on the content (our approach completely disregards it) or those exploring a graph of users.

Since there may be a lot of users (memory management), since their biases should not be lost in case of failure (durability), since the querying process must access these biases (isolation) and for the other assurances brought by the respect of the ACID principles [Gray 1981], B, B_{known} and S are stored in database tables with the indexes chosen as explained above. The space to store these variables is $O((|B| + |B_{\text{known}}|).|\mathcal{C}|)$.

3.4 Querying Biases

3.4.1 *User Bias.* Given a user u, $B[u]$ is her absolute bias. Alone, this information is not that valuable. Indeed, it is computed from the simultaneous expression with known users whose total quantities of messages may not be balanced, i. e., some opinions may be over-expressed (in particular if the repartition of the known users is not that of the classes). For example, in the American political context, B_{known} may globally be biased towards a party and, furthermore, the known users biased toward this party may be more vocal, i. e., write more messages. Only looking at the absolute biases would overestimate the support of the users to this party, a problem known in machine learning theory as class imbalancement [Ling and Sheng 2007].

To avoid this pitfall, every class is supposed to globally have the same weight. More precisely, the absolute bias (in B) is normalized so that the sum of the normalized biases of every known user is the vector $(1, \ldots, 1)$. Algorithm 2 does so. It divides every component of a user absolute bias by the sum of the respective component over all known users. These sums actually are the vector S computed by Algorithm 1, i. e., the overall absolute bias of the known microbloggers. Notice that the normalization is performed when querying the user biases. Normalizing the database would be less efficient unless the bias of the average user is queried more than once per interval. Indeed, directly storing the normalized biases would require an update of every entry at every time interval because S is updated with this frequency.

Algorithm 2 User bias.

Input: a user u.
return $\left(\frac{B[u]_1}{S_1}, \ldots, \frac{B[u]_{|C|}}{S_{|C|}}\right)$

The components of a normalized bias are proportions. Therefore, the orientation of a vector is to be understood in a relative way. For example, in a political context, a user commenting a few live declarations from a minor party (in the sense that these declarations do not draw many messages on the microblogging platform) may have a strong normalized bias towards this party even though she commented as well many more declarations from a major party (but fewer in terms of proportion of the generated "buzz"). Another interesting observation is that the absolute bias carries useful information too: each of its components quantifies the certainty of the orientation (given by the normalized bias) towards the related class. The larger (i. e., the more messages from known users having a bias towards this class and expressing themselves at the same time as the assessed user), the more certain.

3.4.2 *Overall Bias Along Time.* In Algorithm 1, the vector V gives the absolute bias of the group of messages that is currently processed. This vector can be normalized in the way described in Section 3.4.1. Observing it across time allows to have a glance at the evolution of the bias of the microblogging platform as a whole. In a near future, we plan to implement this idea within the "Observatório da Web[1]", a system that gathers, processes and presents aggregated data collected in blogs, microblogs, and news media sites.

3.4.3 *Running Example.* This section illustrates our algorithm on a small example. Assume three classes (A, B and C) and users that have known biases toward one class only. More precisely these users have, in B_{known}, 1 for the class they are biased toward and 0 for the other classes. The goal is the prediction of the (normalized) bias of a user X using the co-occurrence (i. e., the occurrence in the same intervals) of her messages with those of the known users. Table I lists all relevant information of a hypothetical stream of data until the time when the query is received. More precisely, the second column indicates whether X posted during an interval, V is the bias of an interval, and k is the

[1] www.observatorio.inweb.org.br

Table I: Hypothetical dataset of users from classes A, B and C posting messages in reaction to live events.

interval	**post from user X?**	k	V	$k.V$
1	yes	750	[A = 500, B = 200, C = 50]	[A=375,000 B=150,000 C=37,500]
2	no	180	[A = 100, B = 50, C = 30]	[A=18,000 B=9,000 C=5,400]
3	yes	30	[A = 10, B = 10, C = 10]	[A=300 B=300 C=300]
4	no	70	[A = 10, B = 50, C = 10]	[A=700 B=3,500 C=700]
5	yes	300	[A = 200, B = 0, C = 100]	[A=60,000 B=0 C=30,000]
6	no	250	[A = 100, B = 100, C = 250]	[A=25,000 B=25,000 C=62,500]
7	no	1000	[A = 50,B = 50,C = 900]	[A=50,000 B=50,000 C=900,000]

number of messages from known users in an interval. Because, in this example, every known bias has a component at 1 and the other components at 0, k actually is the 1-norm of V, i. e., $k = \sum_{i=1}^{3} V_i$.

If we run Algorithm 1 over Table I, we will get $B[X] = [A = 710, B = 210, C = 160]$, which is the sum of the number of times members of each class manifested in intervals in which user X also posted a message (intervals 1, 3 and 5). Algorithm 1 also computes $S = [A = 375,000 + 18,000 + 300 + 700 + 60,000 + 25,000 + 50,000 = 592,000, B = 150,000 + 9,000 + 300 + 3,500 + 30,000 + 25,000 + 50,000 = 237,800, C = 37,500 + 5,400 + 300 + 700 + 30,000 + 62,500 + 900,000 = 1,036,400]$, which is the bias of the whole system. We then apply Algorithm 2 to neutralize the effect of some classes being more intense than others in the system (for example, class A may have more adopters, and thus, any computation of bias will tend to favor A) and our estimation of bias of user X based on the moment he posts messages is $B[X] = [A = 0.0012, B = 0.0008, C = 0.0001]$.

Thus, our algorithm predicts that X should be inclined to support class A. Note that this is expected since user X posted on a time interval (interval 1) in which we observed a high fraction of posts from users belonging to class A, what indicates that some event on the real world (a goal from a team, for example) just took place and caused reactions from members of that class. On the other hand, user X did not manifest at interval 7, which is clearly an interval which attracted comments from class C. The result is that our algorithm predicted a very low probability of user X belonging to class C.

4. EXPERIMENTAL RESULTS

This section presents an evaluation of our proposal on real data collected from the Twitter microblogging platform. The datasets are first presented in details. They are then used to analyze several aspects of the learning procedure.

4.1 Data Collection

Twitter is the most popular microblogging platform. It allows any user to send short messages which instantly become available to the users who “follow” the message’s author [Kwak et al. 2010]. Since the service has been launched, in 2006, millions of users have been writing more than 10 billions “tweets” (as “messages” are named in Twitter). These messages can be comments about one’s personal life, can be interesting resources one wants to share with their followers, but it can also be opinions on polemic topics. As explained earlier in this paper, the real-time use of Twitter makes the latter category of messages particularly valuable to learn the bias of a user. Other noticeable uses of the instantaneous aspect of Twitter include the ability to signal early warnings of earthquakes [Sakaki et al. 2010].

In this work, we consider two common topics that generate debate on Twitter: politics and sports. The former category is represented by the 2010 Brazilian Presidential Election and two sport championships are processed: the 2010 Brazilian Soccer League and the 2010-2011 US National Football League. Table II gives a general overview of those three datasets. In all cases, the messages are

Table II: General overview of the datasets.

	Brazilian Soccer League	**NFL**	**Brazilian Elections**
period	2010-05-08 to 2010-12-15	2010-01-10 to 2011-02-17	2010-01-01 to 2010-12-17
bias	team preference	team preference	candidate preference
# of classes	12	22	2
# of tweets	35,834,453	23,094,280	10,173,381
# of users	5,638,906	4,230,731	1,272,723
avg. # of tweets/users	6.4	5.5	8.0
# of known (labeled) users	348,752	171,121	18,744
# of time intervals (minutes)	365,986	262,541	397,428
avg. # tweets/interval	98	88	26
execution time (seconds)	215	94	48
avg. processing time/interval (seconds)	6.10^{-4}	3.10^{-4}	1.10^{-4}

collected via the keyword-based Twitter API[2]. In the Brazilian political context, the names of the two main presidential candidates (Dilma Rousseff and Jose Serra) are used. In the sportive contexts, the team names are chosen keywords (the 12 most popular teams of the 2010 Brazilian Soccer League; all 22 teams of the 2010 NFL Seasons). Notice, in Table II, that our proposal runs very quickly; it only requires a few minutes to process any of the three datasets, which represents months of data. Indeed, each interval is processed (messages are grouped and bias is learned from the group) in less than a milisecond (see Table II). Given the time complexity of our approach, the single core machine with 24GB of RAM memory we used in our experiments should handle applications with hundreds times more messages per interval (if such applications exist).

Our proposal learns biases in a supervised way. As a consequence, some biases must be *a priori* known. We decided to select users which are clearly biased towards one class only. We considered three distinct sources of evidence to generate ground truth of users:

(1) Their use of specific hashtags (hashtags are words that are emphasized in the sense that Twitter use them to classify the message) helps their identification. For example, users who post *#goJets* are likely to support the New York Jets football team.
(2) Some users explicitily unveil their team preference in the description of their Twitter profiles, by mentioning the name of their favorite team.
(3) In the two sportive contexts, the number of times a user refers to a team (either with a hashtag or with a simple mention) is counted. Whenever the most cited team is three times higher than the sum of the mentions of all other teams, the related user is supposed known and associated with a bias that is the characteristic vector of the team. In the context of the Brazilian Political Election, only hashtags, and not mentions, are considered. Indeed, candidates are often mentioned in a negative way what makes this information a very uncertain indication of a support.

Users who do not fit in any of those three criteria are considered unknown. Table II gives the number of known users in each application.

The considered topics being highly dynamic, a 60 second time interval is chosen. Figure 2 shows the cumulative distribution of the number of messages per time interval. In all datasets most time intervals contain very few messages: about 60% (Elections), 50% (NFL Season) and 35% (Brazilian Soccer League) of the time intervals gather less than 10 tweets, whereas less than 5% of the intervals contain more than 1,000 messages. These intervals relate to periods during which users are reacting to live events. Interestingly, sport competitions drive more spikes of activity than political discussions.

[2]available at http://apiwiki.twitter.com/

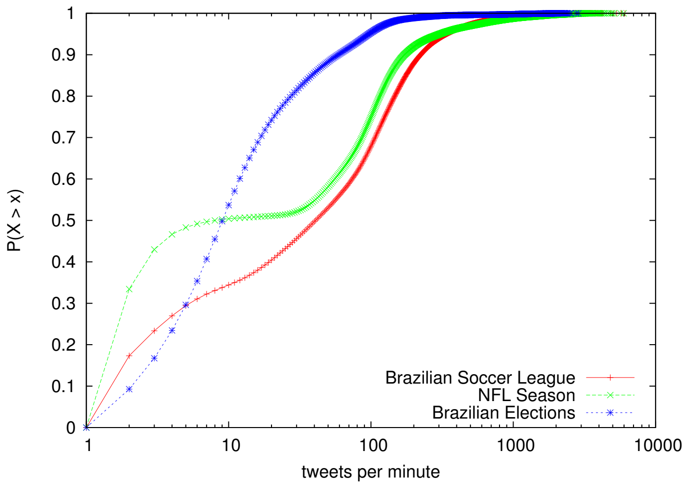


Fig. 2: Cumulative distribution of the number of messages per minute. Whatever the dataset, most of the one-minute intervals contain very few messages.

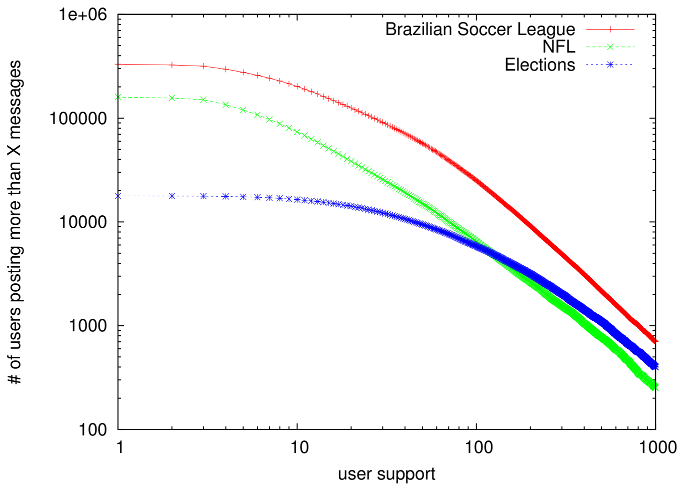


Fig. 3: Number of users w.r.t. the quantity of messages they send. Most of users send less than 10 messages over the period we analyzed, in all three datasets.

This can be explained by the fact that politic events are not commented live like sport matches. Figure 2 clearly shows it: in sportive contexts only, an inflexion of the curve separates the intervals during live events from those outside these events. In contrast, most of the political events are unplanned, therefore are less prone to be commented live. Nevertheless, the following sections show that, even in this context, biases can be learned from the temporal locality. This locality is just more diffuse.

Most microbloggers send very few messages but a few of them are very active. This classical distribution is plotted in Figure 3 and explains why, in the remaining of this section, the analysis is focused on the users who author less than a few hundreds tweets. The subsequent sections evaluate the applicability of our approach. All quality scores are the result of a traditional 5-fold cross validation.

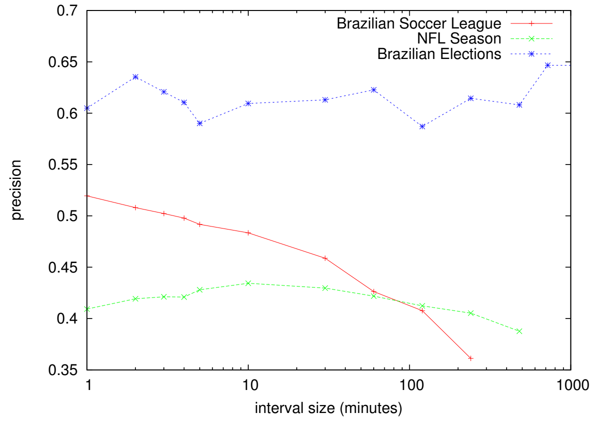


Fig. 4: Precision according to different interval sizes.

4.2 Precision versus Interval Size

The only parameter of our algorithm is the size of the temporal windows used to group messages. In Figure 4 we vary this size and evaluate its impact in accuracy. As we can see, impact of interval size depends on the scenario, specially when we contrast sportive and political scenarios: in the Brazilian Soccer League dataset, precision decreases as the interval increases, this is, indeed, a result that shows how temporal locality is high in this scenario. On the other hand, precision for NFL supporters start decreasing after we consider intervals of 60 minutes, what indicates that these users may have a different way of reacting during games than the Brazilian soccer supporters. Finally, precision for Brazilian Elections is somewhat independent of time interval, what is probably a result of the fact that political comments are not driven by instantaneous real-life events, but more long-term events such as a scandal about a candidate which is debated during a whole week.

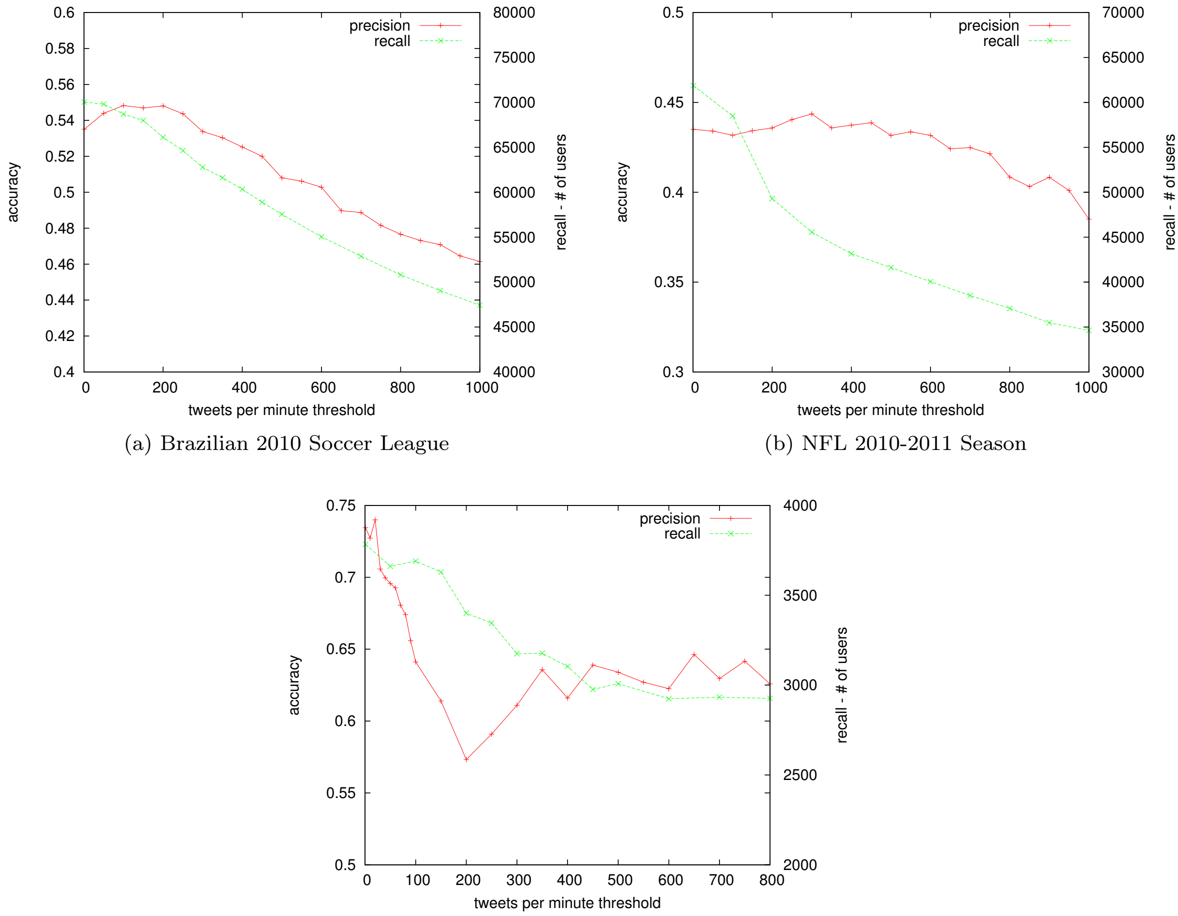
4.3 Precision versus Minimal Number of Messages in the Intervals

Let us first assess whether the intervals with little activity (i. e., a few messages) makes it harder to learn the users' biases. Figure 5 plots the precision (given a user, only the class that dominates her bias is considered) and the recall (i. e., the number of users whose biases are learned) obtained when more and more of the least active intervals are discarded (the x axis gives the minimal number of messages an interval must contain to be considered). Equations 1 and 2 detail how precision and recall are calculated.

$$\text{Precision} = \frac{|\{\text{users whose class has been correctly predicted}\}|}{|\{\text{users who have posted at least once during the intervals considered}\}|} \tag{1}$$

$$\text{Recall} = \frac{|\{\text{users who have posted at least once during the intervals considered}\}|}{|\{\text{total number of users in the dataset}\}|} \tag{2}$$

Of course, the recall monotonically decreases when we require more activity in order to consider a interval, as less data is considered as we increase this threshold. The precision first increases and then decreases. Back to Figure 2, it appears that, the maximum is obtained when only the top 20% (even less for the NFL data) of the intervals (w.r.t. the number of messages they contain) is considered. This confirms the supposed reason for the observed temporal locality. Users express their bias in reaction to live events. When no such an event happens (or is about to happen when the event is planned), considering the related time periods makes it harder to learn the biases. Note that



(a) Brazilian 2010 Soccer League

(b) NFL 2010-2011 Season

(c) Brazilian Elections

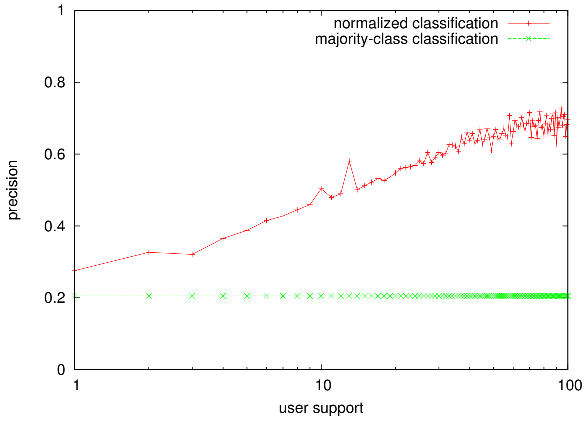
Fig. 5: Recall and precision w.r.t. the minimal number of messages a time interval must contain to be considered in this experiment. The recall decreases when less data is considered. The precision reaches a maximum when the periods of least activity are ignored.

simultaneous events can drive users from different viewpoints to manifest at the same time; those multiple events will help on discriminating those viewpoints from the viewpoints which are currently not related to those events. In general, however, most events should occur in isolation, for example, most of goals and touchdowns do not occur at the same time even if multiple matches are taking place simultaneously.

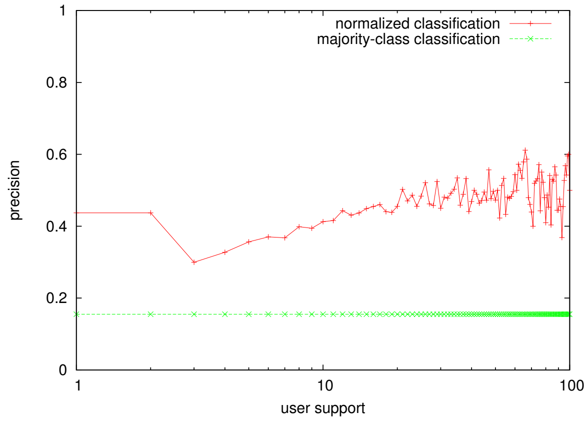
Let us emphasize, however, that our proposal does not *require* to fix such a minimal number of messages per time interval. Indeed, in Figure 5, the precisions at 0 are not much different from those at the maximums of the curves. The achieved precisions are far above those of a random learning ($\frac{1}{|\mathcal{C}|}$). Almost 55% (vs. 8.3% for a random learning) of the soccer supporters, 45% (vs. 4.5%) of the NFL self-made commentators and 75% (vs. 50%) of the users interested in Brazilian politics have their dominant bias that are correctly learned by our proposal.

4.4 Precision versus User Frequency

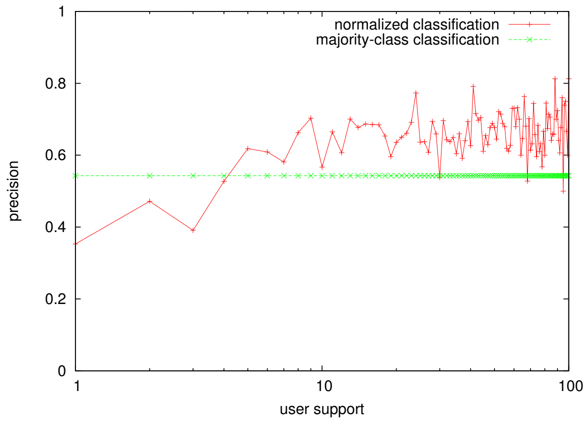
Another aspect that certainly affects the ability to automatically learn a given users' bias is how often she expresses herself on the studied topic. Figure 6 shows the precision of the bias w.r.t. the number



(a) Brazilian 2010 Soccer League



(b) NFL 2010-2011 Season



(c) Brazilian Elections

Fig. 6: Precision of the normalized biases w.r.t. to the number of messages available to learn it. In Sport contexts, our proposed technique is significantly better than a majority-class classifier. In the political context, usefulness of temporal information in detecting users' bias is less evident, although it is still better than a majority-class classifier

of messages that are available to learn this bias. Each point in the plotted curves corresponds the average precision the users having sent, in the considered data, the number of tweets defining the abscissa.

Of course, the more messages are sent, the better the user's bias is learned. Nevertheless, in the sport contexts, even when one single message is available, the user bias is learned in a way that is significantly higher than the result of a majority-class classifier, i.e., a classifier which always guess the most frequent class. In the political context, it is remarkable that, although political comments on Twitter do not generate significant peaks of activity (see Figure 2), biases can be learned from the temporal proximity between users' posts. We believe that the events, in reaction to which the users express their biases, are more diffuse than those of sport (e. g., goals). For example, politically biased users are keen to send messages 1) when a scandal involving the political opponent is revealed, 2) when their candidate is on TV presenting her/his proposals, or 3) during a debate that turns in favor of her candidate of choice. As a consequence, longer time intervals may improve the results some more.

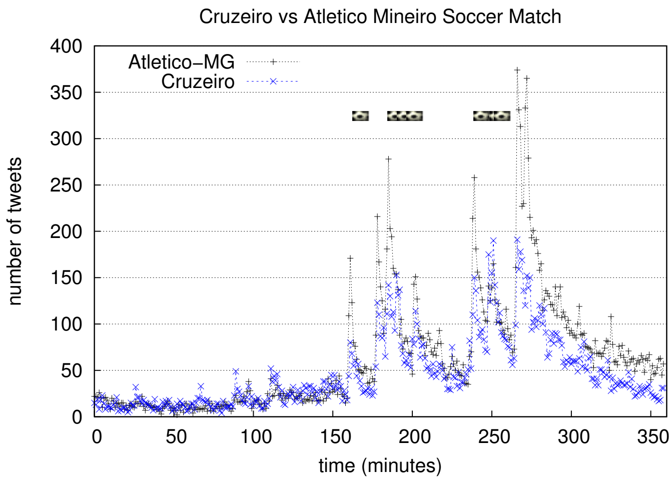


Fig. 7: Cruzeiro 3 – 4 Atletico Mineiro match. When a team scores, its supporters send a lot more messages than the supporters of the opposite team. After Atletico Mineiro's win, at 270', its supporters celebrate even more intensively.

4.5 Monitoring Events in Real-Time Using User Bias Knowledge

In this subsection, our proposal is shown to be valuable to understand the evolution of Twitter's bias as a whole. Figure 7 shows the number of supporters of the teams Cruzeiro and Atletico Mineiro, posting during the Brazilian Soccer match which clashed them on October 24, 2010. The number of messages increases as soon as the match begins at time=150. At minute=200, Atletico Mineiro was already winning by 3-0, and, indeed, three spikes are observable in the interval [150,200]. At those spikes, comments from supporters of Atletico Mineiro are at least twice more frequent than those from Cruzeiro's supporters. The reason is simple: users celebrate the goals of their favorite teams more than they comment on the failure of its defense. Thus, monitoring the bias of who is expressing itself on Twitter allows to infer to whom the event is in favor of. In some way, this can be understood as a group sentiment analysis.

We used our proposal to follow Twitter's bias along the Pittsburgh Steelers vs. Green Bay Packers NFL match. Figure 8 not only plots the number of tweets sent by supporters of each of the two competitors, but also the number of messages authored by users biased towards other teams. During this match, most of the comments came from Steelers fans. At the 260^{th} minute, when the game ended and the Packers became NFL Champions, an activity spike occurs. Although there were, during the match, a minority of Packers supporters, they dominate this spike.

Following the dynamics of Twitter's bias during events is a straight-forward application of our approach. As users' bias are learned from their previous messages, this analysis can be performed "online" (i. e., as the events occur and the user biases are learned). In other terms, as long as a "buzz" is associated with a class, our proposal can catch it while it occurs and in a content-independent way.

5. CONCLUSIONS AND OUTLOOK

In today's Web, several microblogging platforms allow anyone to comment events as soon as they occur. Because of such resource, users who are biased preferably comment the events that support their opinions, we have assumed that microbloggers with similar biases send messages at about the same time. We have designed a very simple algorithm only relying on this idea. It partitions the stream of messages w.r.t. to a fixed time interval and computes, from known users in every interval (supervised learning), the bias of this interval and modify its users' biases accordingly.

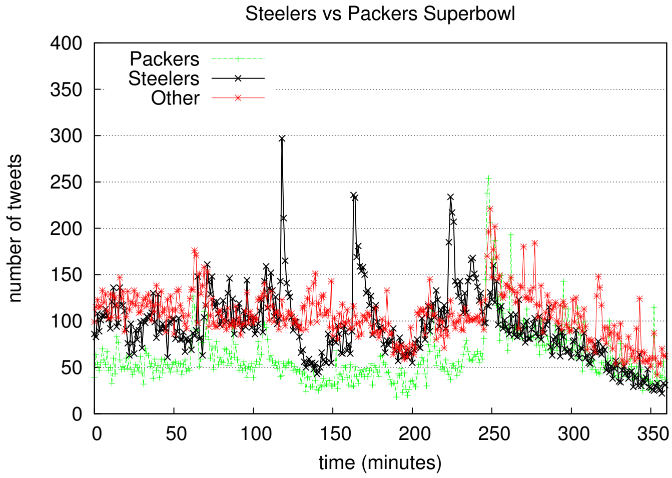


Fig. 8: SuperBowl match between Pittsburgh Steelers and Green Bay Packers, won 31–25 by the Packers. Packers supporters react more intensively once the match comes to an end, i. e., after the team is proclaimed NFL Champions. The match also caught the attention of users biased towards other NFL teams.

Our results show that temporal information can add useful knowledge about users in three scenarios (soccer, football and political discussion in Twitter). Therefore, we can confirm our assumption: microbloggers with similar biases tend to express themselves synchronously. Besides those results, we have emphasized the two main advantages of our approach. First of all, it is applicable to any topic that is discussed live on the microblogging platform, whatever the language of the discussion. Indeed, our approach completely disregards the content. For the same reason, it is fast and even a modest hardware can process, on the fly, all Twitter messages about a popular topic like soccer.

This opens up many applicative perspectives. Indeed, knowing the preferences and inclinations of a user is a key to offer her personalized (hence better) content, services, recommendations, advertisement, etc. We hope our work motivates researchers to use the temporal dimension to build and design applications that use it to customize and enhance user experience in social media. For example, a possible application is to design a real-time recommendation algorithm that recommends content in real-time, or recommends users that tend to post at the same time. From a sociologist point of view, this information is valuable as well. It helps in grasping a collective view of the users over polemic topic such as politics and sports.

Another application of our results is to combine the temporal patterns we found with other evidences of user's preferences, such as social ties and textual content, what could generate better results than when these characteristics are considered in isolation.

As future work, we plan to investigate whether a sophistication of the time interval selection (using, e. g., activity peak detection) may provide more accurate results. If so, we could not only better learn the user biases but also better understand how microblogging platforms are used. Another exciting research direction is that of semi-supervised and active learning. The former approach would use the biases of *all* users in every time interval, whereas the latter would automatically populate the set of users with known biases (i. e., that are used to derive the biases of the other users). Finally, the temporal locality only is one evidence of the bias. Combining it with other evidences, such as endorsements – retweets – and adoption of tags, could provide greater recalls and, probably, more accurate results.

REFERENCES

Adamic, L. A. and Glance, N. The political blogosphere and the 2004 u.s. election: divided they blog. In *Proceedings of the International Workshop on Link Discovery*. Chicago, Illinois, pp. 36–43, 2005.

Azzini, DellAnna, R., Ciocchetta, F., Sboner, F. D. A., Blanzieri, E., and Malossini, A. Simple methods for peak detection in time series microarray data. In *Proceedings of Critical Assessment of Microarray Data*, 2004.

Earl, J., Martin, A., McCarthy, J. D., and Soule, S. A. The use of newspaper data in the study of collective action. vol. 30, pp. 65–80, 2004.

Gamon, M., Basu, S., Belenko, D., Fisher, D., Hurst, M., and König, A. C. Blews: Using blogs to provide context for news articles. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*. Seattle, WA, USA, 2008.

Gray, J. The transaction concept: virtues and limitations (invited paper). In *Proceedings of the International Conference on Very Large Data Bases*. Cannes, France, pp. 144–154, 1981.

Guerra, P. H. C., Veloso, A., Meira, Jr, W., and Almeida, V. From bias to opinion: A transfer-learning approach to real-time sentiment analysis. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. San Diego, CA, USA, 2011.

Hofgesang, P. I. Web personalisation through incremental individual profiling and support-based user segmentation. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*. IEEE Computer Society, Washington, DC, USA, pp. 213–220, 2007.

Kamath, K. Y. and Caverlee, J. Transient crowd discovery on the real-time social web. In *Proceedings of the ACM International Conference on Web search and data mining*. Hong Kong, China, pp. 585–594, 2011.

Kienpointner, M. and Kindt, W. On the problem of bias in political argumentation : an investigation into discussions about political asylum in germany and austria. *Journal of Pragmatics* 5 (27): 555–585, 1997.

Kwak, H., Lee, C., Park, H., and Moon, S. What is twitter, a social network or a news media? In *Proceedings of the International Conference on World wide web*. Raleigh, North Carolina, USA, pp. 591–600, 2010.

Ling, C. and Sheng, V. Cost-sensitive learning and the class imbalanced problem. In *Encyclopedia of Machine Learning*. Springer, 2007.

Liu, B. Sentiment analysis: A multi-faceted problem. In *IEEE Intelligent Systems*, 2010.

Milyo, J. and Groseclose, T. A measure of media bias. Working Papers 0501, Department of Economics, University of Missouri. Jan., 2005.

Mustafaraj, E. and Metaxas, P. From obscurity to prominence in minutes: Political speech and real-time search. In *Proceedings of the WebSci'10: Extending the Frontiers of Society On-Line*. Raleigh, NC, USA, 2010.

Pang, B. and Lee, L. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval* 2 (1-2): 1–135, 2008.

Phelan, O., McCarthy, K., and Smyth, B. Using twitter to recommend real-time topical news. In *Proceedings of the third ACM Conference on Recommender systems*. New York, New York, USA, pp. 385–388, 2009.

Rocha, L., Mourão, F., Pereira, A., Gonçalves, M. A., and Meira, Jr., W. Exploiting temporal contexts in text classification. In *Proceedings of the ACM Conference on Information and Knowledge Management*. Napa Valley, California, USA, pp. 243–252, 2008.

Sakaki, T., Okazaki, M., and Matsuo, Y. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the International Conference on World wide web*. Raleigh, North Carolina, USA, pp. 851–860, 2010.

Salles, T., Rocha, L., Pappa, G. L., Mourão, F., Meira, Jr, W., and Gonçalves, M. Temporally-aware algorithms for document classification. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. Geneva, Switzerland, pp. 307–314, 2010.

Skiena, S. S. *The algorithm design manual*. Springer-Verlag New York, Inc., 1998.

Turney, P. D. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the Annual Meeting on Assoc. for Computational Linguistics*. Philadelphia, Pennsylvania, pp. 417–424, 2002.

Walton, D. Bias, critical doubt, and fallacies. Number 28. Argumentation and Advocacy, pp. 1–22, 1991.

Widmer, G. and Kubat, M. Learning in the presence of concept drift and hidden contexts. *Mach. Learn.* 23 (1): 69–101, 1996.

Yang, J. and Leskovec, J. Patterns of temporal variation in online media. In *Proceedings of the fourth ACM International Conference on Web search and data mining*. Hong Kong, China, pp. 177–186, 2011.