Nwala et al. *EPJ Data Science* (2023) 12:33
https://doi.org/10.1140/epjds/s13688-023-00410-9

EPJ Data Science
a SpringerOpen Journal

EPJ.org

REGULAR ARTICLE Open Access

A language framework for modeling social media account behavior

Alexander C. Nwala[1,2*], Alessandro Flammini[1] and Filippo Menczer[1]

*Correspondence:
acnwala@wm.edu
[1]Observatory on Social Media, Indiana University, Bloomington, Indiana, USA
[2]*Present address:* William & Mary, Williamsburg, Virginia, USA

Abstract

Malicious actors exploit social media to inflate stock prices, sway elections, spread misinformation, and sow discord. To these ends, they employ tactics that include the use of inauthentic accounts and campaigns. Methods to detect these abuses currently rely on features specifically designed to target suspicious behaviors. However, the effectiveness of these methods decays as malicious behaviors evolve. To address this challenge, we propose a language framework for modeling social media account behaviors. Words in this framework, called BLOC, consist of symbols drawn from distinct alphabets representing user actions and content. Languages from the framework are highly flexible and can be applied to model a broad spectrum of legitimate and suspicious online behaviors without extensive fine-tuning. Using BLOC to represent the behaviors of Twitter accounts, we achieve performance comparable to or better than state-of-the-art methods in the detection of social bots and coordinated inauthentic behavior.

Keywords: Social media; Encoding online behavior; Bot detection; Coordination detection

1 Introduction

The widespread use of social media makes them a prime target for exploitation by bad actors. Efforts to inflate the popularity of political candidates [1] with social bots [2], influence public opinion through the spread of disinformation and conspiracy theories [3, 4], and manipulate stock prices through coordinated campaigns [5, 6] have been widely reported. The threats posed by malicious actors are far-reaching, endangering democracy [7, 8], public health [9–11], and the economy [12]. In response, researchers have developed various tools to detect malicious inauthentic accounts.

However, we are in an arms race. With new detection methods and prevention mechanisms from platforms, malicious actors continue to evolve their behaviors to evade detection. For example, consider the evolution of social bots: in the early days, spam bots were easy to identify because they often lacked meaningful profile information and/or demonstrated naive behaviors [13, 14]. In recent years, bot accounts have become more sophisticated. Some display detailed profiles, either stolen from other users or generated by deep neural networks [15]. Some mimic human actions and build social connections [16].

Springer

Figure 1 BLOC strings for a sequence of three tweets (a reply, an original tweet, and a retweet) by the `@NASA` account. Using the *action* alphabet, the sequence can be represented by three word *p.T.r* separated by dots. Using the *content* alphabet, it can be represented by these three words (*Emt*)(*mmt*)(*mmmmmUt*) enclosed in parentheses. See Sect. 3 for details

Others adopt strategies such as *coordinated inauthentic behaviors*.[1] Such coordinated behaviors appear to be normal when inspected individually, but are centrally controlled to achieve some goal [6].

The arms race has spawned a series of more complex detection methods [6, 16, 17]. An important limitation of these methods is that they rely on features crafted specifically to target previously observed malicious behaviors [18]. These features may not generalize well to other suspicious behaviors. For example, methods designed to detect sophisticated social bots tend to overlook coordinated behaviors, and vice versa [19]. Existing methods also become less useful when facing novel malicious actors, unless the features are adjusted accordingly.

To address this challenge, we propose a framework of Behavioral Languages for Online Characterization (BLOC), designed to represent social media account behaviors. Not to be confused with neural network-based language models, the BLOC framework involves formal languages specified by sets of rules for generating strings of symbols that describe online behaviors. BLOC words consist of symbols drawn from distinct alphabets representing an account's actions and content. As an example, Fig. 1 illustrates possible representations of a sequence of tweets by the official Twitter handle for NASA. BLOC languages are highly flexible in that they can represent a broad spectrum of legitimate and suspicious behaviors without extensive fine-tuning. In this paper we show that meaningful behavioral patterns emerge from such representations, facilitating tasks related to the classification of social media accounts.

To demonstrate the effectiveness of the BLOC framework, we evaluate it on social bot and coordinated behavior detection tasks, together with previous methods specifically designed for each of the two tasks. To the best of our knowledge, BLOC is the only representation framework that has been applied to both tasks. Although methods based on BLOC use significantly fewer features than state-of-the-art methods —making them much more efficient— they yield better or comparable performance.

[1] about.fb.com/news/2018/12/inside-feed-coordinated-inauthentic-behavior.

2 Related work

We can think of at least two dimensions to characterize inauthentic online behaviors: automation and coordination. Accounts could be automated but independent, or coordinated but closely managed by humans, or both automated and coordinated, and everything in between. Below we outline research aimed to detect inauthentic behaviors along these dimensions. Note that not all automated or coordinated behavior is necessarily inauthentic or malicious. For example, some self-declared bots are harmless or even useful; and some grassroots campaigns may use coordination to promote beneficial social movements.

2.1 Automation

The behavioral spectrum of social media account automation has human behavior at one end and bot-like behavior at the opposite end. Somewhere in between are "cyborgs" [20, 21], accounts that cycle between human and bot-like behaviors.

Various machine-learning methods have been proposed for identifying specific kinds of automated behavior. These methods typically utilize some combination of features such as social network structure, content/profile characteristics, and temporal patterns [2].

Multiple researchers have characterized authentic human behaviors online in ways that can inform the design of methods to distinguish them from behaviors generated by automated, algorithm-driven accounts. Wood-Doughty et al. studied one million accounts to explore how different demographic groups used Twitter [22]. This was based on the assumption that user behavior is reflected by indicators such as profile personalization, temporal information, location sharing, user interaction, and devices. He et al. provided a method for identifying five classes of behaviors on Twitter: individual, newsworthy information dissemination, advertising and promotion, automatic/robotic, and other activities [23]. Researchers have also studied human behavior across other social media platforms. Maia et al. represented YouTube users as feature vectors over a vocabulary consisting of number of uploads, videos viewed, channels visited, system join date, age, and so on [24]. They then clustered the users into predefined profiles such as small community member, content producer, and content consumer. Benevenuto et al. studied the online behavior of over 37 thousand users who accessed four social networks (Orkut, MySpace, Hi5, and LinkedIn) by analyzing their clickstream data [25].

On the other end of the automation spectrum are social bots [2]. A common theme of the literature is to build algorithms to distinguish bot-like and human accounts [16], which requires representing the account characteristics first. The rich information obtained from social media platforms makes it possible to describe accounts along many different dimensions. Depending on the types of the target accounts, existing methods use profile information [26], content [13, 27, 28], actions [17], social network [29], and temporal signatures [17].

Another common approach is to combine account characteristics from different dimensions in the same model [14, 18, 19, 30–32]. Botometer,[2] for example, is a publicly available supervised machine learning system that extracts over 1000 features from a Twitter account's profile, content, sentiment, social network, and temporal activity.

[2] botometer.org.

Digital DNA (DDNA), proposed by Cresci et al. [27, 28], is the most similar method to BLOC. DDNA encodes each account as a pair of strings of symbols representing actions and content, respectively. It then considers accounts with long common substrings as bots. While BLOC languages similarly use sequences of symbols to encode actions and types of content, they differ significantly from DDNA conceptually and technically.

Conceptually, DDNA was designed for characterizing behaviors, especially automated ones. In contrast, the focus of BLOC is broader: modeling behaviors irrespective of the class of the account (bot, cyborg, or human) or intent (malicious or benign). This means BLOC can be applied to multiple tasks such as individual/group characterization, bot, and coordination detection as demonstrated in this paper.

There are also two crucial technical differences between BLOC from DDNA. The first has to do with language features capturing pauses and repetitions (see Sect. 3.1). Pauses enable studying a variety of behaviors (e.g., repetitive, dynamic, and bursty). For example, the absence of long pauses could be revealing of automated behaviors. Additionally, DDNA truncates repeated content characters, whereas BLOC content words can capture repetitions to emphasize different behaviors. Repetitions provide a means to weigh and identify important signals, which is crucial since not all aspects of behaviors are expressed equally. This can help identify accounts engaging in repetitive behaviors, such as long sequences of retweets, typical of certain inauthentic accounts.

The second important difference is between the signature-based method of DDNA and the vector-based approach of BLOC (see Sect. 3.2). BLOC languages represent behaviors with words that may capture distinct behavioral patterns. Accounts are thus represented as word vectors, allowing for similarity measures beyond string matching. Word representations are also important since accounts can change behaviors, as illustrated by the cyborg account in Fig. 2. Changes in behavior map to changes in BLOC words. This representation provides opportunities to study when accounts have been hijacked or repurposed.

2.2 Coordination

Malicious social bots evolve in sophistication over time, making them more effective and harder to detect. In some cases, it is not sufficient to study individual accounts. A group of inauthentic accounts can be coordinated by a single entity, whether their behavior are human-controlled or automated. These kinds of sophisticated deception can only be detected through observations at the group level [16]. This has led to multiple research efforts to detect malicious coordinated behaviors.

While individual bot detection aims to separate individual human and bot-like accounts, coordination detection involves clustering suspiciously similar accounts into groups [6]. Appropriate definitions of similarity measures are subjective and vary across different studies. A common choice is to focus on the temporal dimension, with the action time series of different accounts compared directly [33, 34] or modeled using temporal point processes [35]. Other similarity measures focus on duplicated or partially matched text [36, 37] or on shared retweets [38]. Some methods focus on specific components of the content, such as embedded links, hashtags, and media [6, 37, 39–41]. Account profile information can also be used to identify similar accounts [42]. Finally, it is possible to aggregate similarity measures based on different criteria [43].

These methods typically extract account features designed to target specific suspicious behavioral patterns [6]. The BLOC framework encodes behavioral information into features that can be used to calculate similarities without a predefined target behavior. As a

Table 1 BLOC language parameters

Param.	Context	Explanation	Values
p_1	Pauses	Session delimiter threshold	Time
p_2	Pauses	Time granularity	$f_1(\Delta)$ or $f_2(\Delta)$
p_3	Word	Use sessions for content words	Yes or No
p_4	Word	Tokenization	N-gram or pause
p_5	Word	Sort symbols	Yes or No
p_6	Word	Word truncation length	Integer

result, BLOC languages are versatile and can be applied to characterize a broad spectrum of behaviors. We next provide an in-depth introduction to the BLOC framework.

3 Behavioral languages for online characterization

The central component of BLOC is a collection of two alphabets: *actions* and *content*. Each consists of a set of symbols that represent activities or traits. Collectively, these alphabets encode behaviors that can be utilized to build models for various tasks and platforms. The BLOC framework is in fact platform agnostic; similar alphabets can be easily adapted to, say, Instagram, Facebook, and TikTok.

BLOC languages have several parameters, shown in Table 1. Different combinations of values for these parameters correspond to different languages and representations. Below we discuss these parameters in detail, noting recommended values based on extensive experiments. In Sects. 4 and 5 we apply different BLOC representations to various tasks.

3.1 BLOC alphabets

Let us illustrate how to generate BLOC strings drawn from the alphabets for an arbitrary Twitter account `@Alice`. (Note that all symbols below could also be used for Instagram accounts, excluding the action symbol for resharing one's own posts, since this action is not supported on Instagram.)

3.1.1 Action alphabet

The *action* alphabet includes two sets of *action* and *pause* symbols. An action symbol characterizes a single post by an account with a symbol as outlined below:

T: Post message
P: Reply to friend
p: Reply to non-friend
π: Reply to own post
R: Reshare friend's post
r: Reshare non-friend's post
ρ: Reshare own post

For example, the string $Tp\pi R$ indicates that `@Alice` posted a tweet, then replied to a non-friend, followed by a reply to herself, and finally retweeted a friend.

The *pause* symbols characterize the pauses between consecutive actions. Pauses provide additional context for actions. For example, actions taken with very short (e.g., less than a second) or highly regular pauses could indicate automation [44].

Let us first define Δ as the time between two consecutive actions. Based on parameter p_2, we have two possible pause alphabets defined by functions that map Δ values to

symbols. The function f_1 is defined as:

$$f_1(\Delta) = \begin{cases} \text{no symbol} & \text{if } \Delta < p_1 \\ . & \text{otherwise} \end{cases} \tag{1}$$

where p_1 is a *session* delimiter threshold. A session is thus defined as a maximal sequence of consecutive actions separated by pauses shorter than p_1. Sessions are important because they provide natural word boundaries for tokenizing BLOC words (see Sect. 3.2). We recommend using a value of a minute or less for p_1 in Eq. (1).

As an illustration, let us punctuate `@Alice`'s string of actions ($Tp\pi R$) with pause symbols using f_1 and p_1 = 1 minute. Say that Alice pauses 2.5 minutes between the first tweet and the reply to a non-friend, then 50 seconds pass until her self-reply, and finally she waits 3 days before the final friend retweet. The resulting BLOC string would be $T.p\pi.R$, indicating three sessions whose boundaries are marked by the dots.

An alternative pause alphabet assigns different symbols to long pauses for better granularity. We discretize time into a logarithmic scale to represent a wide range of pauses, e.g., hours vs. days vs. weeks, by defining f_2 as:

$$f_2(\Delta) = \begin{cases} \text{no symbol} & \text{if } \Delta < p_1 \\ t_h & \text{if } p_1 \leq \Delta < 1 \text{ hour} \\ t_d & \text{if 1 hour} \leq \Delta < 1 \text{ day} \\ t_w & \text{if 1 day} \leq \Delta < 1 \text{ week} \\ t_m & \text{if 1 week} \leq \Delta < 1 \text{ month} \\ t_y & \text{if 1 month} \leq \Delta < 1 \text{ year} \\ t_z & \text{otherwise.} \end{cases} \tag{2}$$

Using the same example as above, `@Alice`'s string of actions using the f_2 pause symbols with p_1 = 1 minute would be $Tt_hp\pi\, t_wR$.

3.1.2 Content alphabets

The *content* alphabet provides a lexical characterization of a post — whether it contains text, links, hashtags, and so on. Unlike the *action* alphabet, a single social media post can contain multiple *content* symbols from the following list:

- t: Text
- H: Hashtag
- M: Mention of friend
- m: Mention of non-friend
- q: Quote of other's post
- ϕ: Quote of own post
- E: Media object (e.g., image/video)
- U: link (URL)

As an illustration, let us imagine that `@Alice`'s first tweet only contains text; her reply to a non-friend has two images and one hashtag; her self-reply mentions one friend and has one link; and finally she retweets a post that mentions a non-friend. The resulting content string depends on the p_3 parameter. If sessions are not used, each action corresponds to a

separate content word: $(t)(EEH)(UM)(m)$. Here the contents of the reply to a non-friend (EEH) and of the self-reply (UM) are separated, even though they were part of the same session. Using sessions, we get $(t)(EEHUM)(m)$. Note that parentheses separate content words, and the order of content symbols within a word is arbitrary and defined in the implementation.

3.2 BLOC vector models

A flexible representation used in many machine-learning tasks, including online bot detection and coordination detection, is obtained by mapping each data point (an account or behavior in our setting) to a point in a vector space. There are multiple ways to generate vector representations from BLOC strings. One approach would be to train a deep-learning model [45] such as word2vec [46] and embed BLOC words or sentences into dense feature vectors. However, such an abstract vector space would fail to benefit from the interpretability of BLOC symbols.

Alternatively, we can obtain a vector representation by first tokenizing BLOC strings into words and then using these words directly as vector space dimensions. Tokenization can be done using one of two methods, *n-gram* or *pause*, based on parameter p_4 (Table 1). The n-gram method generates tokens of fixed size n by sliding an n-sized window over the BLOC string. Using $n = 2$, we generate bi-grams resulting in a vocabulary of two-symbol words. For example, given the action string $Tp\pi.r$ and the BLOC content string $(t)(EH)(U)(mm)$ with $n = 2$, we obtain the set of words $\{Tp, p\pi, \pi., .r, tE, EH, HU, Um, mm\}$.

The pause method uses pauses to break BLOC action strings into words of variable length. In addition to serving as word boundary markers, pause symbols are included in the vocabulary as single-symbol words. For content strings, individual posts mark word boundaries: all symbols in the same post form a single word. The symbols within each word may be sorted alphabetically depending on parameter p_5. To illustrate pause tokenization without sorting, given the same BLOC action string $Tp\pi.r$ and BLOC content string $(t)(EH)(U)(mm)$, we obtain the set of words $\{Tp\pi, \cdot, r, t, EH, U, mm\}$.

Pause tokenization often results in long words, for example, the 13-symbol word $\pi\pi\pi\pi\pi\pi TT\pi\pi\pi\pi\pi$ from the *cyborg* account in Fig. 2. Long words occur when the pauses between multiple consecutive actions are shorter than p_1, meaning that actions are performed in bursts, which often indicates automation. The distinction between, for example, *rrrr* and *rrrrr* is often not important, so instead of representing both as separate words in our vocabulary, we could truncate long words after a limit. For example, setting $p_6 = 4$ would truncate characters that repeat four or more times. The words *rrrr*, *rrrrr*, and *rrrrrr* would all be replaced by *rrr+*.

After tokenization, we can represent any account as a vector of BLOC words. In a vector model, each account is represented as a point $(w_1, w_2, \ldots, w_k)$ in a k-dimensional vector space where each dimension i corresponds to a word. We wish to define a weight w_i that represents how well an account is described by i. The number of times f_i that word i occurs in the BLOC representation of the account, known as term frequency (TF), is not very discriminative because some words, such as t (text), may be common across all accounts. Therefore the term frequency is multiplied by a second factor, called inverse document frequency (IDF), that captures how rare a word is across accounts. We use the TF-IDF

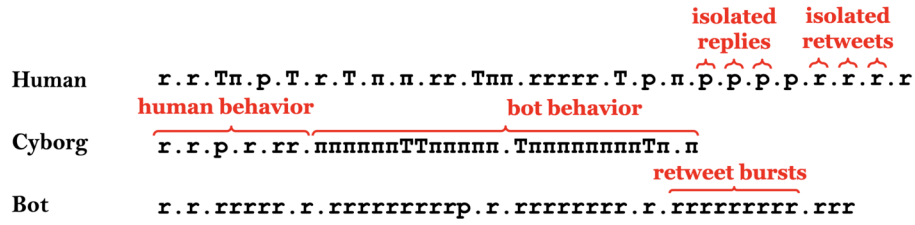

Figure 2 Illustrations of BLOC action strings (p_1 = 1 minute) for a human, a cyborg, and a bot Twitter account illustrating some behavioral differences across these individuals. If strings are tokenized using pauses, the human account has the shortest words (average length 1.35 vs. 3.88 for the cyborg and 4.0 for the bot) and is dominated by isolated retweets and replies. The cyborg account — which we created to post threads of news updates — exhibits both human (isolated posts) and bot behavior (thread bursts). The bot account mainly generates retweet bursts

weight [47] for account a defined as follows:

$$w_i(a) = f_i(a)\left(1 + \log \frac{D}{d_i}\right) \tag{3}$$

where d_i is the number of accounts with word i and D is the total number of accounts. Finally, the vectors can be used to build bot or coordination detection systems.

4 Discriminative power of BLOC

The BLOC framework lets us study behaviors at different levels of granularity. We may study different classes of accounts, such as humans vs. bots. Or we might study different types of individual accounts within a class, for instance, a political vs. an academic human account or a spam bot vs. a self-declared bot. In this section we demonstrate such a multi-resolution approach by characterizing the behavior of individual accounts and groups of accounts, both when their class labels are known and unknown.

4.1 Characterizing individuals and groups

Figure 2 illustrates the behavioral differences between three individual accounts: a human account belonging to a journalist; a cyborg account used by one of the authors to post news updates, either manually or using a software script; and a spambot account identified by Mazza et al. [17]. These accounts are represented by their respective BLOC action strings. We observe multiple differences. First, when we tokenize the strings into words separated by pauses, the human account has the shortest words, with mostly one-symbol words (e.g., r, T, p). This captures the fact that humans tend to rest between posts. Second, the cyborg account exhibits a human substring with shorter words, followed by a bot substring created in a burst. Third, the bot account tends to amplify content with retweet bursts (e.g., *rrrrrrrr*) rather than creating new content.

Let us shift our focus to studying groups of accounts. Figure 3 presents a Principal Component Analysis (PCA) of the BLOC TF-IDF vectors of equal numbers of bot and human accounts from six different datasets (see Table 2). We observe that the bot and human accounts in the left column of the figure express more distinct behavioral patterns than those in the right column. Consequently, accounts in the left column have fewer words in common and are easier to separate. For example, while both bot and human accounts in Fig. 3A tweet text-only (t) content, the bot accounts more often include hashtags (Ht). In

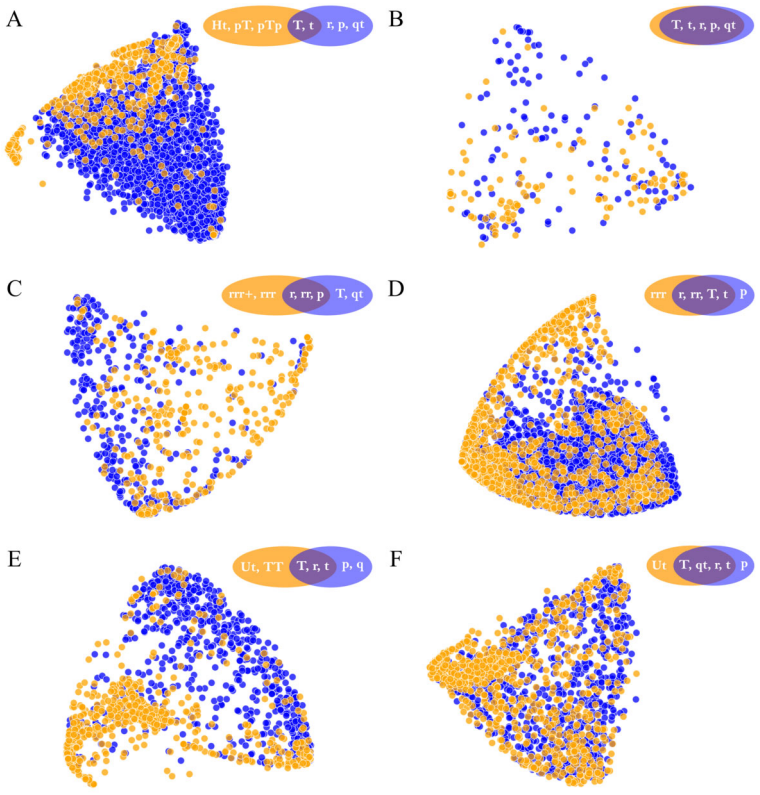

Figure 3 Two-dimensional PCA projections of BLOC TF-IDF vectors of accounts from six datasets that include both humans and bots (see Table 2): (**A**) cresci-17, (**B**) botometer-feedback-19, (**C**) cresci-rtbust-19, (**D**) cresci-stock-18, (**E**) varol-17, and (**F**) gilani-17. From each of these datasets, we select an equal number of bot (orange) and human (blue) accounts. We use all the accounts in the minority class, and sample an equal number of accounts from the majority class. The Venn diagrams show the top five pause-delimited BLOC words for the bot and human accounts shown

Table 2 Annotated datasets used in our bot detection evaluation. For each dataset, we report the reference describing it and the number of accounts that are still active at the time of the present evaluation

Dataset	Ref.	# Bots	# Humans
astroturf-20	[18]	505	0
botometer-feedback-19	[19]	123	364
botwiki-19	[26]	695	0
celebrity-19	[19]	0	20,911
cresci-17	[48]	5812	2744
cresci-rtbust-19	[17]	352	340
cresci-stock-18	[49]	6926	6155
gilani-17	[32]	1058	1381
midterm-18	[26]	0	7409
political-bots-19	[19]	62	0
pronbots-19	[19]	14,867	0
varol-17	[31]	728	1483
vendor-purchased-19	[19]	928	0
verified-19	[26]	0	1986
Total	–	32,056	42,773

Fig. 3C, bots amplify content with burst of retweets (*rrr*, *rrr+*) unlike humans who create original content (*T*). In Fig. 3E, bots share more external links (*U*) while humans tend to engage in conversations and commentary (*p*, *q*).

In Fig. 3B, bots and humans express similar behavioral traits: both classes have the same five top words. In Fig. 3D and F, bots and humans share four of their five top words. The bot accounts are more likely to amplify content (*rrr*) and link to external websites (*Ut*) in Figs. 3D and F, respectively, while their corresponding human accounts are more likely to engage in conversations (*p*). In summary, the figure suggests that the behaviors displayed by the humans tend to be consistent across datasets, whereas the bots have distinct behaviors based on the purpose for which they have been created. These findings are consistent with prior analysis based on ad-hoc features [18]. The BLOC representation is sufficiently powerful to capture significant differences between these behaviors.

4.2 Behavioral clusters

When behavioral class labels are unavailable, we can characterize online behaviors in an unsupervised way, using BLOC representations to cluster accounts according to behavioral similarity.

We analyzed tweets collected between January 4 and September 30, 2021 from the CoVaxxy project,[3] which studies how online misinformation impacts COVID-19 vaccine uptake [11]. The dataset [50] consists of over 200 million English-language tweets about COVID-19 and vaccines, posted by over 17 million accounts. The tweets were collected with 76 keywords and hashtags covering a variety of neutral (e.g., *covid*), pro-vaccine (e.g., *getvaccinated*), anti-vaccine (e.g., *mybodymychoice*), and conspiratorial (e.g., *greatreset*) topics.

Given the large number of accounts present in the dataset and the quadratic cost of pairwise comparison, we focused on the one thousand most active accounts each month. We based our definition of activity on the number of days in which an account posted tweets; to break ties (especially for accounts active every day), we used the total number of tweets an account posted during the collection period.

We applied a three-step, network-based method to identify clusters of accounts with highly similar behaviors. First, we generated BLOC TF-IDF vectors for each account using pauses to tokenize words, without sorting symbols, and truncating words ($p_6 = 4$). Second, we computed the cosine similarities among the 1000 vectors. We built a network by linking only nodes (accounts) with similarity of at least 0.98 and removing singletons. This threshold ensures a focus on accounts with a suspiciously high level of similarity. Third, we applied the Louvain method to identify communities [51].

This procedure was applied every month (January – September) to produce nine behavioral similarity networks consisting of clusters of accounts with highly similar behaviors. Figure 4 visualizes 24 of the 163 identified clusters. In the figure, a single dot represents a cluster positioned on axes representing its *mean variety of behavior* and *mean automation* score. For a single account, we measured its variety of behavior by the entropy of its BLOC string (before tokenization). We estimated account automation by the fraction of times the account posted using the Twitter API. A user has to create an app in order to use the Twitter API, and Twitter data includes a "user-agent" that identifies the app. Some user-agent

[3] osome.iu.edu/tools/covaxxy.

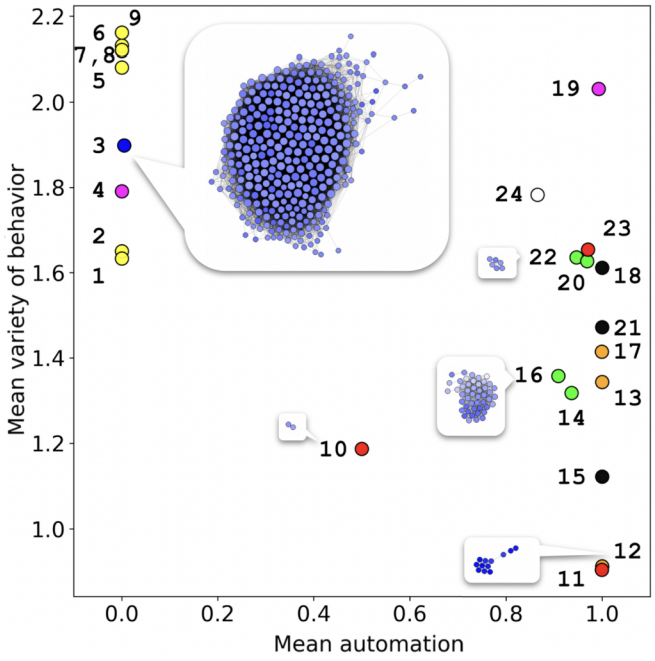

Figure 4 Mean variety of behavior vs. mean automation (see text) for 24 communities of accounts with highly similar behaviors. Each community is represented by a dot, colored according to manual classification (see text). A few selected communities are highlighted by visualizing the corresponding subnetworks, with node size and darker color representing degree and tweet count, respectively

values correspond to Twitter native apps (`TweetDeck`, `Twitter for Advertisers`, `Twitter for Advertisers (legacy)`, `Twitter for Android`, `Twitter for iPad`, `Twitter for iPhone`, `Twitter for Mac`, `Twitter Media Studio`, `Twitter Web App`, and `Twitter Web Client`). While software could in principle be written to control native apps, we assume the vast majority of these apps are operated manually. Similarly, we assume non-native apps indicate the use of the Twitter API and thus likely automation, even though some could be operated manually. The entropy and automation scores are averaged across the accounts in each cluster. The clusters in Fig. 4 are well separated along the automation axis, suggesting a robust distinction between human and bot accounts.

We manually inspected the clusters in Fig. 4 to describe the dominant behaviors, summarized in the groups below. Each cluster number has a suffix indicating the month when it was observed. All the clusters in each group have the same color in Fig. 4.

- **Giant connected component (blue): Cluster 3-Sep** includes accounts with low automation scores and high variety of behaviors. These are likely legitimate users who mostly retweet and occasionally post tweets, with normal pauses. Similar large components were present on each month.
- **Vaccine availability/appointment bots (orange): Cluster 12-Apr** includes 12 self-identified bot accounts that track the availability of vaccines and appointments in various US cities, such as `@DCVaxAlerts` and `@FindAVac_Austin`. These accounts posted messages such as "New available appointments detected! – Provider: CVS Pharmacy – City: Alamo Heights – Registration link: www.cvs.com/immunizations/covid-19-vaccine..." They created long bursts of tweets

consisting mostly of URLs and text. Overall, these accounts posted the most content. Similarly, **Cluster 17-Jan** includes two vaccine appointment bots (`@kcvaccinewatch` and `@stlvaccinewatch`) that created tweet threads. **Cluster 13-Jul** includes `@CovidvaxDEL`, a vaccine appointment status bot for New Delhi, India; and `@ncovtrack`, a bot that posted vaccine statistics for various countries.

- **News posting accounts (green): Clusters 14-Apr**, **16-Jan**, **20-Apr** and **22-Feb** include many accounts that mostly post tweets linking to news websites hourly, such as `@canada4news` and `@HindustanTimes`. Some accounts are owned by international news organizations such as `@Independent` and `@guardian`.
- **Content amplifying, likely bot accounts (purple): Cluster 4-May** includes a pair of accounts that create no content; they retweet mostly the same tweets repeatedly. **Cluster 19-May** includes self-identified bots created by the same self-identified developer. These bots, `@EdinburghWatch` and `Glasgow_Watch`, retweet random content from Glasgow and Edinburgh, respectively.
- **Misinformation sharing and local news accounts (white): Cluster 24-Feb** includes `@USSANews` owned by `ussanews.com`, a misinformation website according to `factcheck.org`. This account posted links with headlines such as: "31 Reasons Why I Won't Take the Vaccine." The same cluster includes `@abc7newsbayarea`, the account of a legitimate local news organization. Both accounts mostly post multiple tweets with images separated by pauses under an hour.
- **Spam bots (red): Clusters 10-Mar**, **11-Apr**, and **23-Aug** include accounts that post repeated content. The accounts in **Cluster 10-Mar** repeatedly linked to their respective blogs with exactly seven or thirteen hashtags. **Cluster 11-Apr** posted messages soliciting others to follow a specified account. The two accounts in **Cluster 23-Aug** posted the same pro-vaccine messages repeatedly, 133 and 72 times respectively.
- **Coordinated bots (black):** The three accounts in **Cluster 21-May** created no content; they retweeted the same account exactly 1004 times each. During the first week of May 2021, the first 44 characters of their BLOC strings matched. Similarly, accounts in **Cluster 15-May** did not create content but always retweeted the same collection of multiple business accounts advertising various merchandise. **Cluster 18-Mar** includes a pair of accounts that retweeted one another 313 times.
- **Various low automation accounts with different stances on vaccine (yellow):** Finally, Fig. 4 also features clusters of accounts with pro-vaccine (**Clusters 1-May** and **2-Jan**), anti-vaccine (**Clusters 5-Mar**, **6-Apr**, **7-Mar**, and **8-May**), or a mixture of both sentiments (**Cluster 9-Jun**).

5 Evaluation

In this section we evaluate the performance of BLOC models on bot and coordination detection tasks on Twitter. BLOC code and datasets used in our experiments are available [52].

5.1 Bot detection

The bot detection task involves separating accounts that are likely operated by human users from accounts that are likely automated. This is a challenging task, as behaviors of both classes of accounts are heterogeneous and time-evolving.

5.1.1 Methods

The BLOC language parameters used for the evaluation are as follows: p_1 = 1 minute, $p_2 = f_2(\Delta)$, and p_4 = bi-gram (Table 1). The other parameters are not applicable to bi-gram tokenization. We extracted BLOC *action* and *content* bi-grams for each annotated Twitter account. This resulted in a set of 197 bi-grams. These bi-grams can be used as features within any machine learning model. We obtained TF-IDF feature vectors for each account and used them to train a random-forest classifier.

We compared the performance of the BLOC model to three baseline models: Botometer-V4 (the current version of Botometer at the time of writing) [18] and two DNA-based methods, namely DDNA [27, 28] and DNA-influenced [53]. The latter were selected because they share some similarities with BLOC.

Botometer-V4 utilizes 1161 different features that can be grouped into six categories that focus on different account characteristics. For example, user profile features are extracted from the user profile, like the numbers of friends and followers. Temporal features measure temporal patterns of the posts, such as frequency and times of day. In the deployed system, different classifiers in an ensemble are trained on different accounts types, and then these classifiers vote to obtain the final bot score [18]. Here instead, to compare the representation power of BLOC vs. Botometer features with all other things being equal, we trained a single random-forest classifier with the same features used to train Botometer-V4.

Digital DNA classifies accounts as bots if they share long sequences of symbols representing actions and content. Cresci et al. [28] provided their Python code [54], which wraps a C implementation for the Longest Common Substring (LCS) algorithm. We modified the code to implement the method described by the authors. The method yields a maximal common substring length from the training data. This length is then used to determine a set of accounts in the test data that share a maximal common substring of the same length. These accounts are classified as bots. We finally apply cross-validation to evaluate the classifier.

The DNA-influenced bot classifier is based on the rationale that bot accounts are more likely to be similar to each other, compared to human accounts. The method relies on a formula to calculate a probability distribution for a given string, and on the *symmetrized KL divergence* to calculate the distance between the probability distributions associated with two strings [55]. In this way, the method calculates the distance between the DDNA strings corresponding to two accounts [53]. To implement this method, we partitioned the bot accounts in the training dataset into groups of 50, similar to Gilmary et al. [53]. For each group, we calculated the average distances across all pairs of accounts in the group. The maximum average distance across all the groups was then used as a decision threshold: any two accounts in the test dataset were classified as bots if their distance was less than or equal to the decision threshold.

5.1.2 Datasets

Our evaluation datasets (Table 2) consist of 32,056 Twitter accounts labeled as bots and 42,773 accounts labeled as humans, all selected from the bot repository.[4] These accounts were collected and labeled by multiple researchers between 2017–2019 [19]. To eliminate

[4] botometer.osome.iu.edu/bot-repository.

Table 3 Precision, recall, and F_1 for different bot classifiers using 5-fold cross-validation, along with numbers of features. The best values for each metric are shown in bold. DNA-influenced classifiers produced recall of 1.0 because they always predicted that all account were bots

Model	Precision	Recall	F_1	Features
BLOC	0.899	0.884	0.892	**197**
Botometer	**0.929**	0.914	**0.921**	1160
DNA-influenced	0.499	**1.000**	0.666	–
Digital DNA (*b3_type*)	0.796	0.529	0.636	–
Digital DNA (*b3_content*)	0.866	0.183	0.303	–
Digital DNA (*b6_content*)	0.868	0.187	0.308	–

a potential bias in the comparative analysis that might result from the class imbalance, we took the union of all datasets but used a random sample of 32,056 accounts from the majority class (humans).

5.1.3 Results

We evaluated the BLOC model, Botometer, three variants of Digital DNA (*b3_type*, *b3_content*, and *b6_content*) [28], and DNA-influenced by predicting bot and human labels, all on the same annotated dataset in Table 2. We computed precision, recall, and F_1 from 5-fold cross validation.

As reported in Table 3, Botometer-V4 slightly outperformed the BLOC model on the F_1 metric. However, the BLOC model used significantly fewer features. DNA-influenced outperformed Digital DNA, even though it labeled *all* accounts as bots.

5.2 Coordination detection

Multiple nation states utilize social media for information operations that target their citizens, foreign nationals, organizations, etc. Twitter defines information operations as a form of platform abuse, which involves artificial amplification or suppression of information or behavior that manipulates or disrupts the user experience.[5] Twitter deletes the public tweets of accounts engaged in information operations, but publishes datasets containing these tweets.

Let us use the term *driver* to refer to an account engaged in some information operation. Drivers may employ tactics such as spamming, impersonation, obfuscation, and/or targeting of individuals or communities. We consider all these behaviors *coordinated* but do not distinguish among them. Our task is to separate the drivers from regular (control) accounts tweeting about the same topics.

5.2.1 Methods

Coordination detection is typically based on unsupervised learning, namely, identifying clusters of accounts with suspiciously similar behaviors. Our coordination detection method and evaluation are more related to supervised learning, as described below. BLOC words express behavioral traits. We generated TF-IDF vectors as described in Sect. 5.1.1 and then calculated the similarity between two accounts via the cosine between their two vectors.

We compared the BLOC model to three baseline methods, which make different assumptions about the behavioral traits that may be shared among coordinated accounts [6]:

[5] help.twitter.com/en/rules-and-policies/platform-manipulation.

hashtag sequences (Hash), *activity* (Activity), and *co-retweet* (CoRT). The hashtag baseline method identifies coordinating accounts by finding those that mostly use the same sequences of hashtags (e.g., the same hashtag 5-grams). The activity method looks for accounts that are synchronized in the times when they post tweets: accounts that often tweet or retweet within the same time window are considered suspicious. Similar to Pacheco et. al. [6], we considered accounts that consistently posted tweets within 30-minutes from one another to be suspicious. The co-retweet method identifies coordinating accounts by finding those that mostly retweet the same sets of tweets. We generated TF-IDF vectors of hashtag 5-grams, of activity time intervals, and of retweeted tweet IDs, as described by Pacheco et. al. [6]. For all baselines, the cosine between TF-IDF vectors was used to calculate similarity.

We also evaluated a *combined* method. For a pair of accounts, the combined method takes the maximum among four cosine similarity values computed with the BLOC model and the three baselines.

We employed a leave-one-out classification in conjunction with k-nearest-neighbors (KNN) classifiers to label an account as a driver or non-driver. This approach is "supervised" in the sense that KNN infers the unknown label of an account using the known labels of its neighbors. Specifically, for the mixture of driver and control accounts, we computed pairwise cosine distance using their BLOC (or Hashtag, Activity, CoRT) vectors. For each account, we predicted its label to be the majority class from all its k nearest neighbors. We report the maximum F_1 obtained across k values ($k = 1, \ldots, 10$) to compare the five methods.

5.2.2 Datasets

Twitter published over 141 information operation datasets [56]. These datasets include tweets by drivers across 21 countries, during different time periods between 2008 and 2021. To ensure a fair assessment of the classifiers for detecting information operation drivers, we built control datasets that include tweets by accounts not engaged in information operations, but who posted about the same topics around the same time. For each information operation, we extracted all the hashtags used by the drivers. Then we used these hashtags as queries to Twitter's academic search API,[6] which does not impose date restrictions. We extracted accounts that posted tweets on the same dates and with the same hashtags as the drivers. Finally, for each of these accounts, we reconstructed their timelines by extracting a maximum of 100 tweets posted on the same dates as the drivers. We were able to create control datasets for 36 information operations, as shown in Table 4. These represent 18 of the countries and the entire time period.

Some information operations lasted a few months (e.g., China_3 in Table 4), others over five years (e.g., Iran_7 in Table 4). Therefore, we could run the experiment of detecting drivers for different time periods (e.g., first year, last year, all years). From the perspective of mitigation, we followed the principle that it is desirable to detect drivers as early as possible, with as little information (tweets) as possible. We believe it is more difficult to detect drivers early, since sufficient tweets with coordination signals might be absent.

Based on the above principle, we ran each experiment by incrementally adding two weeks of data until the end of the first year in which at least 10 drivers were observed,

[6] developer.twitter.com/en/products/twitter-api/academic-research.

Table 4 Selected information operation. We list life spans, the number of weeks used for evaluation (since the start of the information operations), and the counts of drivers and control accounts active during the evaluation weeks. Note that the evaluation weeks are not necessarily contiguous

Information Op.	Life span	Eval. week	# Drivers	# Control
Armenia	2014 – 2020	4	27	1462
Bangladesh	2010 – 2018	8	10	929
Catalonia	2011 – 2019	20	14	906
China_1	2008 – 2019	20	11	905
China_2	2008 – 2019	44	53	4019
China_3	2021	4	95	165
China_4	2020 – 2021	18	1623	4681
China_5	2021	10	1247	4890
Cuba	2010 – 2020	26	11	3415
Ecuador	2010 – 2019	22	10	1767
Egypt_UAE	2012 – 2019	40	59	350
Ghana_Nigeria	2014 – 2020	28	53	1102
Iran_1	2012 – 2018	32	23	824
Iran_2	2010 – 2015	22	11	407
Iran_3	2011 – 2020	20	14	791
Iran_4	2014 – 2019	38	13	2088
Iran_5	2013 – 2019	28	14	701
Iran_6	2020 – 2020	8	104	1247
Iran_7	2010 – 2020	52	16	11,842
Mexico_1	2019 – 2019	18	119	2097
Mexico_2	2020 – 2021	44	240	6340
Qatar	2013 – 2020	18	11	7393
Russia_1	2009 – 2018	4	12	498
Russia_2	2011 – 2018	2	10	168
Russia_3	2009 – 2020	6	10	5200
Russia_4	2014 – 2020	24	21	2973
Spain	2019 – 2019	8	215	1681
Thailand	2018 – 2020	8	166	1133
UAE	2011 – 2019	34	17	1662
Uganda_1	2019 – 2020	54	124	11,323
Uganda_2	2020 – 2021	54	342	10,526
Venezuela_1	2010 – 2018	28	48	891
Venezuela_2	2015 – 2018	8	71	995
Venezuela_3	2012 – 2019	16	17	1412
Venezuela_4	2020 – 2021	54	139	12,365
Venezuela_5	2021	22	249	5447
Total			5219	114,595

or the end of the campaign — whichever occurred first. In other words, the first instance of our experiment was run on two weeks of data, the second on four weeks of data, and so on. The use of increasing evaluation intervals is meant to explore how accuracy depends on the amount of data accumulated. For each coordination detection method, we generated vectors corresponding to all driver and control accounts active in each information operation and evaluation interval. Table 4 reports on the full evaluation periods and numbers of driver and control accounts in our datasets.

5.2.3 Results

Figure 5 plots the F_1 values of the best-performing classifiers for a subset of information operations. The best KNN classifier is the one with the k value ($k = 1, \ldots, 10$) yielding the maximum F_1. The x-axis for each plot represents the number of evaluation weeks, while the y-axis represents the F_1 score of the best classifier. The information operations are ordered in descending order of their respective *combined F1@Week 10* score, to capture

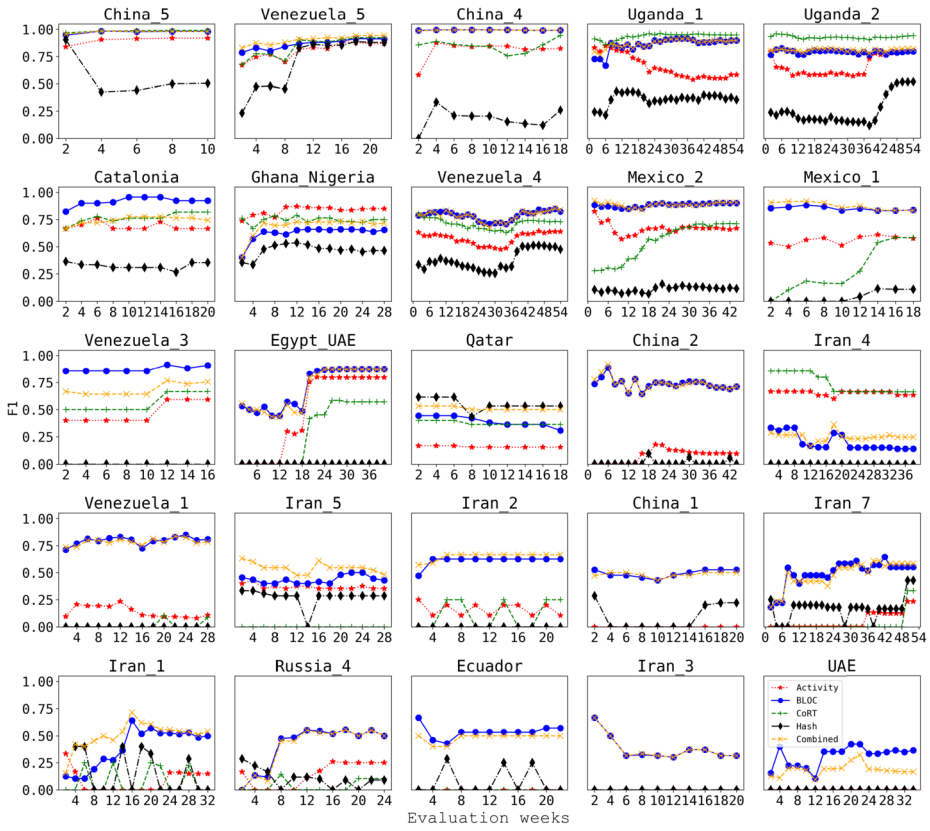

Figure 5 F_1 scores of the best-performing classifiers for detecting information operation drivers for the subset of campaigns with at least 10 weeks worth of data. The weeks displayed on the x-axis represent those in which the drivers were active (evaluation weeks); they are not necessarily contiguous. The plots are ordered in descending order of F_1 score calculated at week 10 using the combined method

the difficulty of detecting their drivers. The *combined F1@Week 10* score of an information operation is the F_1 score calculated with 10 weeks worth of data (*F1@Week 10*) using the combined method. Table 5 outlines the *F1@Week 10* scores for all information operations.

According to Fig. 5 and Table 5, the BLOC model outperforms the baselines in most campaigns. The drivers from information operations originating from China (e.g., China_4 and China_5) were the easiest to detect; the F_1 scores for all coordination detection methods except Hash were above 0.9. The hardest drivers to detect were those from the UAE information operation. We also note in Fig. 5 that in some campaigns (Venezuela_4, Venezuale_3, and Egypt_UAE), the accuracy of different methods improves in a correlated fashion with more training data. This suggests that drivers display multiple coordination signals simultaneously. Yet, having more data does not necessarily imply higher accuracy in detecting drivers. In several campaigns there is no clear temporal trend, and in a few cases (e.g., Iran_4 and Iran_3) adding more data hinders detection. This suggest that drivers may change their behaviors and become harder to detect as a result.

Figure 6 compares the performance of BLOC and the three baseline coordination detection methods. The x-axis represents the mean F_1 and the y-axis represents the mean number of features of all classifiers, across all information operations. Both values were calculated with data from the first 10 weeks of the information operations. The BLOC classifiers

Table 5 F_1 scores of BLOC and baseline classifiers for the detection of information operation drivers, calculated with data from the first 10 weeks of each campaign (*F1@Week 10*). For campaigns with less than 10 weeks of data, the entire dataset was used. Information operations are sorted by the F_1 score of the combined method (*combined F1@Week 10*). The best method for each campaign is shown in bold. Note that $F_1 = 0$ when the similarity signal used by a classifier cannot be observed in the behavior of a particular campaign's drivers. No co-retweets were observed between any pairs of drivers in China_1

Information Op.	BLOC	Activity	CoRT	Hash	Combined
China_3	**0.995**	0.968	0.182	0.973	0.995
China_4	**0.996**	0.846	0.844	0.205	0.994
China_5	0.981	0.919	**0.991**	0.504	0.980
Iran_6	0.961	**0.986**	0.978	0.000	0.976
Venezuela_2	**0.936**	0.864	0.000	0.000	0.929
Venezuela_5	0.869	0.815	**0.903**	0.830	0.910
Spain	0.876	**0.936**	0.892	0.859	0.904
Mexico_2	**0.851**	0.569	0.312	0.071	0.874
Mexico_1	**0.833**	0.511	0.163	0.000	0.859
Uganda_1	0.850	0.807	**0.898**	0.432	0.850
Thailand	0.769	**0.815**	0.808	0.226	0.832
Uganda_2	0.800	0.574	**0.940**	0.227	0.807
Venezuela_4	**0.819**	0.603	0.754	0.391	0.779
Venezuela_1	**0.820**	0.188	0.000	0.000	0.776
Catalonia	**0.957**	0.667	0.762	0.308	0.774
China_2	**0.762**	0.000	0.000	0.000	0.762
Armenia	0.760	0.875	**0.895**	0.851	0.760
Ghana_Nigeria	0.612	**0.866**	0.741	0.528	0.700
Russia_1	**0.667**	0.000	0.000	0.000	0.667
Iran_2	**0.625**	0.200	0.000	0.000	0.667
Venezuela_3	**0.857**	0.400	0.500	0.000	0.643
Iran_5	**0.435**	0.353	0.000	0.286	0.545
Qatar	0.381	0.154	0.364	**0.533**	0.500
Russia_3	**0.500**	0.000	0.000	0.000	0.500
Ecuador	**0.533**	0.000	0.000	0.000	0.500
Iran_1	**0.286**	0.000	0.000	0.000	0.500
Russia_4	**0.483**	0.000	0.000	0.118	0.452
Egypt_UAE	**0.444**	0.000	0.000	0.000	0.435
China_1	**0.429**	0.000	0	0.000	0.429
Iran_7	**0.476**	0.000	0.000	0.200	0.421
Bangladesh	**0.600**	0.316	0.333	0.286	0.345
Iran_3	**0.316**	0.000	0.000	0.000	0.316
Iran_4	0.182	0.667	**0.857**	0.000	0.267
Russia_2	**0.200**	0.000	**0.200**	0.000	0.211
UAE	**0.200**	0.000	0.000	0.000	0.182

outperformed all baselines in the coordination detection task with a mean $F_1 = 0.659$ with the least number of features (108). The combined classifiers had a similar mean $F_1 = 0.658$, but employed the largest number of features (5869).

6 Discussion

In response to the far-reaching threats posed by influence operations on social media, researchers developed methods that target specific kinds of malicious behaviors. The effectiveness of some of these — which mostly depend on hand-crafted features — is however temporary since malicious actors evolve their tactics to evade detection. In this paper, we proposed BLOC, a language framework that represents the behavior of social media users irrespective of class (e.g., bot or human) or intent (e.g., benign or malicious). BLOC words map to features derived in an unsupervised manner. We note that the BLOC framework

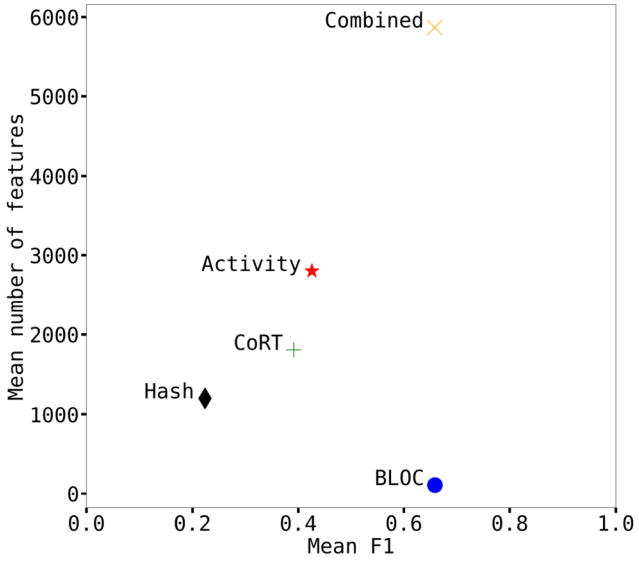

Figure 6 Mean number of features vs. mean F_1 of BLOC and four baseline classifiers for detecting drivers across information operations, estimated with data from the first 10 weeks of each information operation's lifespan

does not make feature engineering irrelevant, in fact one could engineer features using BLOC.

Although BLOC is a platform-agnostic framework, we demonstrated its flexibility through two real-world applications on Twitter. In the bot detection task, a BLOC model performed better than similar methods (Digital DNA and DNA-influenced) and comparably to a state-of-the-art method (Botometer-V4), with a much lower number of features.

It is not straightforward to attribute the BLOC improvement over DDNA because the two methods are different in terms of both language features and machine-learning algorithms — vector-based versus signature-based models. To tease apart these two contributions of the BLOC framework, we evaluated a hybrid method using the same vector representation and machine-learning model as in BLOC but in conjunction with the DDNA language. In other words, both BLOC and DDNA strings were tokenized into bigrams, and then TF-IDF vectors were classified with a random forest model. BLOC slightly outperformed the hybrid method (precision 0.879, recall 0.851, F_1 0.865). The bulk of the improvement can therefore be attributed to the higher recall provided by the vector-based classifier over the signature-based approach. The language features distinguishing BLOC and DDNA representations — pauses and repetitions — also provide useful signals in discriminating between automated and human accounts across a variety of datasets.

In the coordination detection task to identify the drivers of information operations during the early stages of their life span, a BLOC model outperformed baseline methods. The performance of all classifiers varied across information operations, which highlights the heterogeneity of the driver behaviors. This is consistent with Twitter's reports, which reveal that drivers include humans, automated accounts, coordinating accounts, and so on [57, 58]. So it comes as no surprise that the drivers of some information operations are easier than others to detect.

We also compared the performance of all classifiers on information operation datasets extracted from the *last* weeks of the life spans of the drivers — right before they were detected by Twitter. The average F_1 scores of all classifiers increased significantly (by 25–101%), suggesting that Twitter detected the drivers when their behaviors became more conspicuous. The activity method slightly outperformed the BLOC model when the evaluation was run during the last weeks of a campaign, with mean $F_1 = 0.855$ vs. 0.824, albeit using 1680 vs. 116 features. This suggests that synchronization is a strong signal for platforms to identify coordinated campaigns. For example, the activity method failed to identify the drivers of the Egypt_UAE information operation based on early data ($F_1 = 0$), but succeeded at the end ($F_1 = 1$). By contrast, the BLOC model achieved $F_1 = 0.444$ based on early data and $F_1 = 0.978$ at the end.

Collectively, these results indicate that BLOC models are versatile, effective, efficient, and applicable to multiple tasks. Also importantly, they indicate that one can achieve promising results in bot and coordination detection tasks without accounting for semantics that can be extracted from content.

We must note, however, that since BLOC representations do not capture specific content or contextual information, they may not be applicable to a variety of tasks that require a semantic understanding of the content, such as user stance detection and ideology prediction. This means that a pair of accounts on opposite ends of the political spectrum (e.g., `@HuffPost` and `@FoxNews`) could share similar BLOC words since BLOC strips away semantics such as ideology, instead focusing on behavior (e.g., original content creation). Therefore, future directions for this work include augmenting the BLOC framework to capture semantic information.

The development of new techniques such as BLOC for identifying malicious behavior could trigger changes in tactics by "puppet masters" to evade detection. We argue that even though this is possible, the introduction of BLOC could raise the bar for malicious social media accounts to appear authentic, discounting the benefits of automated tactics such as flooding and coordination.

Acknowledgements
We are grateful to Manita Pote for help with the control dataset and Kristina Lerman for suggesting the use of pause symbols in the BLOC action alphabet.

Funding
This work was supported in part by DARPA (grants W911NF-17-C-0094 and HR001121C0169), Knight Foundation, and Craig Newmark Philanthropies. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Abbreviations
API, Application Programming Interface; BLOC, Behavioral Languages for Online Characterization; COVID-19, Coronavirus Disease 2019; DDNA, Digital DNA; DNA, Deoxyribonucleic acid; KL, Kullback–Leibler; KNN, K-Nearest Neighbors; LCS, Longest Common Substring; NASA, National Aeronautics and Space Administration; PCA, Principal Component Analysis; TF-IDF, Term Frequency – Inverse Document Frequency; UAE, United Arab Emirates; URL, Uniform Resource Locator; US, United States.

Availability of data and materials
The code and datasets generated and/or analyzed during the current study are available in the Github repository, https://github.com/anwala/general-language-behavior.

Declarations

Competing interests
The authors declare that they have no competing interests.

Author contributions
ACN contributed to the project conception, data analysis, experiment design, and writing. AF and FM contributed to project conception and direction, experiment design, and writing. All authors read and approved the final manuscript.

Received: 3 November 2022 Accepted: 2 August 2023 Published online: 23 August 2023

References

1. Ratkiewicz J, Conover M, Meiss M, Gonçalves B, Flammini A, Menczer F (2011) Detecting and tracking political abuse in social media. In: Proc. Intl. AAAI conf. on weblogs and social media (ICWSM)
2. Ferrara E, Varol O, Davis C, Menczer F, Flammini A (2016) The rise of social bots. Commun ACM 59(7):96–104
3. Lazer D, Baum M, Benkler Y, Berinsky A, Greenhill K, Menczer F, Metzger M, Nyhan B, Pennycook G, Rothschild D, Schudson M, Sloman S, Sunstein C, Thorson E, Watts D, Zittrain J (2018) The science of fake news. Science 359(6380):1094–1096. https://doi.org/10.1126/science.aao2998
4. Grinberg N, Joseph K, Friedland L, Swire-Thompson B, Lazer D (2019) Fake news on Twitter during the 2016 US presidential election. Science 363(6425):374–378
5. Cresci S, Lillo F, Regoli D, Tardelli S, Tesconi M (2019) Cashtag piggybacking: uncovering spam and bot activity in stock microblogs on Twitter. ACM Transactions on the Web (TWEB) 13(2):1–27
6. Pacheco D, Hui P-M, Torres-Lugo C, Truong BT, Flammini A, Menczer F (2021) Uncovering coordinated networks on social media: methods and case studies. In: Proc. Intl. AAAI conf. on web and social media (ICWSM), vol 15, pp 455–466
7. Schiffrin A (2017) Disinformation and democracy: the Internet transformed protest but did not improve democracy. J Int Aff 71(1):117–126
8. Woolley SC, Howard PN (2018) Computational propaganda: political parties, politicians, and political manipulation on social media. Oxford University Press, London
9. Tasnim S, Hossain MM, Mazumder H (2020) Impact of rumors and misinformation on Covid-19 in social media. J Prev Med Public Health 53(3):171–174. https://doi.org/10.3961/jpmph.20.094
10. Allington D, Duffy B, Wessely S, Dhavan N, Rubin J (2021) Health-protective behaviour, social media usage and conspiracy belief during the Covid-19 public health emergency. Psychol Med 51(10):1763–1769. https://doi.org/10.1017/S003329172000224X
11. Pierri F, Perry B, DeVerna MR, Yang K-C, Flammini A, Menczer F, Bryden J (2022) Online misinformation is linked to early Covid-19 vaccination hesitancy and refusal. Sci Rep 12:5966. https://doi.org/10.1038/s41598-022-10070-w
12. Fisher M (2013) Syrian hackers claim AP hack that tipped stock market by $136 billion. Is it terrorism. https://archive.ph/VJzwk. Accessed: 2022-04-12
13. Yardi S, Romero D, Schoenebeck G et al (2010) Detecting spam in a Twitter network. First Monday 15(1)
14. Lee K, Eoff BD, Caverlee J (2011) Seven months with the devils: a long-term study of content polluters on Twitter. In: Proc. Intl. AAAI conf. on web and social, Media (ICWSM)
15. Nightingale SJ, Farid H (2022) AI-synthesized faces are indistinguishable from real faces and more trustworthy. Proc Natl Acad Sci 119(8):2120481119. https://doi.org/10.1073/pnas.2120481119
16. Cresci S (2020) A decade of social bot detection. Commun ACM 63(10):72–83
17. Mazza M, Cresci S, Avvenuti M, Quattrociocchi W, Tesconi M (2019) Rtbust: exploiting temporal patterns for botnet detection on Twitter. In: Proc. of ACM conference on web science (WebSci), pp 183–192
18. Sayyadiharikandeh M, Varol O, Yang K-C, Flammini A, Menczer F (2020) Detection of novel social bots by ensembles of specialized classifiers. In: Proc. of ACM intl. Conf. on information & knowledge management (CIKM), pp 2725–2732. https://doi.org/10.1145/3340531.3412698
19. Yang K-C, Varol O, Davis CA, Ferrara E, Flammini A, Menczer F (2019) Arming the public with artificial intelligence to counter social bots. Hum Behav Emerg Technol 1(1):48–61
20. Chu Z, Gianvecchio S, Wang H, Jajodia S (2012) Detecting automation of Twitter accounts: are you a human, bot, or cyborg? IEEE Trans Dependable Secure Comput 9(6):811–824
21. Chu Z, Gianvecchio S, Wang H, Jajodia S (2010) Who is tweeting on Twitter: human, bot, or cyborg? In: Proc. of annual computer security applications conference (ACSAC), pp 21–30
22. Wood-Doughty Z, Smith M, Broniatowski D, Dredze M (2017) How does Twitter user behavior vary across demographic groups? In: Proc. of workshop on NLP and computational social science (NLP + CSS), pp 83–89
23. He S, Wang H, Jiang ZH (2014) Identifying user behavior on Twitter based on multi-scale entropy. In: Proc. of IEEE intl. Conf. on security, pattern analysis, and cybernetics (SPAC). IEEE, pp 381–384
24. Maia M, Almeida J, Almeida V (2008) Identifying user behavior in online social networks. In: Proc. Workshop on social network systems, pp 1–6
25. Benevenuto F, Rodrigues T, Cha M, Almeida V (2009) Characterizing user behavior in online social networks. In: Proc. ACM SIGCOMM conf. on Internet measurement (IMC), pp 49–62
26. Yang K-C, Varol O, Hui P-M, Menczer F (2020) Scalable and generalizable social bot detection through data selection. In: Proc. of AAAI conf. on artificial intelligence (AAAI), pp 1096–1103
27. Cresci S, Di Pietro R, Petrocchi M, Spognardi A, Tesconi M (2017) Social fingerprinting: detection of spambot groups through dna-inspired behavioral modeling. IEEE Trans Dependable Secure Comput 15(4):561–576
28. Cresci S, Di Pietro R, Petrocchi M, Spognardi A, Tesconi M (2017) Exploiting digital DNA for the analysis of similarities in Twitter behaviours. In: IEEE intl. Conf. on data science and advanced analytics (DSAA). IEEE, pp 686–695
29. Beskow DM, Carley KM (2018) Bot conversations are different: leveraging network metrics for bot detection in Twitter. In: IEEE/ACM intl. Conf. on advances in social networks analysis and mining (ASONAM). IEEE, pp 825–832
30. Davis CA, Varol O, Ferrara E, Flammini A, Menczer F (2016) Botornot: a system to evaluate social bots. In: Proc. of intl. Conf. Companion on world wide web, pp 273–274
31. Varol O, Ferrara E, Davis CA, Menczer F, Flammini A (2017) Online human-bot interactions: detection, estimation, and characterization. In: Proc. Intl. AAAI conf. on web and social, Media (ICWSM)
32. Gilani Z, Farahbakhsh R, Tyson G, Wang L, Crowcroft J (2017) Of bots and humans (on Twitter). In: Proc. of intl. Conf. on advances in social networks analysis and mining (ASONAM). ACM, New York, pp 349–354

33. Chavoshi N, Hamooni H, Mueen A (2016) Debot: Twitter bot detection via warped correlation. In: IEEE intl. Conf. on data mining (ICDM), pp 817–822. https://doi.org/10.1109/ICDM.2016.0096
34. Keller F, Schoch D, Stier S, Yang J (2017) How to manipulate social media: analyzing political astroturfing using ground truth data from South Korea. In: Proc. of intl. AAAI conf. on web and social, Media (ICWSM)
35. Sharma K, Zhang Y, Ferrara E, Liu Y (2021) Identifying coordinated accounts on social media through hidden influence and group behaviours. In: Proc. of ACM SIGKDD conf. on knowledge discovery & data mining, pp 1441–1451
36. Assenmacher D, Clever L, Pohl JS, Trautmann H, Grimme C (2020) A two-phase framework for detecting manipulation campaigns in social media. In: Intl. Conf. on human-computer interaction (HCI). Springer, Berlin, pp 201–214
37. Vargas L, Emami P, Traynor P (2020) On the detection of disinformation campaign activity with network analysis. In: Proc. of ACM SIGSAC conf. on cloud computing security workshop, pp 133–146
38. Nizzoli L, Tardelli S, Avvenuti M, Cresci S, Tesconi M (2021) Coordinated behavior on social media in 2019 uk general election. In: Proc. Intl. AAAI conf. on web and social media (ICWSM), pp 443–454
39. Keller FB, Schoch D, Stier S, Yang J (2020) Political astroturfing on Twitter: how to coordinate a disinformation campaign. Polit Commun 37(2):256–280
40. Giglietto F, Righetti N, Rossi L, Marino G (2020) Coordinated link sharing behavior as a signal to surface sources of problematic information on Facebook. In: Intl. Conf. on social media and society, pp 85–91
41. Giglietto F, Righetti N, Rossi L, Marino G (2020) It takes a village to manipulate the media: coordinated link sharing behavior during 2018 and 2019 Italian elections. Inf Commun Soc 23(6):867–891
42. Fazil M, Abulaish M (2020) A socialbots analysis-driven graph-based approach for identifying coordinated campaigns in Twitter. J Intell Fuzzy Syst 38(3):2961–2977
43. Magelinski T, Ng LHX, Carley KM (2021) A synchronized action framework for responsible detection of coordination on social media. Preprint, arXiv:2105.07454
44. Ghosh R, Surachawala T, Lerman K (2011) Entropy-based classification of retweeting activity on Twitter. In: Proc. of KDD workshop on social network analysis (SNA-KDD)
45. Jurafsky D, Martin JH (2018) Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition, 2nd edn. Prentice Hall, New York
46. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Burges CJ, Bottou L, Welling M, Ghahramani Z, Weinberger KQ (eds) Advances in Neural Information Processing Systems, vol 26. https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf
47. Sparck Jones K (1972) A statistical interpretation of term specificity and its application in retrieval. J Doc 28(1):11–21
48. Cresci S, Di Pietro R, Petrocchi M, Spognardi A, Tesconi M (2017) The paradigm-shift of social spambots: evidence, theories, and tools for the arms race. In: Proc. of intl. Conf. Companion on world wide web, pp 963–972
49. Cresci S, Lillo F, Regoli D, Tardelli S, Tesconi M (2018) $ FAKE: evidence of spam and bot activity in stock microblogs on Twitter. In: Proc. Intl. AAAI conf. on web and social, Media (ICWSM)
50. DeVerna MR, Pierri F, Truong BT, Bollenbacher J, Axelrod D, Loynes N, Torres-Lugo C, Yang K-C, Menczer F, Bryden J (2021) In: CoVaxxy: a collection of English-language Twitter posts about Covid-19 vaccines. Proc. Intl. AAAI conf. on web and social media, ICWSM, vol 15, pp 992–999
51. Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. J Stat Mech Theory Exp 2008(10):10008
52. Nwala A, Flammini A, Menczer F (2022) A General Language for Modeling Social Media Account Behavior. https://github.com/anwala/general-language-behavior. Accessed: 2022-10-10
53. Gilmary R, Venkatesan A, Vaiyapuri G, Balamurali D (2022) DNA-influenced automated behavior detection on Twitter through relative entropy. Sci Rep 12:8022
54. Salvatore B, Stefano C, Giuseppe G, Antonio M, Angelo S, Maurizio T (2019) Digital DNA Toolbox. https://github.com/WAFI-CNR/ddna-toolbox. Accessed: 2022-08-15
55. Yu C, Deng M, Yau SS-T (2011) Dna sequence comparison by a novel probabilistic method. Inf Sci 181(8):1484–1492
56. Twitter: Information Operations. https://transparency.twitter.com/en/reports/information-operations.html. Accessed: 2022-06-15 (2022)
57. Twitter Safety: Disclosing networks to our state-linked information operations archive. https://t.co/etMqAUGwo2. Accessed: 2022-10-01 (2020)
58. Twitter Safety: Disclosing state-linked information operations we've removed. blog.twitter.com/en_us/topics/company/2021/disclosing-state-linked-information-operations-we-ve-removed. Accessed: 2022-06-15 (2021)

Publisher's Note