

A reliable and evolutive web application to detect social capitalists

Nicolas Dugué, Anthony Perez, Maximilien Danisch, Florian Bridoux, Amélie Daviau, Tennessy Kolubako, Simon Munier, Hugo Durbano

▶ **To cite this version:**

Nicolas Dugué, Anthony Perez, Maximilien Danisch, Florian Bridoux, Amélie Daviau, et al.. A reliable and evolutive web application to detect social capitalists. ASONAM 2015, Aug 2015, Paris, France. hal-01165487

HAL Id: hal-01165487
https://hal.science/hal-01165487

Submitted on 19 Jun 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A reliable and evolutive web application to detect social capitalists

Nicolas Dugué*, Anthony Perez*, Maximilien Danisch†
Florian Bridoux*, Amélie Daviau*, Tennessy Kolubako*, Simon Munier*, Hugo Durbano*
* Univ. Orléans, INSA Centre Val de Loire
LIFO EA 4022, FR-45067
† Sorbonne Univ., UPMC Univ Paris 06, UMR 7606
LIP6, F-75005 Paris, France.
Emails: nicolas.dugue@univ-orleans.fr, anthony.perez@univ-orleans.fr, maximilien.danisch@lip6.fr

***Abstract*—On Twitter, social capitalists use dedicated hashtags and mutual subscriptions to each other in order to gain followers and to be retweeted. Their methods are successful enough to make them appear as influent users. Indeed, applications dedicated to the influence measurement such as Klout and Kred give high scores to most of these users. Meanwhile, their high number of retweets and followers are not due to the relevance of the content they tweet, but to their social capitalism techniques. In order to be able to detect these users, we train a classifier using a dataset of social capitalists and regular users. We then implement this classifier in a web application that we call *DDP*. *DDP* allows users to test whether a Twitter account is a social capitalist or not and to visualize the data we use to make the prediction. *DDP* allows administrator to crawl data from a lot of users automatically. Furthermore, administrators can manually label Twitter accounts as social capitalists or regular users to add them into the dataset. Finally, administrators can train new classifiers in order to take into account the new Twitter accounts added to the dataset, and thus making evolve the classifier with these new recently collected data. The web application is thus a way to collect data, make evolve the knowledge about social capitalists and to keep detecting them efficiently.**

I. Introduction

Social capitalists were introduced by Ghosh et al. [4]. The authors observed that these users were the ones that follow the most spammers. Indeed, social capitalists are performing suscription exchanges to gain followers with the basic techniques FMIFY (Follow Me, I Follow You), IFYFM (I Follow You, Follow Me). By gaining followers, they aim to gain visibility [6], to spread more efficiently information, to appear as influent [1], or even to get some attention [2]. Dugué and Perez show that social capitalists sometimes use specific hashtags (Figure 1) to tag their tweets in order to increase the efficiency of their methods [3]. They shown that these tagged tweets allow some of these users to be highly retweeted and followed, which leads to make them influent according to measurement tools like Klout and Kred [1]. It is thus important to design tools able to detect these users in order to measure their real influence.

In this paper, we present a web application that responds to this need : *DDP*. This application discriminates efficiently in real-time social capitalists from the real users. We first introduce a method to detect accurately social capitalists presented by Danisch et al. [1]. This method uses a dataset of social capitalists and regular users to train a logistic regression classifier. We also present the new features we consider in *DDP*. Then, we describe the functionnalities available to the DDP users, including getting data about social capitalists and visualizing their behavior on Twitter. Afterwards, we present functionnalities available to the DDP administrators which are mostly used to make evolve the logistic regression model and the dataset. Subsequently, we show how to visualize and export the *DDP* dataset. Finally, we describe a few perspectives about this application.



Fig. 1. A representative tweet dedicated to social capitalism.

II. Classification

A. Dataset

Danisch et al. [1] collected a dataset of regular users and social capitalists. Regular users were obtained by getting followees of randomly sampled Twitter users. These followees are more likely to be active users than random users, and a few of them only may be social capitalists, which does not bias the study. Social capitalists of the dataset were users that posted at least three tweets tagged with hashtags dedicated to social capitalism : *#TeamFollowBack*, *#instantfollowbackdedicated* and *#teamautofollow*. Using these methods, Danisch et al. [1] collected data about $55,000$ regular users and $25,000$ social capitalists. For all these users, the features presented in Table I were collected.

To design our web application, we use the same data as Danisch et al. [1]. Nevertheless, DDP also collects the features described in Table II. We claim that these features should help learning a more robust classifier and characterizing better social capitalists behavior. Furthermore, DDP stores in a NOSQL database MongoDB all the features to constitute a long-time experiment dataset.

B. Polarity analysis

As shown in Table II, the percentage of tweets considered of each polarity (positive or negative) by *DDP* is a new feature.

CATEGORY	FEATURES
activity	number of: 1) Statuses (*i.e.* tweets) 2) lists containing the user 3) tweets being favorited
local topology	4) Size of friends set 5) Size of followers set
tweets' content	average number of: 9) characters per tweets 10) hashtags per tweets 11) URLs per tweets 12) mentions per tweets
tweets' characteristics	18) average number of retweets for a tweet 19) average number of retweets for a retweet 20) percentage of retweets among tweets
sources	using proportion of 23) Twitter official web application 24) management dashboard tool 25) automatic Follow or Unfollow tool 26) automatic tweet tool 27) other applications (Vine, Wiki, Soundcloud...) 28) phones or devices applications

TABLE I. DESCRIPTION OF THE DIFFERENT FEATURES COLLECTED THAT ARE CURRENTLY USED FOR CLASSIFICATION.

CATEGORY	FEATURES
local topology	1) Size of the intersection of the $5,000$ most recent friends and followers sets 2) Standard deviation of the $5,000$ most recent friends identifiers 3) Standard deviation of the $5,000$ most recent followers identifiers
tweets' content	average number of: 4) distinct hashtags per tweets ; 5) distinct URLs per tweets ; 6) distinct user mentionned per tweets ; 8) average and standard deviation of the similarity between tweets ; 9) percentage of positive and negative tweets.
tweets' characteristics	10) average and standard deviation of the of seconds between tweets

TABLE II. DESCRIPTION OF THE FEATURES COLLECTED THAT ARE NOT YET USED FOR CLASSIFICATION.

The polarity analysis is implemented with **NLP Tools** library. Indeed, this library allows us to remove stop words from tweets, and to learn a Naïves Bayes model to achieve this task. The *Naïve Bayes* classifier is trained offline using a dataset from Go et al. [5]. This dataset contains $2,000,000$ tweets automatically labelled using emoticons. Our classifier trained with $500,000$ of these tweets achieves an *F-Score* of 76% and an *Accuracy* of 77% on the $1.500.000$ other tweets. On a manually labelled dataset of $12,000$ tweets, it achieves an *F-Score* of 75% and an *Accuracy* of 76%, which is sligthly worse than Go et al. [5]. Still, this classifier fastly runs online and will thus allows to give an insight about the tweets polarity of social capitalists.

C. Logistic regression

The classification algorithm used in Danisch et al. [1] is logistic regression. This algorithm gives highly reliable results, runs quickly and allows interesting interpretations. Danisch et al. [1] obtained an Accuracy of 86% (correctly labelled examples) with this algorithm. So far, we do not consider the attributes described in Table II, waiting to get enough data to use them. Still, our online classifier achieves the same performance as the one described in Danisch et al. [1].

In the next section, we focus on the functionnalities implemented in *DDP*. We show that DDP allows to keep collecting data about social capitalists, to store these data that are more complete (Table II), and to make evolve the classifier in order to achieve great accuracy even if social capitalists behavior change.

III. USER FUNCTIONNALITIES

The *DDP* homepage (http://45.55.196.48/WebApp/) allows the user to learn about our project by clicking on *How does it work ?* or *About* at the top right. Furthermore, it is especially possible to log in to *DDP* using a Twitter account by clicking on *Sign in with Twitter* (Figure 2). This *logging in* step makes Twitter to create a *token pair* that associates the Twitter account and *DDP*. Both these tokens are used by DDP to send request to the Twitter REST API in the name of the user logged in. This step is thus important because it guarantees the user to be able to have a certain amount of requests with *DDP*.



Fig. 2. DDP homepage, sign in with Twitter to try DDP.

Once *logged in*, one can fill a form with any Twitter *identifier* (without @) to test whether or not the user with this identifier is a social capitalist. *DDP* collects data about the corresponding user u to the screen name filled. Features listed in Tables I and II are fastly obtained using the *tokens*. *DDP* then uses the *logistic regression* coefficients stored in the MongoDB database to predict the probability of u to be a social capitalist.

All the features obtained and the probability of u to be a social capitalist are then displayed on a dedicated page. At the top of the webpage, we present general information. One can observe the profile image of u and the Klout and Kred scores of u (Figure 3). The DDP score (Equation 1) filled in white inside a red circle is displayed on the profile image. *DDP* also displays the probability (P_{Ksoc}) returned by the logistic regression (Figure 4).

$$S_{DDP} = \begin{cases} S_{Klout} & \text{if } P_{Ksoc} \leq 0.5 \\ 2(1 - P_{Ksoc})S_{Klout} & \text{if } P_{Ksoc} > 0.5 \end{cases} \quad (1)$$



Fig. 3. Klout, Kred and DDP scores of *z_o_m_b_ii_e*.



Fig. 4. Probability of *z_o_m_b_ii_e* to be a social capitalist.

Then, *DDP* shows information about user *Activity* (see Table I) : number of followers, friends, tweets, favourites tweets and lists the user belongs to (Figure 5).

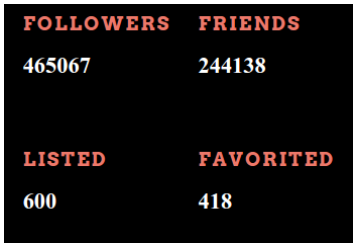


Fig. 5. *z_o_m_b_ii_e* activity features.

Afterwards, *DDP* describes the tweets characteristics :

- The percentage of tweets and retweets among what is tweeted by u ;
- The sources used by u to tweet (Table I);
- The hours when the u tweets were posted;
- The three most popular tweets of u.

As one can observe, hourly numbers of tweets are useful to detect robots. Figure 6 shows at right hourly tweets of an automated account, and at left a manual social capitalist.

On Figure 7, from left to right, DDP displays interactive word clouds containing :

- User mentionned by u;
- The distinct hashtags collected in u's tweets.

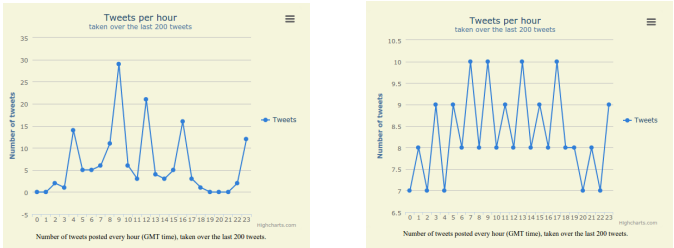


Fig. 6. Daily tweets of *z_o_m_b_ii_e* at left and *_followers_team_time*.



Fig. 7. Cloud words about hashtags used and users mentionned by *z_o_m_b_ii_e*.

We also display the distinct URLs collected in u's tweets using cloud words. In a cloud, the larger a user, a hashtags or an URL is, the more it is observed in the tweets posted by u.

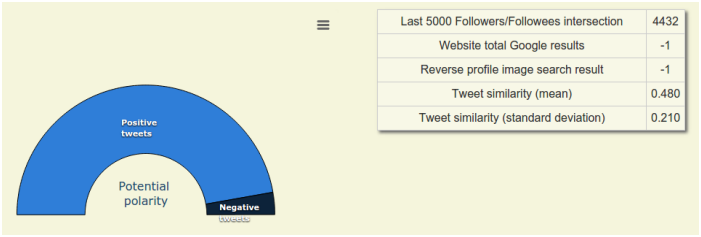


Fig. 8. Tweets polarity, similarity of *z_o_m_b_ii_e* and a few other features.

Finally, the last result screen (Figure 8) presents at left the percentage of tweets detected with a positive and negative *polarities*. At right, *DDP* displays :

- The overlap index between the $5,000$ friend and follower sets;
- The number of results returned by the request submitted to *Google* search engine with the website filled by u;
- The mean similarity and standard deviation between the tweets posted by u.

If u has not already filled a website, then -1 is displayed.

IV. Administrator Functionalities

Refresh Scores

username	Screen Name	friends	Followers	Score	klout	kred	Capitalist
SondyNews	#TeamFollowback	265	70744	0.4669	50	79	CAP
z_o_m_b_ii_e	#TeamFollowBack	244090	465098	0.9996	null	null	CAP
Leboss107	#TeamFollowBack	41461	134402	0.8062	44	79	CAP
TeamFollowWacky	#TeamFollowBack	6218	478658	0.2142	64.684994457047	100	CAP
BEZESIRO	#TeamfollowBack #FF	9206	83360	0.0556	46.614730862678	79	CAP
ManmiPaLa	#Teamfollowback	35	17378	0.5542	null	null	CAP
AliceJard	Alice Jard	1825	1244	0.3523	null	null	unknown
CookieGamerMel	Amélie Daviau	51	17	0.0429	null	null	NOT-CAP

Fig. 9. Labelling users from the administrator side.

Figure 9 shows a list of the Twitter accounts that were filled by the *DDP* users. All these accounts and their features that were collected using the Twitter API are stored in a MongoDB datase. The administration interface allows DDP administrators to confirm or disconfirm results predicted by the classifier. Indeed, administrators can manually label each of the users tested. Three labels are available: *cap*, *not-cap* and *unknown*

which is the default one. Once labelled manually, a user is added to the dataset. Thus, this user can be used to improve classification as a new labeled observation. Furthermore, this user expands information about how social capitalists behave.

accuracy (train)	sensitivity (train)	specificity (train)	f-score (train)
85.78% (85.63%)	88.93% (88.45%)	84.40% (84.44%)	86.60% (86.40%)
85.76% (85.64%)	88.86% (88.48%)	84.40% (84.44%)	86.57% (86.41%)

Fig. 10. Logistic regression viewed from the administrator side.

Logistic regression may be run by administrators by clicking on *Start Classifier*. Logistic regression coefficients and results are stored in MongoDB and displayed as seen in Figure 10. The green line is the one chosen by the administrator: the corresponding coefficients to the results are the ones used in *DDP* to make predictions. So far, the attributes described in Table I are used to learn a model. Once we will have enough users with all the features, especially those in Table II, these features will be used. In order to fastly collect data, a batch tool was designed (Figure 11).

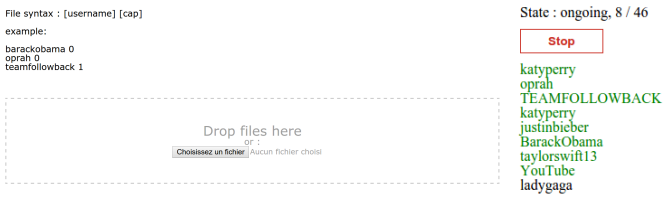


Fig. 11. Collecting data in *batch* mode.

At left, it is possible to upload a file with users labeled in order to obtain their features. Once data about each of these users are collected, these users are added to the dataset and all their features stored in MongoDB. At right, we can see how is going the data collection (Figure 11).

V. Export Dataset

All users can access functionnalities to export and visualize data stored in MongoDB (http://45.55.196.48/WebApp/db/users.php). Figure 12 shows how users tested in DDP are displayed. Each user features can be exported in JSON (Figure 13), and the full database is also available in JSON.

Get all data in JSON Format

Twitter	JSON	Followers	Friends	Statuses	Score	klout	Capitalist
adrienguille	adrienguille	28	23	1	0.00020475511357493	0	NOT-CAP
barackobama	barackobama	59470056	642818	13517	0.00040985546106143	99	unknown
nicolasdugue	nicolasdugue	130	131	723	0.017257044866915		NOT-CAP
alicejard	alicejard	1258	1851	4383	0.56788134671183		CAP

Fig. 12. Visualizing database and exporting it in JSON.

DDP also provides visualization related to the interactions between social capitalists. Users can export the mention graph induced by social capitalists and visualize it online.

VI. Conclusion and Perspectives

The *DDP* web application implements an evolving and reliable method to detect social capitalists based on the Danisch et al. [1] work. First of all, *DDP* implements useful visualization features. Furthermore, new attributes are collected to expand knowledges about social capitalists. Indeed, all collected data are stored in a MongoDB database. This also allows to make evolve the classifier by running logistic regression when new examples are labelled by *DDP* administrators and thus added to the dataset. As a result, *DDP* is a robust tool that allows a long-time accurate detection of social capitalists while collecting new data to better study them.

In a future work, we shall study the relevance of the new features describes in Table II to detect social capitalists. We would do this using the odd ratio of the logistic regression that we use to classify each user. Furthermore, it would be interesting to experiment if spammers and robots are also detected by our classifier. Finally, we saw the same hashtags as the ones used by Danisch et al. [1] appear on Facebook. Considering those Facebook status may be a great improvement.


```
{
    "_id": {
        "$id": "555cbd2463e94eb3991e2d43"
    },
    "username": "dn_junior",
    "vals": {
        "error": "no",
        "screen_name": "Junior #MGWV",
        "screen_name_l": "junior #mgwv",
        "profile_pic": "http:\/\/pbs.twimg.com\/profile_images\/593466792716173313\/7bpaGLTd.jpg",
        "score": 0.9753878385226,
```


Fig. 13. Visualizing user features in JSON.


```
{"user ": "alicejard", "mentions ": "Film_Pics"}
{"user ": "alicejard", "mentions ": "iamfonda"}
{"user ": "alicejard", "mentions ": "droledimage"}
{"user ": "alicejard", "mentions ": "SalomFouve"}
{"user ": "alicejard", "mentions ": "eliroth"}
{"user ": "alicejard", "mentions ": "SW_Stickers4"}
{"user ": "alicejard", "mentions ": "LvObservateur"}
{"user ": "alicejard", "mentions ": "NicolasCeurs"}
```

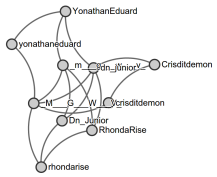



Fig. 14. Mention graph induced by social capitalists.

References

[1] Maximilien Danisch, Nicolas Dugué, and Anthony Perez. On the importance of considering social capitalism when measuring influence on twitter. In *Behavioral, Economic, and Socio-Cultural Computing*, 2014.

[2] Shaun W. Davenport, Shawn M. Bergman, Jacqueline Z. Bergman, and Matthew E. Fearrington. Twitter versus facebook: Exploring the role of narcissism in the motives and usage of different social media platforms. *Computers in Human Behavior*, 32(0):212 – 220, 2014.

[3] Nicolas Dugué and Anthony Perez. Social capitalists on Twitter: detection, evolution and behavioral analysis. *Social Network Analysis and Mining*, 4(1):1–15, 2014. Springer.

[4] S. Ghosh, B. Viswanath, F. Kooti, N. Sharma, G. Korlam, F. Benevenuto, N. Ganguly, and K. Gummadi. Understanding and combating link farming in the Twitter social network. In *WWW*, pages 61–70, 2012.

[5] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, pages 1–12, 2009.

[6] Vincent Labatut, Nicolas Dugué, and Anthony Perez. Identifying the community roles of social capitalists in the twitter network. In *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*, pages 371–374. IEEE, 2014.