A reliable and evolutive web application to detect social capitalists

Nicolas Dugué*, Anthony Perez*, Maximilien Danisch†
Florian Bridoux*, Amélie Daviau*, Tenessy 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 subscription 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 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 functionalities available to the DDP users, including getting data about social capitalists and visualizing their behavior on Twitter. Afterwards, we present functionalities 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.

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.
### Table I. Description of the Different Features Collected That Are Currently Used for Classification.

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

### Table II. Description of the Features Collected That Are Not Yet Used for Classification.

<table>
<thead>
<tr>
<th>CATEGORY</th>
<th>FEATURES</th>
</tr>
</thead>
<tbody>
<tr>
<td>local topology</td>
<td>1) Size of the intersection of the 5,000 most recent friends and followers sets</td>
</tr>
<tr>
<td></td>
<td>2) Standard deviation of the 5,000 most recent friends identifiers</td>
</tr>
<tr>
<td></td>
<td>3) Standard deviation of the 5,000 most recent followers identifiers</td>
</tr>
<tr>
<td>tweets’ content</td>
<td>average number of:</td>
</tr>
<tr>
<td></td>
<td>4) distinct hashtags per tweets ;</td>
</tr>
<tr>
<td></td>
<td>5) distinct URLs per tweets ;</td>
</tr>
<tr>
<td></td>
<td>6) distinct user mentioned per tweets ;</td>
</tr>
<tr>
<td></td>
<td>8) average and standard deviation of the similarity between tweets ;</td>
</tr>
<tr>
<td></td>
<td>9) percentage of positive and negative tweets.</td>
</tr>
<tr>
<td></td>
<td>10) average and standard deviation of the of seconds between tweets</td>
</tr>
</tbody>
</table>

The polarity analysis is implemented with **NLP Tools** library. Indeed, this library allows us to remove stop words from tweets, and to learn a Naïve 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 slightly 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 functionalities 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 Functionalities

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**.

![DDP homepage, sign in with Twitter to try DDP.](Image)

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 (**PKsoc**) returned by the logistic regression (Figure 4).

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

(1)
Fig. 3. Klout, Kred and DDP scores of \textit{z_o_m_b_i_e}.

**#TEAMFOLLOWBACK**

According to our algorithm, \#TeamFollowBack has 100\% of chance of being a social capitalist.

Fig. 4. Probability of \textit{z_o_m_b_i_e} to be a social capitalist.

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

![Activity Features](image)

Fig. 5. \textit{z_o_m_b_i_e} activity features.

Afterwards, \textit{DDP} describes the tweets characteristics:

- The percentage of tweets and retweets among what is tweeted by \textit{u};
- The sources used by \textit{u} to tweet (Table I);
- The hours when the \textit{u} tweets were posted;
- The three most popular tweets of \textit{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 mentioned by \textit{u};
- The distinct hashtags collected in \textit{u}’s tweets.

![Word Clouds](image)

Fig. 6. Daily tweets of \textit{z_o_m_b_i_e} at left and _followers_team_time.

![Word Clouds](image)

Fig. 7. Cloud words about hashtags used and users mentioned by \textit{z_o_m_b_i_e}.

We also display the distinct URLs collected in \textit{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 \textit{u}.

![Tweets Polarity](image)

Fig. 8. Tweets polarity, similarity of \textit{z_o_m_b_i_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, \textit{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 \textit{u};
- The mean similarity and standard deviation between the tweets posted by \textit{u}.

If \textit{u} has not already filled a website, then $-1$ is displayed.

**IV. ADMINISTRATOR FUNCTIONALITIES**

Figure 9 shows a list of the Twitter accounts that were filled by the \textit{DDP} users. All these accounts and their features that were collected using the Twitter API are stored in a MongoDB database. 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: \textit{cap}, \textit{not-cap} and \textit{unknown}.

![Labeling Users](image)

Fig. 9. Labelling users from the administrator side.
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.

![Fig. 10. Logistic regression viewed from the administrator side.](image)

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.](image)

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 functionalities 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.

![Fig. 12. Visualizing database and exporting it in JSON.](image)

**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 knowledge 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 described 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.

**REFERENCES**


