Facebook Users Relationships Analysis based on Sentiment Classification

Diego Terrana, Agnese Augello, Giovanni Pilato
ICAR - Istituto di Calcolo e Reti ad Alte Prestazioni
CNR - Consiglio Nazionale delle Ricerche
Viale delle Scienze - Edicio 11 - 90128 Palermo, Italy
Email: {terrana, augello, pilato}@pa.icar.cnr.it

***Abstract*—It is presented an approach aimed at analyzing the homepage of a Facebook user or group in order to automatically detect who has discussed what and how it has been discussed. All public posts shared by an user are retrieved by an ad hoc built crawler. Information such as a text messages, comments, likes, is extracted for each post. Each post is classified as belonging to a set of predefined categories and its sentiment is also detected as being positive, negative or neutral. All the comments to that post are therefore analyzed and categorized together with its sentiment polarity. For each category it is created a graph where it is highlighted the concordance of sentiment between the posts and the related comments. The graph can be therefore used to profile the user relationships according to sentiment classification.**

***Index Terms*—Sentiment Analysis, Users Profiling, Facebook**

I. Introduction

In recent years we have seen an evolution in the world of the World Wide Web due to the birth and the exponential growth of social networks. A social network is a social structure made up of actors (individuals or organizations) connected together by different social relationships (friendships, work relationships, family relationships, etc ...), organized around one or more thematic areas. Social networks such as $Facebook$ [1] and $Twitter$ [2] have started a $great\ revolution\ of\ social\ relations$. Discussions among users of social networks are potentially of great interest. Millions of users, in fact, share through social networks information, news, general events.

Finding similar users and modeling their profile and social relationships is a key issue in several research fields. In the context of social networks, the simplest way to compute user similarities is by means of accepted similarity metrics. In literature, Adamic and Adar [3] have proposed that the similarity of two users increases if they share $friends$ who have a low number of friends themselves. Catanese et al. [4] have developed a tool for analyzing quantitative and qualitative properties of social networks to understand the organization of Facebook adopting techniques and algorithms such as breadth-first-search (BFS) algorithm. Ereteo et al. [5] have developed a framework for exploiting the graph models representations of social networks. They have provided an ontology of characteristics used to annotate social networks. Akram Al-Kouz et al. [6] have analyzed the explicit and implicit social graph to model user interests and fields of expertise. They classified users as groups and posts by using a general ontology derived to infer a valid model to represent the user expertise. Bernhard Rieder [7] described an extraction application ($NETVIZZ$) to analyze quantitatively and qualitatively groups and pages of Facebook. Amparo E. Cano et al [8] propose a variation of the PageRank algorithm for analysing users topical and entity influence to derive influential users based on the retweet subgraph of the Twitter graph.

The proposed system analyzes the textual data generated or shared by a user or a group of facebook in order to identify automatically who has discussed $what$ and how has discussed.

Our system analyzes the $homepage$ of a user or a group on Facebook. We define the analyzed account as the $User\ Root$. This user discusses and shares posts with other Facebook users (who) related to various topics ($what$). They may express positive, negative or neutral opinions (how). All public posts shared by the $User\ Root$ are retrieved by a crawler built using $Facebook\ Graph\ APIs$ [9] and $Facebook\ Query\ Language\ (FQL)\ Table\ Reference$ [10] ($Facebook\ Crawler$). Information such as $text\ messages$, $comments$, $likes$ is extracted for each post ($Information\ Detection$). The textual content is analyzed by using Linguistic Inquiry and Word Count (LIWC) [11] software to calculate the degree to which the $User\ Root$ uses different categories of words ($Category\ Identification$). A simple sentiment classifier is built to classify facebook messages based on their emotional content as positive, negative and neutral ($Polarity\ Calculation$). Finally, the proposed system creates graphs of the topics grouped by type and sentiment where nodes represent the users who interact with the $User\ Root$ and edges that connect them are shared posts ($GrapphCreation$).

II. Facebook User Profile Analyzer

The proposed procedure to perform the analysis of the $User\ Root$ profile is illustrated in Figure 1.

The following sections describe each one of the steps in detail.

A. Facebook Crawler

We have used $Facebook\ Graph\ APIs$ and $Facebook\ Query\ Language\ (FQL)\ Table\ Reference$

978-1-4799-4003-5/14 $31.00 © 2014 IEEE
DOI 10.1109/ICSC.2014.59

Fig. 1: Facebook User Profile Analyzer

to obtain data of the *User Root* from the Facebook social graph.
A simple and flexible Facebook Graph API (RestFB [12]) client written in Java has been used to capture Facebook messages and making queries in real time. A valid access token is necessary to invoke Facebook services and acquire the data of an user or a publicly available page. The access token is a string generated by the Facebook system at the end of the authorization process. It represents a set of permissions that have been granted and it can be used in the context of a specific application or a specific person.
First of all, we have created a client for Facebook information accessing. Subsequently, we have invoked the *fetchObject* method to obtain the *FacebookPage* about the *User Root*:

B. Information Extraction

The client defined in the previous step allows us to catch data such as name, location, history, favorites, education, friends, etc.. Many of the objects in the returned data are links that we can explore. The information, displayed according to the Facebook's privacy policy and accepted by the user, regards : *Home*, *Friends*, *Family*, *Activities*, *Interests*, *Music*, *Books*, *Movies*, *Posts*, *Links*, *Notes*, *Groups*, *Posts*, *Photos*, *Likes* and *Statuses*.
Facebook imposes some limitations on the number of posts, comments and likes retrievable through its APIs. It is not possible to retrieve the information of more than 25 *like* or *comment*. We have used *Facebook Query Language (FQL) Table Reference* to partially overcome these limitations and to directly access to the tables *LIKE*, *USER* and *COMMENT* Facebook and retrieve the missing information. The extracted information is in Java Script Object Notation (JSON) format as reported below:


```
{{"Posts":[{
    "id":
       "100000503162381_741986625828103",
    "Caption": "","Description":
       "","Name": "",
    "Place": {},
    "CreatedTime": "Fri Oct 1802:47:19
       CEST 2013",
    "UpdatedTime": "Fri Oct 1802:49:20
       CEST 2013",
    "Message": "",
    "Type": "link", "Source": "",
    "messageTags": [],"likesCount": "",
    "To": {},"usersLike": [],
    "comments":[{"id":
       "741986625828103_32094960",
          "From": {"Id":
             "100000503162381",
                "Name": "zzzzz",
                "Category":
                   "","Type": ""}
          "LikeCount": 1,"Type": "",
          "CreatedTime": "Fri Oct 1802
             :49:20CEST 2013",
          "Message": "....",
          "UserLikes": false,
          "likes":[{"sex: ": "male",
          "first_name: ":
             "xxx","last_name: ":
             "yyyy",
          "name":
             "xxxyyyy","middle_name:
             ": "xxx.yyyy",
          "username: ": "xxx.yyyy",
          "birthday_date: ": null,
          "from": {"Id":
             "100001321111710",
          "Name": "xxxyyyy",
          "Category": "","Type":
             ""}}]}]}}
```


C. Category Identification

We have calculated the degree to which the *User Root* uses different categories of words across a wide array of Facebook messages. At this step, we have used a simplified version of the dictionary provided by the Linguistic Inquiry and Word Count (LIWC). It is a text analysis software program designed to process written text by looking for a dictionary match for each word composing the text. If the target word matches the dictionary word, the appropriate word category scale for that word is increased. In our study, we have analyzed only 46 (*Table I*) by the standard 80 categories provided by LIWC

D. Polarity Calculation

The proposed system identifies the main topics discussed by the *User Root* and classifies several posts identifying the sentiment polarity (*positive*, *negative*, *neutral*). We have built a simple and totally unsupervised sentiment classifier using the system proposed in [13] where short and informal texts such as those contained in social networks are preprocessed and classified according to on their emotional content as being *positive*, *negative* or *neutral*.

Family	Friends	Humans	Positive emotion
Negative emotion	Anxiety	Anger	Sadness
Inhibition	Home	Money	Religion
Death	Work	Time	Body
Sport	Sexual	Certainty	Affective
TV	Body	Motion	Music
School	Discrepancy	Causation	Leisure
Consensus	Positive Feeling	Optimism	Cognitive mechanism
Introspective	Possibility	Sensitive Process	Feeling
Social	Comment	Generic Reference	Movement
Employment	Metaphysical	Physical	Food
Sleep	Body Care	Dirty Word	Formal

TABLE I: List of categories used in the proposed procedure.

We do not use any external resource such as lexicons, dictionaries or other manually tagged datasets; we use only the sentiment expressed by emoticons in the training dataset as a source of information. We assume that the presence of an emoticon is a clue about the sentiment that the microblogger wants to express in the post. The training dataset has been built using the stream of tweets due to technical difficulties in obtaining a complete and consistent stream of public posts on Facebook.

Tweets messages containing emoticons have been retrieved by using the Twitter APIs and then they have been collected into two sets: positive set and negative set. We exploit a set of emoticons classified as being positive or negative. As a consequence we split the tweets messages into two class. We assume that a word appearing frequently in positive class and rarely in negative class should have a high polarity positive score and, analogously, that a word appearing frequently in negative rather than in positive class, should have a low polarity score.

For each word w_i, we count the number of occurrences in positive and negative emoticons sets and we use the following formula to estimate the polarity score of it:

$$polarity(w) = \frac{occ_+(w_i) - occ_-(w_i)}{occ_+(w_i) + occ_-(w_i)} \quad (1)$$

where:
$occ_+(w_i)$ = word occurrences in the positive class.
$occ_-(w_i)$ = word occurrences in the negative class.

The polarity value of a word is a value between -1 and 1 (-1 means strongly negative, 0 means neutral, 1 means strongly positive,).

The list of all the polarized words automatically extracted from the training corpus constitutes the $opinion\ words$ of the dictionary to be used in the classification step of a new post. Since posts from social networks contain usually $noisy\ text$, i.e. text that does not comply with the standard rules of orthography, syntax and semantics, we filter out not frequent words with less than a number of k characters.

If the polarity of a text exceeds a given positive threshold value we classify it as positive; if its score is below a negative threshold value we classify it as negative, otherwise it is considered being neutral.

The polarity score of a post is given by the average of the sum of the polarity scores of its words. We consider all the words as sentiment indicators also verbs and nouns in addition to adjectives. Given a message m which can be regarded as a collection of n words $w_1, w_2, ..., w_n$, we define its polarity score as the mean of polarity scores of all the terms having more than k characters:

$$polarity(m) = \frac{\sum_{i=1}^{N} polarity(w_i)}{N} \quad (2)$$

where
w_i = is an opinion word.
$polarity(w_i)$ = is the polarity score of a term w_i calculated by using the constructed affective lexicon in the previous step according to the positive and negative tweets in the training corpora.
N = is the number of words in the tweet s.
Sentiment classification of message m is obtained exploiting the $polarity(m)$ as value:

- If $polarity(m) \geq polarityAvg + \epsilon$ then the text is considered to have a positive polarity;
- If $polarity(m) < polarityAvg + \epsilon$ and $polarity(m) > polarityAvg - \epsilon$ then the text is considered neutral;
- If $polarity(m) \leq polarityAvg - \epsilon$ then the text is considered to have a negative polarity.

where
$polarityAvg$ is the mean polarity of all the words in the lexicon.
ϵ is the threshold value experimentally calculated. It defines the polarity of the neutral tweets.

E. Graph Creation

The system creates a graph G = (V, E) for each topic identified, where:
V = set of user nodes. They are users connected by a few post or like with the $User\ Root$
E = set of edges. An edge is always related with the $User\ Root$ and a node of V and it represents an exchanged or shared post among themselves.
The $User\ Root$ posts are caught using RestFB [12]. It is a simple and flexible Facebook Graph API client written in Java able to catch facebook messages and make queries in real

time. The training dataset for polarity classification has been retrieved by using the Twitter APIs (statuses/samplemethod). The twitter4j [14] JAVA library has been used to caught tweets and make queries in real time. The returned data is a set of documents, one for each tweet. We stored tweets into text files according to the Java Script Object Notation (JSON) format. Each retrieved tweet, in addition to the plain text of the tweet contains many other fields, including $userid$, $date$, $source$, $type$, $profile$, $location$, $number\ of\ favorites$, $friends$, $followers$, URL, $hashtag$, etc.. The JSON tweet document contains attributes describing the tweet, user information, tweet relations with other tweets, a lists of urls, hashtags and user mentions contained in the tweet. In some cases information related with the location of the user is also provided in the document. Given the average number of 170 million tweets sent per day [15] it is expected that the size of the data retrieved by the Streaming APIs (1%), in 24 hour will be around 1.700.000 tweets.

III. Experimental Results

In our experiments, we have analyzed the profile of several users of Facebook and some public pages including those related to political Italian events such as $Berlusconi - Dimettiti$[1] page born to demand the resignation of Prime Minister Berlusconi. The system has caught and analyzed all the posts on the page in order to study the profile of the page user's manager compared to all the others who have commented or followed the same page.
We have created a large dataset[2] of tweets using Twitter search API from December 2012 to February 2014, to be used in training of polarity classifier. We have collected more than 600 million tweets partitioned into documents of ten thousand lines within folders, one for each day of scanning.
Each file contains one sigle row for each tweet in JSON (JavaScript Object Notation) format which stores in addition to text other additional information such as date, type, source, user profile, location, followers number, friends number, favorites number. All the information is stored by using a key / value encoding for each data. Some data such as location and profile information have not been considered in this approach and they will be used in future work. The tests were carried out by analyzing Italian Facebook account. We have split these tweets in a positive and a negative class with respectively 2132116 and 37323 tweets. The positive and negative retrieved corpora are the real proportions of positive and negative tweets found on Twitter and they have been used to set up the classification parameters in order to recognize positive and negative sentiments. The created dictionary of words contains 103011 polarity terms with more than $K = 3$ characters. The mean polarity calculated to validate our approachis $polarityAvg = 0.670392$.
The system has identified, for the analyzed pages, these topics: $work$, $death$, sex, $religion$, $sport$, $family$, $friendship$, TV, $money$, $time$, $social$, $school$, $food$. It has highlighted: $inhibition$, $positive\ feelings$, $negative\ feelings$, $certainty$, $sadness$, $anger$, $consent$, $swearwords$, $positive\ emotions$, $negative\ emotions$, $discrepancy$, $affection$, $optimism$, $sensory\ process$, $cognitivemechanism$, $introspective$. Figure 2 shows the graph connecting all the posts between the $User\ Root$ ($Berlusconi - Dimettiti$ page) and other users of the page. The darker areas represent users with the highest number of interactions with the page.
Figure 3 shows the graphs of posts with positive and negative emotions respectively. The graph of negative posts has a higher density, it shows a prevalence of negative content.

One of the most discussed topics is $Social$(Figure 4). Our system has analyzed the polarity of posts identifying a graph with negative polarity, one with neutral polarity and the one with positive polarity between the $User\ Root$ posts and the comments of other users. Again, the graphs density shows a prevalence of posts with negative polarity compared with neutral and positive posts. We have also analyzed the accounts of different users. For privacy, we report the results obtained by analyzing the facebook account of one of the authors of this article. Figure 5 is the graph of his posts. The darker area shows the users that interact more closely with him. The system has shown that this user interacts more closely with 156 users, with 68 for a number of times greater than 10.
$Table\ II$ are the main LIWC categories identified by the system for the analyzed user. $Table\ III$ are the main sentiments identified by the system for the analyzed user.

IV. Conclusions and Future Work

In this paper, we have proposed a system to automatically analyze the $homepages$ of Facebook public profiles by identifying main topics, emotions and polarity class. The system is able to generate graphs of user interactions with other users. It highlights the users with whom the $User\ Root$ mainly discusses, what they discuss and people with whom he agrees and with whom he is in disagreement.
The sentiment analysis phase does not require any external dictionary of polarized words to catch the sentiment polarity in everyday colloquial texts. The approach makes it possible to map colloquial expressions with new words, slangs and errors. Experimental tests showed the different density of graphs according to the topics covered. As a future work, we plan to trace the complete psychological traits of social network users by analyzing the textual content of posts published and to analyze how these have changed over time.

V. Acknowledgements

This work has been partially supported by the PON01_01687 - SINTESYS (Security and INTElligence SYSstem) Research Project.

[1] https://www.facebook.com/pages/Berlusconi-Dimettiti/173470385996413?fref=ts

[2] The dataset will be available at http://lithium.pa.icar.cnr.it/twitter_dataset/.

References

[1] "Facebook - resources available at:https://www.facebook.com/."
[2] "Twitter for business - resources available at:https://business.twitter.com/."

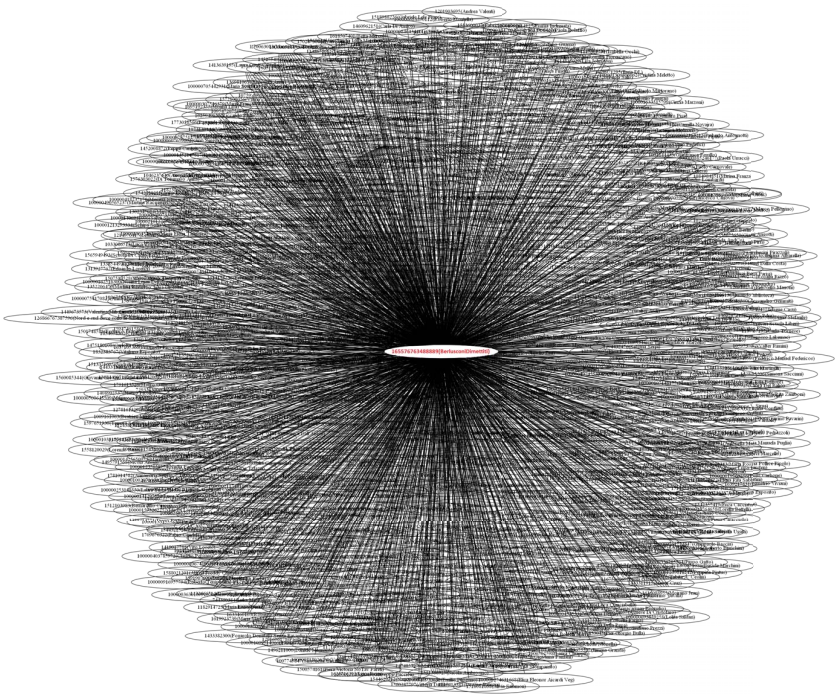

Fig. 2: Graph of interactions with $Berlusconi - Dimettiti$ page. The darker areas represent users with the highest number of interactions with the page.

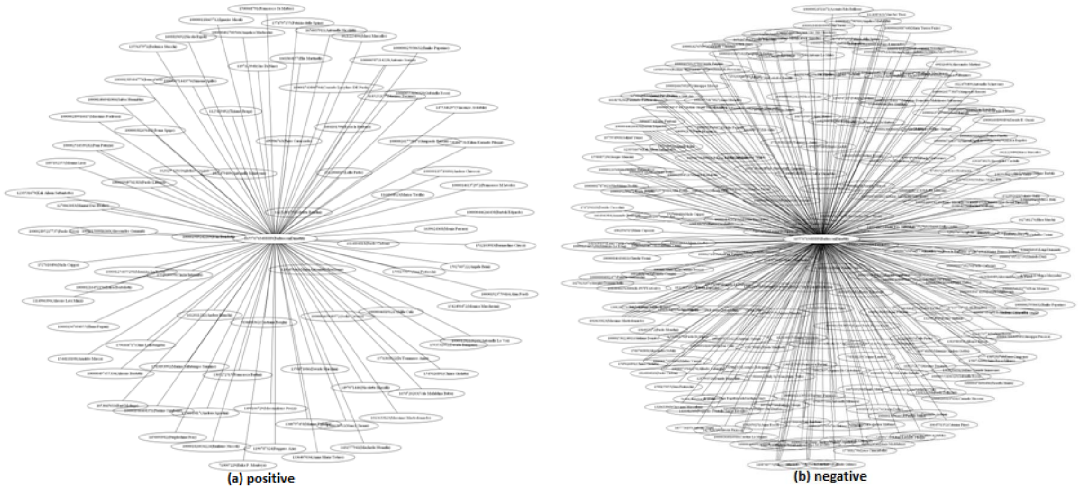

Fig. 3: Graph of posts with positive and negative emotions for $Berlusconi - Dimettiti$ page. The graph of negative (b) posts has a higher density, it shows a prevalence of negative content.

Friends	Home	Comment
Generic_Reference	Body_Care	Sport
Sleep	Family	Work
Food	Death	Music
Religion	School	Sentiment
Sex	Social	Money
hobby	Time	TV
Possibility		

TABLE II: Main LIWC categories for $diego.terrana$ account.

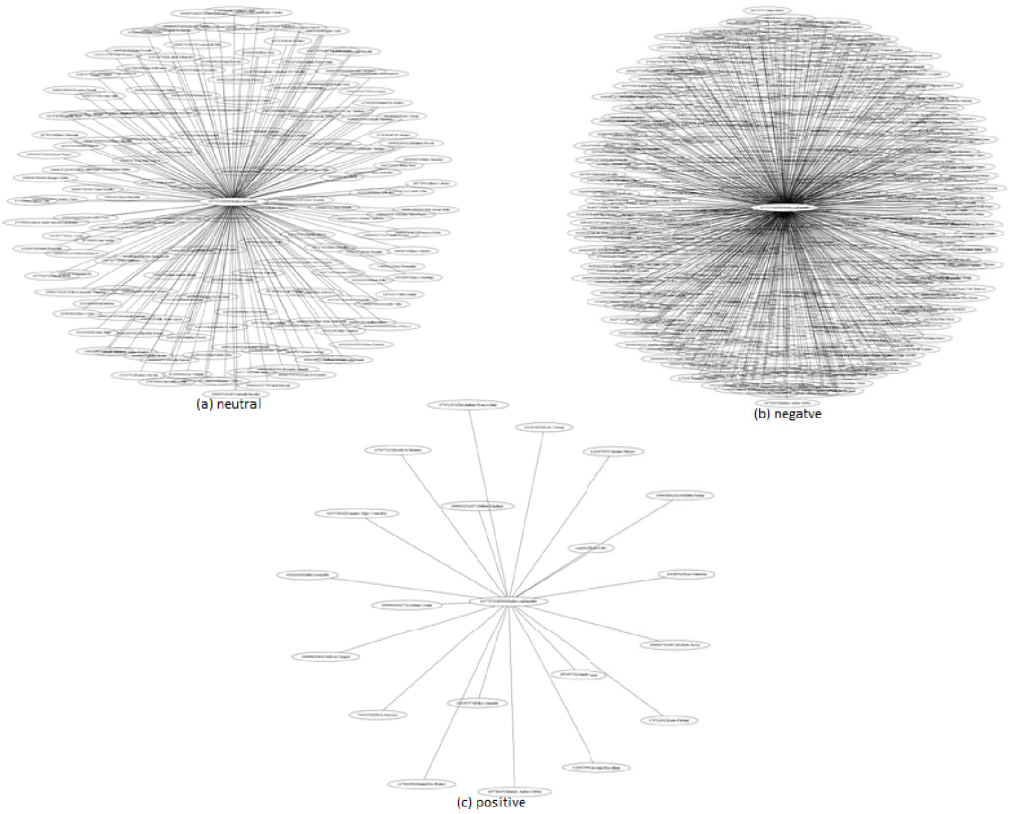

Fig. 4: Graph of posts for the Social topic in $Berlusconi - Dimettiti$ page. The graphs density shows a prevalence of posts with negative polarity (b) compared with neutral (a) and positive (c) posts.

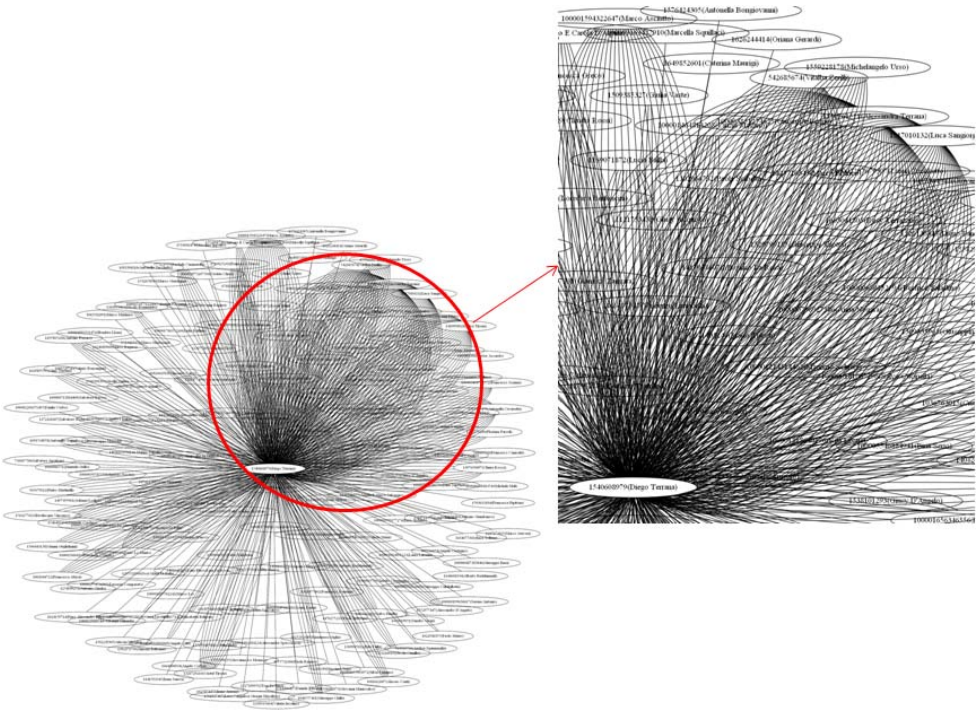

Fig. 5: Graph of posts for $diego.terrana$ page. The darker area shows the users that interact more closely with him.

Affective	Anxiety	Consensus
Certainty	Discrepancy	Negative_Emotion
Positive_Emotion	Inhibition	Introspective
Cognitive_Mechanism	Metaphysical	Optimism
Sensitive_Process	Anger	Sadness

TABLE III: Main sentiments for $diego.terrana$ account.

[3] L. Adamic and E. Adar, "Friends and neighbors on the web," *Social Networks*, vol. 25, pp. 211–230, 2001.

[4] S. Catanese, P. D. Meo, E. Ferrara, G. Fiumara, and A. Provetti, "Extraction and analysis of facebook friendship relations," 2011. [Online]. Available: http://cogprints.org/7668/

[5] G. Erétéo, M. Buffa, F. Gandon, and O. Corby, "Analysis of a real online social network using semantic web frameworks," in *International Semantic Web Conference*, 2009, pp. 180–195.

[6] A. Alkouz, E. W. D. Luca, and S. Albayrak, "Latent semantic social graph model for expert discovery in facebook," in *IICS*, 2011, pp. 128–138.

[7] B. Rieder, "Studying facebook via data extraction: the netvizz application." in *WebSci*, H. C. Davis, H. Halpin, A. Pentland, M. Bernstein, and L. A. Adamic, Eds. ACM, 2013, pp. 346–355. [Online]. Available: http://dblp.uni-trier.de/db/conf/websci/websci2013.htmlRieder13

[8] A.-E. Cano, S. Mazumdar, and F. Ciravegna, "Social influence analysis in microblogging platforms - a topic-sensitive based approach," in *Semantic Web Journal*, 2012.

[9] "Facebook graph apis - resources available at:https://developers.facebook.com/docs/reference/api/search/."

[10] "Facebook query language - resources available at:https://developers.facebook.com/docs/reference/fql/."

[11] "Linguistic inquiry and word count - resources available at:http://www.liwc.net/."

[12] "Restfb facebook graph api - resources available at:http://http://restfb.com/."

[13] D. Terrana, A. Augello, and G. Pilato, "Automatic unsupervised polarity detection on a twitter data stream." in *ICSC*. IEEE, 2014, p. in press.

[14] "Twitter4j - a java library for the twitter api - resources available at:http://twitter4j.org/en/index.html/."

[15] "Internet 2012 numbers - resources available at:http://royal.pingdom.com/2013/01/16/internet-2012-in-numbers/."