Building a Role Search Engine for Social Media

Vanesa Junquero-Trabado
DAMA-UPC
Universitat Politècnica de Catalunya
Jordi Girona 1-3
08034 Barcelona, Spain
junquero@ac.upc.edu

David Dominguez-Sal
DAMA-UPC
Universitat Politècnica de Catalunya
Jordi Girona 1-3
08034 Barcelona, Spain
ddomings@ac.upc.edu

ABSTRACT

A social role is a set of characteristics that describe the behavior of individuals and their interactions between them within a social context. In this paper, we describe the architecture of a search engine for detecting roles in a social network. Our approach, based on indexed clusters, gives the user the possibility to define the roles interactively during a search session and retrieve the users for that role in milliseconds. We found that role selection strategies based on selecting people deviating from the average standards provides flexible query expressions and high quality results.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Information networks*; H.2.8 [**Database Management**]: Database Applications; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

Keywords

Social networks, role search, twitter

1. INTRODUCTION

The explosive growth of people registered in social networks is arousing an interest to understand the patterns of interaction among its users. One comprehensive approach to understand who is who in a social network is to classify the people by the roles that they are playing in the network. The actions performed by individuals in the network are far from random, and people repeat activity patterns that define roles within a social context.

There is a lot of research about identifying particular roles in contexts such as online discussion spaces, Wikipedia and social media [4, 7, 17]. But, most research is focused on analyzing the network and defining roles specific to the network and not on providing fast architectures to retrieve them. First, they analyze the network which is subject of study to find possible roles to detect. Then, they characterize them with some observed features and try to obtain all people that respond to these features [1, 11, 18]. These approaches are dependent on the network and specific to a given role. For instance, research on detecting roles in Twitter has mainly been focused on detecting one particular role such as innovators, celebrities or high quality content producers [4].

WWW 2012 Companion, April 16–20, 2012, Lyon, France.
ACM 978-1-4503-1230-1/12/04.

Research done in other networks like Wikipedia or Youtube has designed a set of rules that classifies people into groups or has clustered people of the network and manually, someone has assigned roles to these clusters [13, 16, 17].

Social role is a well known concept broadly studied by sociologists. In role theory, a role is defined as "those behaviors characteristic of one or more persons in a context" [3]. They are not explicit but are characterized by attributes that emerge regularly within a social context. In this work, we define as a social role to the set of *relevant metrics* that characterize the behavior of certain groups of people within a social context.

In this paper, we describe the architecture of a search engine to perform searches in real time for roles not defined before hand. Our role detection is independent on the network and is not specific to any role. Our approach performs a preprocessing step that consists on representing each person of the network with a feature vector that represents their behavior and their relationships with the other members of the community. Then, it executes a clustering method over the feature vector of people to cluster persons with similar behavior into groups that are indexed. The framework preprocesses the people of the social network and classifies them into clusters, independently of the roles that the user will search afterwards. In real time, users define the query of the role search engine as a set of relevant metrics. We say that a relevant metric is a feature that distinguishes the individuals of that role. The system detects the clusters that are relevant according to the characterization of the roles given. Once the people is classified for a role, the presentation of the results as aggregates facilitate a final step where the user can understand the results and filter out by other metrics. However, this final step is not the focus of the paper and will not be analyzed in our experiments. Clusters provide a conceptual organization of the members of the network. The presentation of results as clusters give to the user a better understanding of the people selected. We found that a clustering process independent of the role, does not have a large impact in the quality of the role assignation.

Our approach provides several contributions: it provides a role query system based on relevant metrics that is flexible and adaptable to any social network. Then, we perform a comparative study between different strategies of identifying groups of people that fit a given role in order to improve the number of people recovered. These strategies are concerned with the normalization data and roles assignment. In our experiments, we take Twitter as a workbench, which is a microblogging service that has emerged as a new medium

to spread rapidly new ideas, trends and events [10]. We show that our system is able to identify groups of people that respond to some important and recognized roles in this network such as celebrities or information propagators.

This paper is organized as follows: in Section 2, we describe the related work; in Section 3, we present the data model and describe the architecture for roles detection; in Section 4, we present the experimental setup; in Section 5, we describe the dataset used for evaluating the system and present the results obtained; and, in Section 6 we conclude the paper and suggest future lines of research.

2. RELATED WORK

Role detection studies in online enviroments started before the emergence of online social networking websites. Nolker et al. studied open discussion bulletin boards and identified two roles that are important to the success of the community: leaders, who spread knowledge and mantain the cohesiveness of the group, and motivators, who keep conversation going [15]. They are defined based on their behavior, conversations and member relationships. Other works focus on a particular network such as Usenet or Yahoo! Groups. In Usenet, some roles are identified: experts, answer people, conversationalists, fans, discussion artists, trolls and lurkers. These roles have been identified through their interaction with other members and their behavioral and structural patterns [7]. The role of the respondent in online discussion groups, who provides helpful and informative responses to other group members' questions [18], is well recognized in this kind of networks. The detection of long-term engaging persons is important since they are key members to mantain alive discussion groups. The contributor role is also appreciated and studied in Yahoo! Groups [1].

There are many proposals for Twitter given that it is easy to obtain data from it. We can differentiate several roles interacting in Twitter such as the mainstream news sources that spread information through the network; the celebrities, who are public figures followed by many persons; or the opinion leaders, who spread widely their opinions and exercise a big influence among other persons in the network [4]. Content in Twitter is produced by hundreds of millions of persons. We can distinguish the most interesting and authoritative author for any topic as another role [16]. But, the relative openness and the incresing widespread interest and growth of these type of networks have attracted a new role: the social spammers [12]. Social spammers use social networks to disseminate malware or to spread commercial spam messages.

Gleave et al. propose qualitative methods to identify an initial set of potential roles and measurements to analyze them [6]. This method is followed in [17] to identify roles in Wikipedia and to define signatures for each role. With these signatures they build a set of rules to classify the people in these roles. Another approach to identify roles is to characterize users with a feature vector composed of certain information related to the user and then, to group users with similar behavior into the same group [13, 16]. On the other hand, [11] also studies roles in Wikipedia. They compute and compare several local metrics, such as the number of articles or the number of comments, and global measures, such as the size of the largest connected component or the mean distance between persons.

Our framework differentiates from the above related work in that it is not specific for a given role because it is the user who defines it. Besides, it is independent on the network because it computes a set of metrics that can be adapted to any social network.

3. ROLE IDENTIFICATION ARCHITECHTURE

3.1 Data Model

We model the dataset as a graph using a generic model of social network, where persons are able to publish and share documents. The schema, which is depicted in Figure 1, can be easily mapped to many networks such as email communications, bibliographic citation networks or online social networks [8]. Since in this paper we experiment with Twitter data, we describe the three types of nodes and five types of edges with examples of this social network:

- Person: is a person registered in the network.
- Document: is a document (tweet) published by a person.
- Tag: is a keyword of a document. In Twitter, they are explicit as a hashtag, which is a convention to create and follow a topic. It is a word prefixed with a '#' character.

We have the following edge types:

- Person-publishes: indicates the person who publishes a document.
- Person-receives: denotes the people that receive a document. In Twitter, the direct communications are tweets whose content is preceeded by a username with the '@' symbol
- Depicts: relates the persons that are mentioned in a document. In Twitter, it is defined by a '@' character followed by the username.
- Knows: states a social relation between persons. In Twitter, we derive the relation from the following-followers list.
- References: the reference relation is created when a message is referencing a previous message. Given that the relation is not explicit in our dataset, we compute it as follows. First, we select all the messages in the dataset that contain the *RT+username* or *via+username* expression. For each message, we pick the messages of the target username and we create the *references* relation with the most recent tweet (with earlier timestamp) that matches at least three words and has more than 75% of the content in common.

3.2 Role identification architecture

We propose an architecture that is able to locate persons of a network that belong to a role in real time. A role query is defined as a collection of *relevant metrics*, which the user of the role search engine finds important for that role. They usually indicate values far from the average of the population for a subset of features, such as having many followers. The persons that verify the conditions of the relevant metrics have the role.

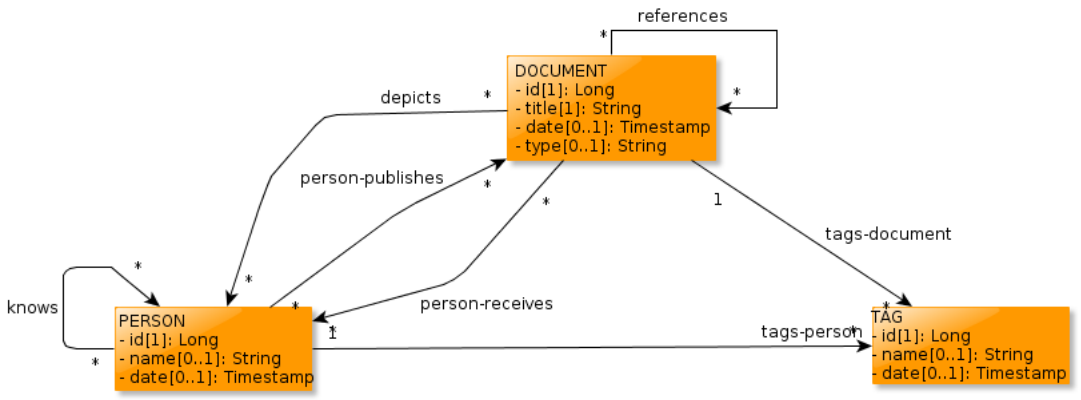


Figure 1: Generic Model

This method has an offline step and online step. The offline step characterizes the persons, normalizes the data and clusters the persons into groups. In the online step, the user queries the system and the search engine performs the role assignment (by means of the seed selection and seed expansion processes).

3.2.1 *Characterization of persons*

Social roles can be defined by the individual behaviors, the relationships among members or a combination of the two. Our architecture is based on defining a very rich set of features that enable the identification of heterogeneous roles. Typically, a role focuses on a small subset of attributes that are peculiar and distinctive of the role. Our current system computes fourteen metrics that describe the activity of the person and its interactions in the network. Our architecture allows for the extension of this set of features. And, as we will see in the experiments, the addition of new features does not have a large impact on the location and recall of the role search engine. Given a person p in the network we define the following features:

- **M1:** Number of people p knows.
- **M2:** Number of people that know p.
- **M3:** Number of reciprocal relationships of p.
- **M4:** Clustering coefficient of the knows relation, which measures how tied are the friends of p.
- **M5:** Average depth of propagation. We compute the reach of the person p in the network. This feature is computed using the following method:
  1. We get all documents published by p.
  2. We get all the persons influenced by the documents retrieved in step 1. We consider that person A influences person B if: B receives a document from A; B references any document published by A; or, B tags any document published by A.
  3. For each person influenced i, we compute the distance between i and p, as the number of edges traversed by the shortest path between i and p using the *knows* relationship.
  4. The average depth is the average of the distances found in step 3.
- **M6:** Maximum depth of propagation computes the highest depth that the influence of p arrives within the network. It is computed as M5, but the step 4 of the algorithm computes the maximum function instead of the average.
- **M7:** Number of messages that p receives.
- **M8:** Number of documents that depict p.
- **M9:** Time in average between actions of influence performed by p. For each influence action between two persons, we compute the difference of time.
- **M10:** Position in average that person p appears in propagation cascades. The cascades are defined over documents and the references relation. For instance, if person B references a document published by person A, and person C references the referenced document of B, we say that A has position 0, B position 1 and C position 2 in the cascade.
- **M11:** Number of publications of p.
- **M12:** Join date of p to the network.
- **M13:** Average number of words in the document published by p.
- **M14:** Percentage of words that exists in a dictionary of the set of documents published by p. The dictionary used is Wordnet[1]. We take this metric as an indicator of the register style of p.

3.3 Data processing

We use *K-means* to cluster persons into k clusters over their feature space with different values of k. The first decision to make is how to normalize the data. Once the data is normalized, we cluster people of similar behavior into groups. The next decision to make is how to assign roles to these clusters. In the following sections we describe the normalization and role assignment strategies evaluated.

[1] `http://wordnet.princeton.edu/wordnet/download/`

Note that the assignation of a person to a role, does not prevent that the person in another query can be assigned to other role.

3.3.1 Normalization strategies

The normalization process takes each attribute described in Section 3.2.1 and converts them into values in a known range. Since the applied clustering method is based on distances between persons, the normalization process allows for a better mapping of the attributes. We implement the following methods:

- **Maximum/Minimum normalization** (Max/Min): Given a value v of a given metric m of the feature vector, we apply the following transformation: $\frac{v-min(m)}{max(m)-min(m)}$ where $min(m)$ is the minimum value of all values of metric m and $max(m)$ is the maximum value of metric m. The result lies between 0 and 1.
- **Log normalization** (Log): we normalize data in logarithmic scale. Then, on the transformed data we apply the same strategy as in Max/Min.
- **Ranking normalization** (Ranking) : we sort the values of each metric in ascending order, and set the attribute as the ordinal number (position in the ranking). Given a value in the ith position of the ranking, its normalized value is: $\frac{i}{N}$ where N is the total number of persons in the database. If there are repeated values, we give the same normalized value to all of them taking the position that is in the middle of the interval.
- **Standard score** (Normal): for each value v of metric m, we compute its standard deviation from the average of all values of m. Unlike the other normalization methods which range from 0 to 1, normalized values with this method can be less than 0 if they are less than the average, or greater than 1 if they are far from the average

3.3.2 Role assignment

After normalizing the data, we run a clustering algorithm on the normalized feature vectors. The next step, is to assign roles to these clusters based on the relevant metrics input by the user of the role search engine. We divide this process into two steps: first, the system selects the clusters that best fit the characteristics of the user query to be the seeds of the role. From these clusters, it computes the centroids of the role and it collects more clusters following the seeds expansion procedure. The process is iterated until it converges. Once the process ends, all the persons in the initial and the expanded seeds are set as belonging to the role.

Seed selection: Each cluster c has an associated feature vector with values $< v_1, ..., v_i >$, derived from the persons assigned to the cluster. The value of v_i is computed as the average value of the metric m_i over all the people assigned to c.

In Figure 2, we show an example that classified the people of a social network in 25 clusters. In this example, the user looks for people that fit a celebrity role, whose relevant metrics are M2 (number of followers) and M8 (number of mentions). Each bar corresponds to the average value of M2 and M8 for each cluster.

We test the following seed selection strategies:

- **Standard deviation** (Sdv) We denote $\overline{m}_i$ as the average value of the clusters for the attribute m_i. For each attribute in a cluster, we compute the standard score with respect to $\overline{m}_i$. The relevant metrics are defined as an interval with the minimum and maximum number of standard deviations that a cluster can deviate from the average $\overline{m}_i$.

  An example input query for locating celebrities is that the relevant metrics (M2 and M8) must be above the average, i.e that M2 and M8 are in the range $(0,\infty)$. Since the dotted lines in Figures 2(a) and 2(b) indicate the mean over all clusters, only clusters 2,3,4,14 and 23 fit the celebrity query. Therefore, Sdv selects the five clusters as seed clusters of celebrity.
- **Selection by the maximum value** (MV): considering an n-dimensional subspace where n is the number of metrics relevant to a given role, we select the cluster that has the largest module in this n-dimensional space. Following the example above, the largest module is given by the first cluster: $\sqrt{0.010^2 + 0.00071^2} = 0.010$. This method selects one cluster as long as there is no more clusters with the same maximum module.
- **Selection by the maximum value on the axes** (MVA): considering an n-dimensional subspace where n is the number of relevant metrics, we select the clusters that have the highest value in each axis. In the example above, clusters 1 and 3 are selected. This method selects n clusters if there are no clusters with the same maximum value.

Seed expansion: The system enlarges the initial seed from the seed selection step by adding similar clusters. The idea is that clusters close to the initial seed are also good candidates to be members of that role. The similarity between clusters is computed using the euclidean distance between the centroids of the clusters considering only their relevant metrics. We test three methods to perform the expansion:

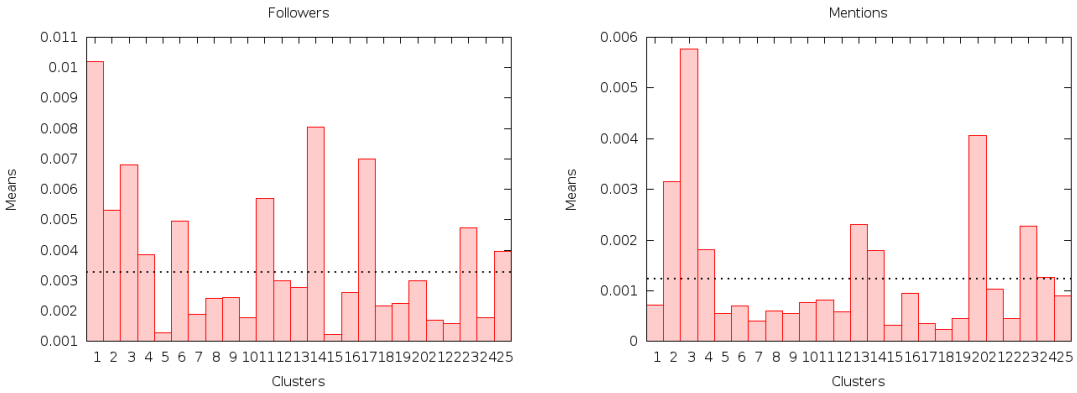
- **Ne**. There is no expansion.
- **Computing the average** (Avg) the centroid for the assigned seeds to the role is recomputed as the average feature vector over the feature vectors of the clusters.
- **Incremental process** (Incr): the centroid for the assigned seeds to the role is the closest seed to the current centroid that has not been centroid before.

4. EXPERIMENTAL SETUP

In this section, we describe all the experimental environment that we use to evaluate our approach. We define the set of roles used, we perform a study of data and we define some metrics that are used to evaluate the quality of the system.

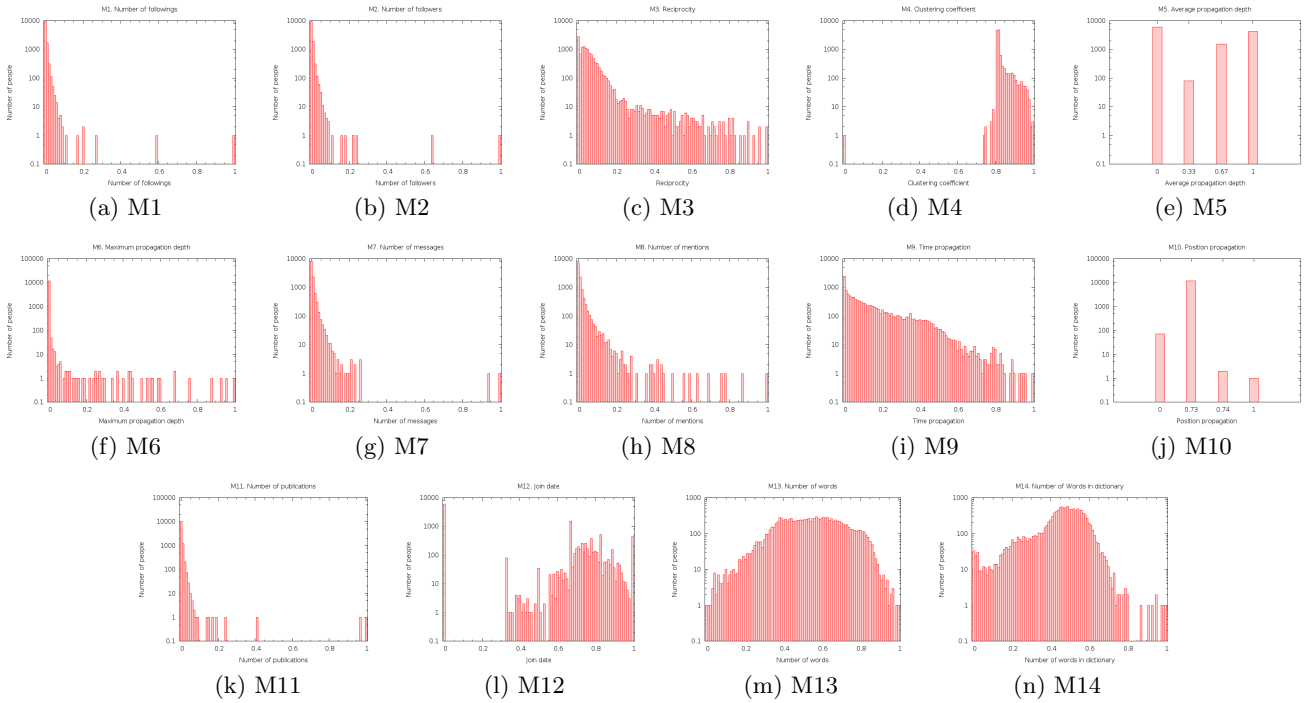
4.1 Definition of roles

Our system is flexible so that we can define any initial set of roles. In order to test our approach, we have chosen four examples of roles that have been seen in Twitter in previous literature. We chose these roles because they are relevant and have been identified by practitioners in several works. We briefly describe each role and the metrics that best capture its behavior:



(a) Number of followers (b) Number of mentions

Figure 2: Example of Role assignment



(a) M1 (b) M2 (c) M3 (d) M4 (e) M5

(f) M6 (g) M7 (h) M8 (i) M9 (j) M10

(k) M11 (l) M12 (m) M13 (n) M14

Figure 3: Data distribution

Celebrities. They are the most followed and mentioned persons, which often are public figures like Lady Gaga or Gisele Bundchen. They attract lots of attention from their audience through conversational interactions since people talk to and about them. Therefore, they are among the most connected but not necessarily the most influential [4]. The relevant metrics used to identify this role are the number of followers (M2) and the number of mentions (M8).

Information propagators. They are mostly news sources and opinion leaders. Research on opinion leaders has found out that communication does not flow to the mass directly but is actually interpreted first by opinion leaders and then forwarded to the rest of people. They act as amplifiers in a process called "Two-Step-Flow of Communication" [9]. They receive information through different sources and spread them through the network. The influence of such opinion leaders is significantly larger than ordinary users of the network. The relevant metrics used are the number of followings (M1), the average and maximum information propagation depth (M5 and M6), the number of publications (M11) and the number of words in their published tweets that exist in dictionary (M14) (this aims at capturing their formality).

Promoters. Ideas and innovations tend to diffuse along social links. A promoter starts a novel idea or trend and then, people in contact with the promoter adopt it. Then, the people in contact with these people also adopt the trend like in a cascade [5]. Promoters belong to the firsts positions of such structure. The metrics used are the average and maximum information propagation depth (M5 and M6) and

position in cascades considering both time and position (M9 and M10).

Early adopters. Some persons play a more active role in distributing content than others, but these influencers are distinct from the early adopters. The early adopters are more susceptible to adopting tendencies earlier irrespective of none or one of their friends have adopted it and do not exercise a big influence over others [2]. The relevant metrics used are the position in cascades considering both time and position (M9 and m10).

Each of these roles have different number of relevant metrics and they can be either positive (we look at people that have as high as possible) or negative (we look at people that have as less as possible).

4.2 Configuration and Evaluation Methods

By default, we configure Sdv with the relevant metrics that we show in Table 1. For each relevant metric of each role, we define a filter that is an interval where the first number is the minimum and the second one is the maximum number of standard deviations. The 0 means that we are in the average and the infinite that we do not have upper or lower limit. The third column indicates the number of persons in our Twitter experimental dataset that responds to this criteria. For MV and MVA we report the results for a distance equal to 0.1 because it gave the best results based on experimentation with our system.

Role	Rel. metric	Persons
Celebrity	**M1:** [0,∞) **M8:** [0,∞)	588
Information Propagators	**M1:** [0,∞) **M5:** [0,∞) **M6:** [0,∞) **M11:** [0,∞) **M14:** [0,∞)	285
Promoters	**M5:** [0,∞) **M6:** [0,∞) **M9:** (−∞,0] **M10:** (−∞,0]	303
Early adopters	**M9:** (−∞,0] **M10:** (−∞,0]	396

Table 1: Roles characterization

In order to evaluate the quality of our system, we generate 150 synthetic persons for each role as follows:

- For each metric, we rank Twitter persons starting from the person with the highest value to the person with the lowest value.
- For each relevant metric of each role, we select a subset of people that is in the top-1% in the ranking. We select one person at random from each subset and we copy the value of the relevant metric into the synthetic person.
- We select another person at random from the whole set of persons and copy their values of the rest of metrics into the synthetic person.

We consider two evaluation metrics: the Synthetic Persons Recovery, that computes the number of synthetic persons that the system is able to recover and the F-measure, which is the harmonic mean between precision and recall. Precision is the number of persons that fit our criteria of role over the total number of persons recovered. Recall is the number of persons that fit our criteria of role and the system recovers over the total number of persons that fit our criteria of role in the network. To avoid subjectivity in the role classification, we consider that a person fits our criteria of role if their values of relevant metrics for a given role are included in the intervals depicted in Table 1. Note that we use less metrics for identifying roles than the total number of metrics in the persons feature vectors. Our objective is to provide a flexible tool that allows the user to recognize a wide variety of roles and to filter them by other metrics than the relevant metrics.

We use *k-means* to group people into clusters. In order to determine k, we compare the Synthetic Persons Recovery with different values of k. We select $k = 500$ clusters since good results are achieved with this value (for more information check the Appendix). In our tests, we use five configurations of role assignment (Sdv+Ne, MA+Avg, MA+Incr, MAV+Avg and MA+Incr), on top of the same clusters.

4.3 Experimental Dataset

We test our framework on a Twitter dataset that includes persons, tweets, following/followers relationships and hashtags[2]. We load and analyze it with the aid of the DEX graph database [14].

The union of the datasets has over 40 milions of persons, 26 milions of tweets and 1,000 milions of followings/followers relationships. Since we combine two sources of data, we made a preprocessing step to select only those people that participate actively in the network. We only keep persons that have at least 25 publications, 20 followers and 20 followings. The total number of active persons is 11,805. The total number of persons is 12,855, considering both synthetic and non synthetic persons.

We studied the distribution of data for each metric. In Figure 3, we see that in general all metrics follow a power law distribution except in a few cases such as the number of words (M13) and the number of words in dictionary (M14) that look like a normal distribution.

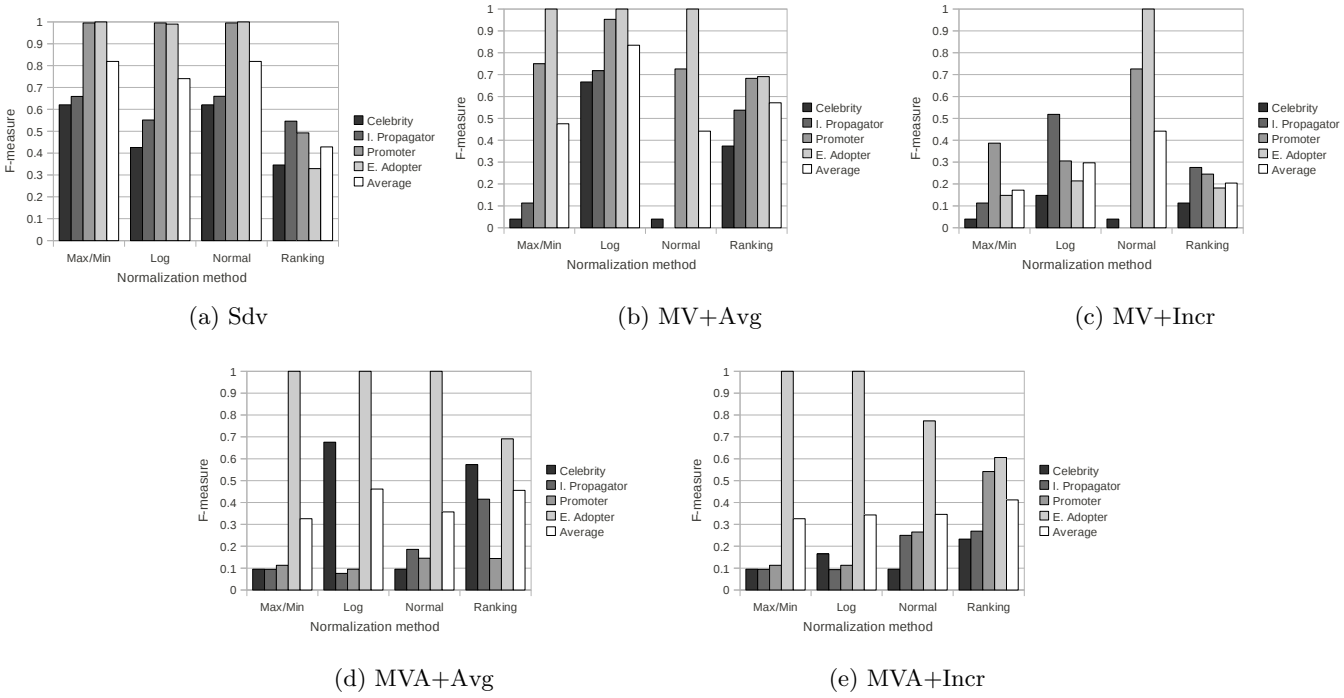
We find that the metrics covered in the dataset have different properties. There are some metrics that have many different values such as M3, M9, M13 and M14 and some that only have four possible values like M5 and M10. There are other metrics where people are concentrated in an interval of values like M1, M2, M4, M7, M11 and M12. Finally, there are some metrics that have gaps between intervals of values like M6 and M8.

5. EXPERIMENTS

In this section we show the results obtained when we test the different normalization data methods and the role assignment strategies.

In Figure 4, we plot for each role assignment method the F-measure results which is the harmonic mean of precision and recall. Precision checks that the the persons in a role are correctly identified, and recall that the algorithm effectively retrieves all people belonging to the role. The x-axis are the normalization methods and the y-axis are the F-measure. In Figure 5, we show for each role assignment method the Synthetic Persons Recovery. This figure shows how good is

[2]Downloaded from `http://snap.stanford.edu/data/bigdata/twitter7/` and `http://an.kaist.ac.kr/traces/WWW2010.html`.



(a) Sdv (b) MV+Avg (c) MV+Incr

(d) MVA+Avg (e) MVA+Incr

Figure 4: F-measure results

our system at identifying people that are very different from the rest. The x-axis are the normalization methods and the y-axis are the Synthetic Persons Recovery.

5.1 Normalization results

In general, normalization methods are dependent on the role assignment strategy applied but the best results are achieved with Max/Min, Log and Normal normalizations with an F-measure in average over the 80%. In Figure 4(a), Max/Min and Normal results are identical. This is due to the fact that *k-means* with this configuration generates the same clusters with both normalizations and when we apply Sdv we are filtering clusters using standard deviations which are the same in both cases. The Early Adopter is well identified by all normalization methods except Ranking. On the contrary, Celebrities and Information Propagators are harder to identify independently on the normalization mehtod.

Regarding to the Synthetic Persons Recovery, the best strategies are Max/Min and Log normalizations recovering more than 90% of the persons in Figure 5(a). Information Propagators are well identified by Log normalization in all cases. Like with the F-measure, Early Adopters are well identified by all normalization methods in almost all cases except with Ranking.

5.2 Role assignment results

In this section we compare the results among the role assignment strategies. As we will see the best strategies are MV+Avg with Log normalization and Sdv with Max/Min normalization.

Sdv centroids selection. In Figure 4(a), we see Sdv obtains a good F-measure, above 80% in average, using Max/Min and Sdv normalizations. These normalizations assign less people to roles than the others. If many people are assigned to roles, then the precision gets worse since more people that do not fit the criteria is considered member of the role. If precision is worse, the F-measure is reduced. We can appreciate this fact observing that with Log normalization the F-measure is worse with Celebrities and Information Propagators, beacuse it assigns much more people than using Max/Min.

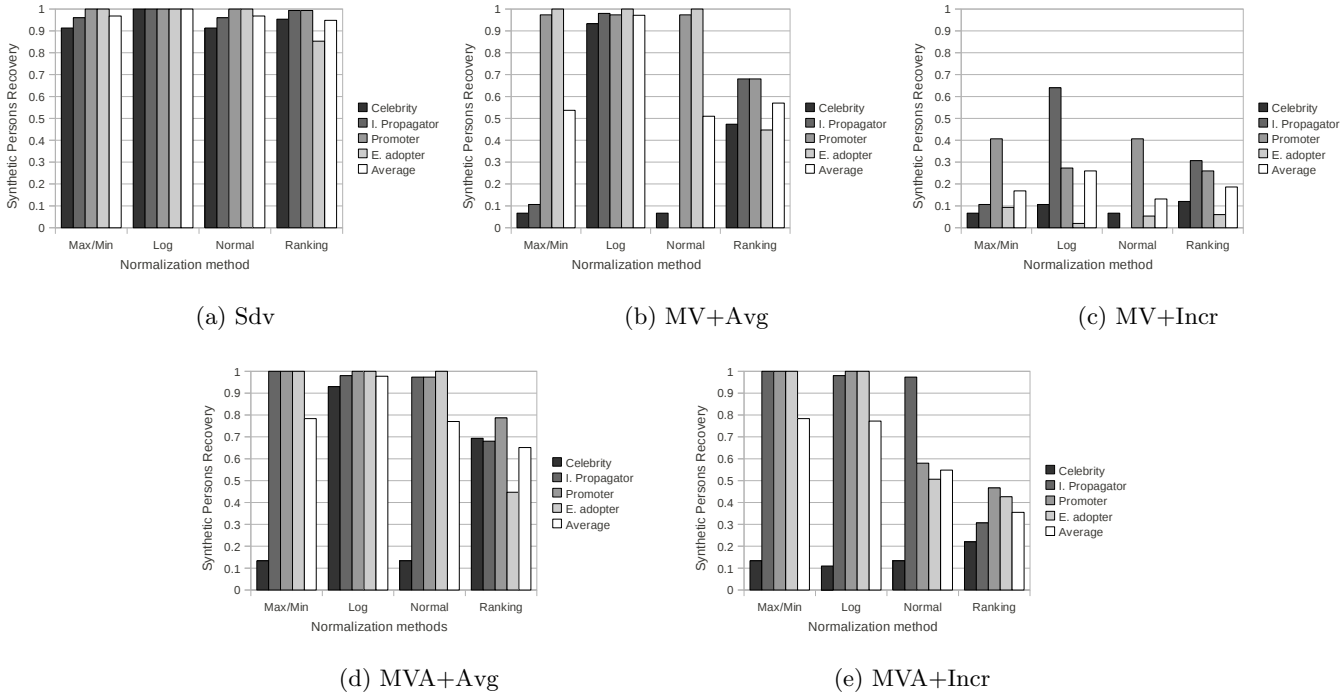
With this role assignment we recover almost all synthetic persons. In the case of Log normalization, we recover all of them. Therefore, this is a very good strategy when we want to search people with very differentiated roles from the rest.

MV+Avg. In this case we select the centroids of each role following the MV+Avg strategy. Note that we only have one seed per role and from this seed we collect clusters that are close to it. In Figure 4(b), we see the F-measure, and in Figure 5(b) the number of synthetic persons recovered for each role.

Log normalization is the strategy that gets the best F-measure (above 80%) followed by Max/Min. Unlike using Sdv, it gets much less people regardless of data normalization because few clusters are selected. The recall is worse because it assigns less people to each role. On the other hand, precision is very good, because almost all people selected fit the role (Figure 4(b)).

Finally, this strategy recovers more than the 90% of synthetic persons with Log normalization. This strategy recovers less people with some normalizations because using these normalizations it assigns few persons into some roles. It is specially critical when it recovers Celebrities and Information Propagators.

MV+Incr and MVA. In this case, we present the F-measure and the Synthetic Persons Recovery with MV+Incr (Figures 4(c) and 5(c)) and with MVA (Figures 4(d), 4(e),



(a) Sdv (b) MV+Avg (c) MV+Incr

(d) MVA+Avg (e) MVA+Incr

Figure 5: Synthetic Persons Recovery

5(d) and 4(e)). The F-measure with these assignments is poorer than for Sdv and MV+Avg. MVA uses more seeds depending on the number of relevant metrics. When it expansions gets more clusters. The more clusters we get, the more people is assigned, reducing precision. When it comes to the number of synthetic persons recovered the best strategy is MVA+Avg with Log normalization recovering more than 90% of the synthetic persons in each role 5(d).

Comparison of role assignment strategies: The best strategies are Sdv and MV+Avg with an F-measure on average above 80%. They are specially good at recovering people that have a very differentiated behavior from the rest recovering more than the 90% of synthetic persons for each role. The rest of the strategies fail at identifying some role and hence their average scores are poorer.

5.3 System robustness

In this section we show two more experiments that aim at testing the robustness of our approach. The first one shows the results when we are interested in being more restrictive with people we select. The second one shows the impact when we use different number of metrics.

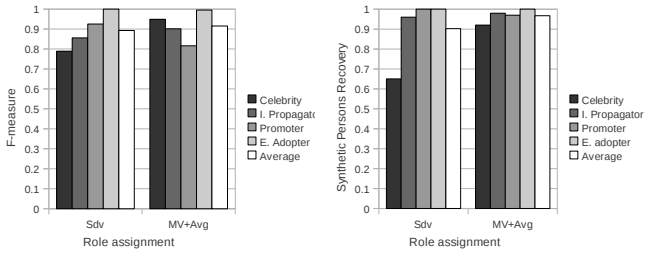
5.3.1 More restrictive roles

We change some of the revelant metrics of our roles as shown in Table 2. We are more demanding and we want to select people that differ more from the rest. With this experiment, we test the robustness of the role selection methods when relevant metrics are changed. We perform these experiments using Sdv with Max/Min normalization and using MV+Avg with Log normalization because they give the best F-measures in Section 5.2.

If we compare Figure 6(a) with Figure 4, we observe that the F-measure is better, over 90%. This result indicates that

Role	Rel. metric	Persons
Celebrity	**M1:** [1,∞) **M8:** [1,∞)	313
Information Propagators	**M1:** [0,∞) **M5:** [1,∞) **M6:** [1,∞) **M11:** [0,∞) **M14:** [0,∞)	195
Promoters	**M5:** [1,∞) **M6:** [1,∞) **M9:** (−∞,0] **M10:** (−∞,0]	202
Early adopters	**M9:** (−∞,-1] **M10:** (−∞,-1]	396

Table 2: Restrictive Roles characterization



(a) F-measure (b) Syn. Per. Rec.

Figure 6: F-measure and synthetic person recovery with restrictive roles

our approach is better when people to be identified have patterns of behavior more differentiated that the rest of persons in the network. Considering the synthetic user benchmark,

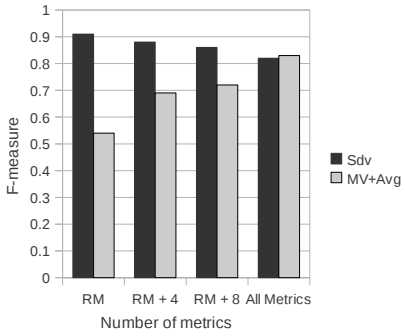


Figure 7: F-measure

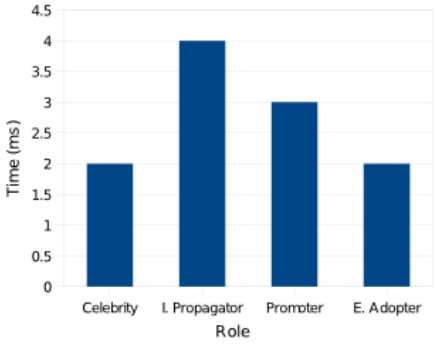


Figure 8: Execution time

we observe in Figure 6(b) that we recover about the 90% of synthetic persons in average with both role assignments.

In a nutshell, the interest range of the relevant metrics does not alter significantly the results. Both Sdv and Mv+Avg keep a good F-measure independently of the restrictions on the relevant metrics.

5.3.2 *Varying the number of metrics*

In this experiment, we modify the number of features computed for the persons in the database. Giving simpler feature models for the people in the network, we check the impact of having a large number of metrics on the role selection policies. First, for each role, we perform all the role search procedure like in Section 5.2 on a database where only the relevant metrics are taken into account. Then, we add four and eight features, at random, into the person feature vector. We perform this experiment using Sdv with Max/Min normalization and MV+Avg with Log normalization, which are the strategies that performed better in previous experiments. In Figure 7, we show the average F-measure computed using the F-measures obtained for each role and we compare it with the results obtained in Section 5.2 (last column in Figure 7). RM in the figure means Relevant Metric.

If we increase the number of metrics, then using Sdv does not have a large impact on the quality of our role search engine. However, with MV+Avg method, the F-measure is worse with two metrics and it increases as we increase the number of metrics. With fewer metrics, the distance between clusters increases, and therefore it needs to be reconfigured. We performed the same experiment but increasing the distance from 0.1 to 0.2 for 2 and 4 metrics. We get an F-measure close to 0.7. It seems that although MV is able to obtain good results, it is necessary to adjust its parameterization for the number of metrics considered in the feature model.

To sum up, we conclude that using Sdv role assignment gets good results independently on the number of metrics or the restrictions we put into roles. However, MV+Avg is dependent on the problem and it needs to be reconfigured if we modify the number of features computed for the persons. This fact is a decisive factor to prefer Sdv rather than MV for the role search problem.

5.4 Execution time

In this section, we measure the execution time that takes our approach to assign roles to clusters. The user of the system introduces the query and only waits for the role assignment. This is an advantage since all the preprocessing step is unseen for her. In our implementation, the computation of the person feature vectors took hours since we did not make an effort to optimize the process. For our dataset, the clustering takes a few dozens of seconds and makes unsuitable computing it for each user query. Besides, such online clustering could make the system difficult to scale. Our architecture hides all the execution time spent computing feature vectors and clustering as a preprocessing step, which is invisible to the user.

In Figure 8, we show that the order of magnitude of role assignment is in milliseconds. The system takes between two and four miliseconds to assign roles to clusters. Therefore, our system is able to identify roles into a social media network very fast. It can be used in an online application that studies the structure and organization of a social media to detect different roles interacting within the network. If we want to accelerate the process even more we can use an R-Tree structure to index the cluster, which is often used for indexing high dimensional data [19].

6. CONCLUSIONS AND FUTURE WORK

We have proposed an architecture that provides fast answers to identify social roles in a network. In the paper, we introduce the concept of a relevant metric to express role related queries. The relevant roles are issued to the search engine by the user to obtain a group of people that follows a certain profile. Our system clusters offline the persons in the network, and thus provides online role assignations for the clusters found in milliseconds. This procedure based on grouping people into clusters rather than classifying people, allows users to better analyze the data and facilitates the task of finding a more definite role. We found that role selection strategies based on Sdv gives good expresivity to write role queries, is stable and does not need parameterization, and have high precision and recall.

The classification of people into roles for marketing, opinion diffusion or advertising among others, is gaining importance in the business process of many companies. Our framework can serve as a starting point to several future work directions. We believe that one architecture such as the presented in this paper can be used to aggregate heterogeneous social network resources and explore the roles of people in such networks.

7. ACKNOWLEDGMENTS

We thank Nuria Trench and Miquel Angel Aguila for their help developing the prototype. The members of DAMA-UPC thank the Ministry of Science and Innovation of Spain and Generalitat de Catalunya, for grant numbers TIN2009-14560-C03-03 and GRC-1087 respectively.

8. REFERENCES

[1] L. Backstrom, R. Kumar, C. Marlow, J. Novak, and A. Tomkins. Preferential behavior in online groups. In *WSDM*, pages 117–128. ACM, 2008.

[2] E. Bakshy, B. Karrer, and L. A. Adamic. Social influence and the diffusion of user-created content. In *ACM Conference on Electronic Commerce*, pages 325–334. ACM, 2009.

[3] B. J. Biddle. Recent developments in role theory. pages 67–92. Annual Review of Sociology, 1986.

[4] M. Cha, H. Haddadi, F. Benevenuto, and P. Gummadi. Measuring user influence in twitter: The million follower fallacy. In *ICWSM*, 2010.

[5] R. Ghosh and K. Lerman. A framework for quantitative analysis of cascades on networks. *CoRR*, abs/1011.3571, 2010.

[6] E. Gleave, H. T. Welser, T. M. Lento, and M. A. Smith. A conceptual and operational definition of 'social role' in online community. In *HICSS*, pages 1–11, 2009.

[7] S. A. Golder and J. Donath. Social roles in electronic communities. In *AOIR*, 2004.

[8] V. Junquero-Trabado, N. Trench-Ribes, M. A. Aguila-Lorente, and D. Dominguez-Sal. Comparison of influence metrics in information diffusion networks. In *CASoN*, pages 31–36. IEEE, 2011.

[9] E. Katz. *The Two-Step Flow of Communication: An Up-To Date Report on an Hypothesis*. The Bobbs-Merrill Reprint Series in the Social Sciences, S137. 1957.

[10] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *WWW*, pages 591–600, 2010.

[11] D. Laniado, R. Tasso, Y. Volkovich, and A. Kaltenbrunner. When the wikipedians talk: Network and tree structure of wikipedia discussion pages. In *ICWSM*. The AAAI Press, 2011.

[12] K. Lee, J. Caverlee, and S. Webb. Uncovering social spammers: social honeypots + machine learning. In *SIGIR*, pages 435–442. ACM, 2010.

[13] M. Maia, J. Almeida, and V. Almeida. Identifying user behavior in online social networks. In *Proceedings of the 1st Workshop on Social Network Systems*, SocialNets '08, pages 1–6, New York, NY, USA, 2008. ACM.

[14] N. Martínez-Bazan, V. Muntés-Mulero, S. Gómez-Villamor, J. Nin, M. Sánchez-Martínez, and J. Larriba-Pey. Dex: high-performance exploration on large graphs for information retrieval. In *CIKM*, pages 573–582, 2007.

[15] R. D. Nolker and L. Zhou. Social computing and weighting to identify member roles in online communities. In *Web Intelligence*, pages 87–93. IEEE Computer Society, 2005.

[16] A. Pal and S. Counts. Identifying topical authorities in microblogs. In *WSDM*, pages 45–54, 2011.

[17] H. T. Welser, D. Cosley, G. Kossinets, A. Lin, F. Dokshin, G. Gay, and M. Smith. Finding social roles in wikipedia. In *Proceedings of the 2011 iConference*, iConference '11, pages 122–129, New York, NY, USA, 2011. ACM.

[18] H. T. Welser, E. Gleave, D. Fisher, and M. Smith. Visualizing the signatures of social roles in online discussion groups. *The Journal of Social Structure*, 8(2), 2007.

[19] D. White and R. Jain. Similarity indexing with the ss-tree. In *Data Engineering, 1996. Proceedings of the Twelfth International Conference on*, pages 516 –523, feb-1 mar 1996.

APPENDIX

A. K-MEANS PARAMETRIZATION

This section describes the cluster analysis performed to determine k value for *kmeans*. Figure 9 shows the number of synthetic persons recovered when we increase k using Sdv with each normalization method.

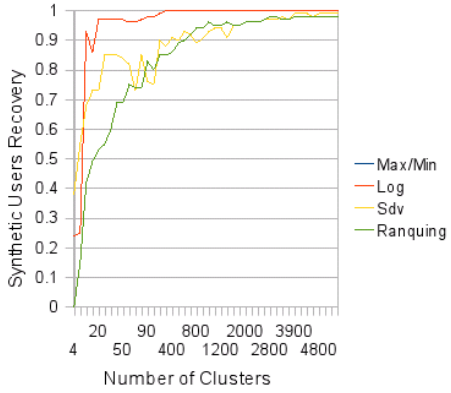


Figure 9: Synthetic Persons Recovery for varying values of K

K is relatively stable from 500 clusters for all normalization methods. However, Log normalization achieves a very good Synthetic Persons Recovery with few clusters whereas the other normalizations need much more clusters to obtain a Synthetic Persons Recovery over the 90%. Finally, Max/Min and Sdv normalizations have exactly the same Synthetic Persons Recovery since *kmeans* gives exactly the same clusters as we mentioned before.

If we use more clusters, we get better precision since we are closer to have a classification problem. Nevertheless, the more clusters we have, the more difficult is for the user to manage such volume of clusters and more people with the same behavior are disperse into different groups. On the contrary, if we have less clusters we loss precision but we facilitate the user the task of searching and refining people fitting a role achieving good results. We select 500 clusters because we get good results and the user of the system can easily manipulate them.