



HAL
open science

Discovering relationship types between users using profiles and shared photos in a social network

Elie Raad, Richard Chbeir, Albert Dipanda

► **To cite this version:**

Elie Raad, Richard Chbeir, Albert Dipanda. Discovering relationship types between users using profiles and shared photos in a social network. *Multimedia Tools and Applications*, 2013, *Multimedia Tools and Applications*, 64 (1), pp.141-170. 10.1007/s11042-011-0853-7. hal-00665036

HAL Id: hal-00665036

<https://hal.science/hal-00665036>

Submitted on 1 Feb 2012

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.

Discovering relationship types between users using profiles and shared photos in a social network

Elie Raad · Richard Chbeir · Albert Dipanda

© Springer Science+Business Media, LLC 2011

Abstract In this paper, we propose a new approach to discover the relationship types between a user and her contacts in a social network. This is of key importance for many applications in the domain of photo sharing, privacy protection, information enriching, etc. Our approach is based, on one hand, on information extracted from users' profiles and their shared photos, and, on the other hand, on a set of predefined rules validated by the main user before being mined and derived according to her preferences and social network content. The contribution of our method is twofold: 1) it is user-based enabling the user to set her preferences and give her feedbacks on the derived rules and results, and 2) it is multi-criteria that exploits and combines several attributes and features from user profiles and shared photos respectively. It also allows the user to define new relationship types. We conducted a set of experiments to validate our approach. The obtained results show the accuracy of our approach in different scenarios.

Keywords Social networks · Link type prediction · Photo based inference · User profile · Photo metadata · Rule mining

1 Introduction

Online social networks provide users, within their platforms, powerful ways to interact with other users through different forms and modalities. Consequently, each user can search and check the profiles of her social network members (for various reasons), exchange messages with some of them, publish some photos, and post comments on shared photos, etc. Naturally, users tend to shape their online contacts in the same way as their offline or real life contacts. For instance, a user can be connected to different types of contacts such as friends, family members, colleagues, etc. However, identifying and defining these different types of relationships in social networks remain not fully explored since:

- Several networking sites provide the users with an exclusive relationship link type with their contacts. It is common that some users indicate other people as friends even

E. Raad (✉) · R. Chbeir · A. Dipanda
LE2I-CNRS, Bourgogne University, Dijon, France
e-mail: elie.raad@u-bourgogne.fr

- though they do not particularly know or trust them [6]. Treating all the contacts in the same way, without differentiating one user from another, is an unsafe and restrictive practice since users need sometimes to share some data (notes, photos, videos, etc.) and interact with some members of their social network but not with all of them. Indeed, users quit such networking sites when their bosses, friends, and family are all presented with the same online persona [12]
- Some social networking sites provide the possibility to define manually how a user knows each of her contacts. However, most of the time, this option is skipped by the users and only the connection existence is indicated [14]
 - Measuring the closeness between users by computing the link strength cannot reveal the type of the relationship shared by users. The frequency of interaction is highly correlated with relationships. It can provide some insights about relationships whose types are already identified. For instance, people at work interact with colleagues more than with friends or family members. On the other hand, outside the work they tend to interact more with friends and family members than with colleagues [24].
 - People relationships are not always symmetric or bidirectional (if *Mary* loves *Patrick*, it does not mean that *Patrick* loves *Mary*, and vice versa). In most of current solutions, the assumption is that people have bidirectional relationships (*know*, *workwith*, etc.)
 - Users' relationships dynamically evolve and change over time. A friend becomes a family member, a colleague becomes a friend, a friend becomes a colleague, etc. Updating the relationships that a user has with her contacts requires constant maintenance which is a tedious and a time-consuming duty [5].

Taking into consideration the flexible nature of social networks, a user must be able to potentially deal with every modification of her contact list, and every relationship's type variation of a contact. For all these reasons, we believe that a user needs support to identify/define (at least semi-) automatically relationship types that she shares with her contacts in a social network. This would certainly help enhance the level of personal privacy protection, facilitate its personalization, and contribute positively in the organization of users' profile contents and its simplification as an action toward a better profile management. However, such functionality is not possible within current approaches.

In this paper, we address this problem by leveraging the characteristics and features of social networks, namely by using, on one hand, the available information within users' profiles and, on the other hand, the embedded information extracted from shared photos in which users' contacts are depicted so to measure and qualify the types of relationship between users. To that end, a set of rules is applied on profiles and on a collection of shared photos. Those rules are initially based on common sense and tuned by the user (labeling the types of relationships, personalizing the values of some attributes and thresholds, etc.) before being extended using the Apriori mining algorithm. Our approach goes beyond analyzing traditional communication data and classification. Our contribution consists into constructing, for a user, different social graphs of relationships where the connection links are enriched by their types. Within each graph, contacts with the same type of relationship are grouped together.

The remainder of the paper is organized as follows. In Section 2, we present related works. Section 3 is dedicated to present our data model and relationship type discovery framework as well as our default basic rules. Section 4 presents our prototype and the battery of experimental tests conducted to validate our framework. In Section 5, we conclude this study and draw some future directions.

2 Related works

Within the huge amount of information available on the web [17] (personal data, photos, emails, etc.), link-based operations have been applied on various sources of information such as email archives [32], hyperlinks implicit social structure [1], web citation information [23], web applications [22], etc. In the literature, link-based operations can be grouped into three categories [36]: 1) *link prediction*, consisting of predicting the existence of a link, 2) *link regression* allowing to show how a link is rated by a user, and 3) *link classification* or *link type prediction* allows to label/define a link between two users. The first two categories are out of the scope of this work and thus won't be detailed further in the following sections. It is to be noted that while most of the work in the area of social interaction has focused on measuring the strength of the links between connected individuals ([18, 22, 28, 31]), only few studies have focused on identifying the type of a link connecting entities [4]. Such approaches can be categorized into two groups: *web-based and domain-specific approaches* and *photo-based approaches in personal album* that we detail below.

2.1 Web-based and domain specific approaches

Most of the methods, mainly in social networks, make use of web search engine to identify the type of relationship among entities. In essence, all these approaches [16, 20, 21, 25, 34] are domain specific and are applied on a particular type of social networks.

In [21], a social network mining system called *Polyphonet* is proposed. It uses search engines to extract social networks and to infer the strength and the type of links among entities. *Polyphonet* can mainly identify four types of relations which are: co-author, co-lab, co-project, and co-conference. This is done by computing the number of co-occurrences between two persons' names on the web. To compute the strength between the two names, the number of returned hits/pages is measured. To infer the relationships' type, classification rules are obtained by manually assigning the correct labels to pages. Then, the system extracts the top ranked search results between two names and applies text categorization on those pages. At this phase, pages can be classified over classes of relations. Several indices, as stated in [21], are used to measure this such as matching coefficient, mutual information, Dice coefficient, Jaccard coefficient, overlap coefficient, and cosine.

In [16], the authors used search engines and mining techniques to extract social networks. Two social networks are used: artists of contemporary art, and famous firms in Japan. The relationship identification is based on the co-occurrence of names on the web. However, the authors argue that using only name co-occurrence becomes ineffective mainly in two cases: 1) when two entities co-occur universally on numerous web pages, and 2) when applied to inhomogeneous communities. To avoid this problem and to efficiently identify the relation between two firms for example they propose to add a relation keyword to the search query to emphasize on a specific relationship. In the case of inhomogeneous community, such that of artists, a search query is made, and then an adaptive tuning of threshold is applied. This method takes into consideration the fact that even though the relation is observed as weak it might be important for the particular actors. Therefore, two criteria are implemented: 1) links are created when the link score is higher than a given threshold, and 2) links are created based on subjective importance even if their score is less than the defined threshold. Hence, this satisfies the condition of having a number of links greater than a minimum number (isolated nodes) and lower than a maximum number (nodes with a big number of links).

In the work presented in [25], the authors propose a method to automatically extract the labels that describe relations among entities in social networks. The approach is applied on a researcher social networks and political social networks. To discover the type of relationship between two entities, the system extracts from web pages the surrounding local contexts in which those entities co-occur. Common keywords are collected and ordered according to TF-IDF¹ “Term Frequency—Inverse Document Frequency” [29] based scoring. Then, entities are clustered according to the similarity of their context. The representative terms in a cluster are extracted as labels to describe the relations of each entity pair in the cluster.

The authors in [20] propose to detect the relationship type between named entities in an open domain where Friend Of A Friend (FOAF) information and interpersonal relations are not available. The relation labeling starts by submitting two entities to a search engine and then by extracting cue patterns from the returned results. Cue patterns, also called local contexts, include person names, organization names, location names, and some other noun phrases. Then, each cue pattern is submitted to the Open Directory Project (ODP), a human edited directory, which will return the top N taxonomy paths. A directed graph is built using the returned taxonomy paths. After that, critical nodes or potential categories are extracted using a web page ranking algorithm such as PageRank, HITS, or Markov chain random process. At the end, most representative snippets are selected and used to extract potential relationships according to the defined critical nodes.

In the work of [34], a time-constrained probabilistic factor graph model (TPFG) is proposed to model the advisor-advisee relationship as a mining problem using a jointly likelihood objective function. The input of the graph model is the DBLP Computer Science Bibliography Database.² First, a homogenous graph is constructed where an edge is created between every author’s node and the publication node. Some assumptions based on common sense are used to remove unlikely relations of advisor-advisee. Two measures are used to reflect the correlation of the two authors and the imbalance of the occurrences between the advisor and the advisee. Within those two measures, the time factor is incorporated. Second, a dedicated learning algorithm developed to infer the TPFG graph is adopted. This graph is constructed to integrate local features like the relationship type and the start/end time of this relationship. By maximizing the joint probability of the factor graph, the relationship is inferred and the ranking score for each relation edge on the candidate graph is computed.

Most of the presented approaches [16, 20, 21, 25] proposed for relationship type discovery are based on web search engines which is really restrictive since they deal, in a way or in another, with communities having information already published on the web (as most of the information available on the web about individuals is information published for professional reasons). Hence, they can be only applied on communities in specific domains (e.g., researchers, politicians, famous firms, artists, etc.). Furthermore, corresponding algorithms, based on heuristics, are only applicable within a particular type of community [34]. Although that the retrieved web documents are domain depended, they are highly heterogeneous, often unstructured, and may contain uncontrollable information that can make the mining process completely inaccurate (advertisements, popups, etc.). Therefore,

¹ TF-IDF is commonly used in Information Retrieval (IR) field as a token-based method to compute the similarity between sentences. Usually, it represents a document as a weighted vector where each term weight is computed in function of the number of its occurrences within a document as well as its occurrences in the entire corpus.

² <http://www.informatik.uni-trier.de/~ley/db/>

there is a need to propose new alternatives that are not only limited to one type of community and are not based on search engines but can take into consideration users who do not have information published about them on the web (such those of social networks).

2.2 Photos based approaches in personal album

The intuition behind this category of approaches is that when two persons are depicted together in a photo, this implies that they share some social connection. Approaches that propose to predict the type of relationship from users-generated data, especially from the profile information and the published photos, are still rare.

The work presented in [11] is a preliminary investigation in measuring social networks through the natural social activity of being depicted in photos. A weighted graph is formed based on people co-appearance information. The edge weight, between two persons, is computed in function of two parameters: 1) the number of photos in which the two users appear (this directly affects the strength of the link), and 2) the number of persons depicted in a photo (this inversely affects the relationship strength).

In [35], the authors propose to detect social clusters from consumers' photos. A scheme is proposed to construct a weighted undirected graph by examining the co-occurrence of individuals in photos. A closeness measure is defined here that takes into consideration different elements in the photo: the distance between two faces, the number of faces, and the number of co-appearances in other photos. To detect the embedded social clusters, a graph clustering algorithm that maximizes the modularity of the graph partition is applied on the constructed graph.

The approach described in [30] proposes to discover the relationships (relatives, friends, child friends, and acquaintances) between the owner of a set of photos and the persons appearing within these photos. Two sets of rules are adopted here: 1) the first set of rules is qualified to be hard rules since related rules are considered to be always true, and 2) the second set of rules is qualified as soft (since related rules are not be always true) and generated from a training set (a photo collection where the relationship type is known). The inputs of this model are the names of the persons, their ages, and genders. Learning on soft rules and inferring relationship types are based on a statistical relational model called Markov Logic.

In [37], the proposed approach estimates the relationship type among people that appear in a collection of photos. From a family photo collection, a relation tree is constructed, and relationship types such as family, extended family, and friend are derived. To build the tree, the approach is mainly based on face analysis techniques that include face recognition, clustering, and similarity methods. Besides those techniques, the approach includes the use of age and gender, co-occurrences and relative position of persons, and timestamps. The relationship estimation starts by applying the face analysis technique to build clusters and to gather information from photos. Then, observations and knowledge from experiments are applied to these photos in order to identify the relationships.

The described approaches in this category are interesting but suffer of several drawbacks. First, they only consider one subset of attributes available in photos. The input is mainly restricted to the following attributes: age, gender, co-occurrences of persons. Considering only those attributes would obviously limit the capacity of the system to exploit other rich information available able to enhance and facilitate the discovery of all relationship types. Second, users' profile, or the profile of the owner of the photo album, is not taken into consideration. Third, they are not user-based. In [30], the authors define an interesting approach to discover relationship between users within consumer photos.

Although, the approach defines a set of hard and soft rules to discover the relationship types, it assigns all the soft rules with the same weight regardless of the users' context. Consequently, this approach is not able to adapt to users by considering their photo collections. Last but not least, the work described in [37] is only restricted to visual features. This means that the discovery of the type of relationships is limited to predefined and not extendable rules obtained from experiments and observations. In Table 1, we summarize and compare the main characteristics of the existing methods.

As observed, within the scope of social networks, approaches that identify social relationships based on users' photo are rare. Existing approaches deal generally with consumer photo collections. Consequently, it is required to have a photo-based approach to enrich relationship types of social network users. It is very likely that the use of a wider set of information embedded in photos' multifaceted attributes can provide very useful information to be exploited. Additionally, user's profile would certainly help to obtain additional useful information as well as the user preference and context.

3 Our photo-based inference approach for relationship type discovery

Defining/identifying relationship types between users of a social network is not an easy task. Even when the relationship types are explicitly assigned by the user (which is rarely the case since users usually skip this step), maintaining the relationship types updated is complex, time consuming, and tedious. A possible way to do that would consist of comparing (the same or similar) values of some particular attributes, part of the user profiles, and consequently inferring some knowledge from the comparison result. However, this would be limited to the set of attributes to be used for comparison in the two profiles. To overcome this, we propose in this study a rule-based user-oriented approach able to exploit several users profiles' attributes with the set of multimedia data shared within the social network of a given user. It is to be noted that 80% of shared information between users on social network is multimedia (photos, songs, videos, etc.) representing mainly personal data³ which could be very fruitful in discovering relationship types between users. Our aim is to discover the relationship types between a given user and her social network contacts (as illustrated in Table 2—Precision and Recall values with different confidence scores Fig. 1). For instance, a friend can also be a colleague and a family member. Although computing the closeness (link strength) between two users is interesting, it is out of the scope of this study.

In order to describe our approach, we first present our data model before detailing our framework.

3.1 Data model

In this section, we present our data model and define the different concepts used in the context of our relationship type inference approach based on *user profiles* and *shared photos*.

Definition 1 Star Social Network: is a set of entities representing the set of users connected using a set of relationships to a given user u_0 . It is defined in our study as a directed star graph:

$$G : (U, R, \pi)$$

³ <http://www.hesge.ch/heg>

Table 1 Comparative study

Approaches	Used attributes	Visual features	Relationship between two persons		Approach		Extended rules	Users' contexts		Persons not in photos	User preferences and feedbacks
			Type	Closeness	Rule based	Graph based		Mined rules	Rules' weight		
Golder et al. [11]	frequency of co-occurrence	No	no	yes	no	yes	-	-	-	no	no
Wu et al. [35]	co-appearance, face distance, number of person in a photo	yes (face detection and clustering)	no	yes	no	yes	-	-	-	no	no
Singla et al. [30]	age, gender, co-appearance	No	only one	no	yes	no	yes	no	no	no	no
Zhang et al. [37]	age, gender, co-appearance, time	yes (face analysis)	only one	no	yes	no	no	no	no	no	no

where:

- U is a set of users. Each user $u \in U$ is defined as $u: uid \cup (a_1, \dots, a_n) \cup (m_1, \dots, m_n)$ where a_i represents an attribute to describe the *user profile* (e.g., name, phone, etc.) in the social network, and m_j represents a multimedia attribute to describe a *shared multimedia object* (e.g., photo, song, video, etc.). These two categories of attributes will be detailed later on.
- $R\{\}$ contains the set of relationship types adopted in the social network (e.g., *knows*, *friendof*, *colleagueof*, etc.).
- $\pi: \{u_0\} \times U \mapsto r$ where $r \in R$, is a function that assigns a relationship type r between a given user u_0 and other users. For instance, *knows*(Dupont, Alice) means that the given user Dupont knows Alice. Here, the relationships are dependent to the given user and thus directed accordingly (as illustrated in Fig. 2).

We note by G_r a sub-graph of G where only the relationship type r is used between users:

$$G_r : (U, r) \quad \square$$

Definition 2 User Profile: allows to describe the characteristics of a user in the social network *via* a set of attributes $(a_1:v_1, \dots, a_n:v_n)$ where each attribute a_i is part of the vocabulary used for describing profiles, and each value v_i is assigned to an attribute such as *name*, *image*, *projects*, *work*, etc. In this work, we adopt FOAF,⁴ among many other vocabularies, to represent a social network profile. It stands for Friend-Of-A-Friend and defines a set of attributes, grouped into categories as shown in Fig. 3. FOAF is a machine-readable semantic vocabulary that describes people, their relationships, their projects, etc. FOAF documents are written in XML syntax and adopt the conventions of the Resource Description Framework (RDF). In reality, FOAF is considered as the richest vocabulary to use in terms of describing users' profiles and has currently become a widely accepted standard since many large social networking websites propose a FOAF profile for their users [10]. \square

In this study, we only focus on the personal photo-oriented multimedia objects: *photo*, *salient object*, and *photo album* that depict one or several people within the captured scene (s). It is to be noted that FOAF provides two attributes used to represent photos: *img* and *image*. While the former attribute is used to represent a photo of a person (like a homepage photo), the latter attribute can be used for any other representation (e.g., animal, thing, etc.). In our study, *image* attribute is used to represent all the remaining photos in the profile.

Definition 3 Photo: is a multimedia blob object that contains embedded attributes with descriptions that illustrate and give important information about the captured scene. More formally, we represent a photo in the following way

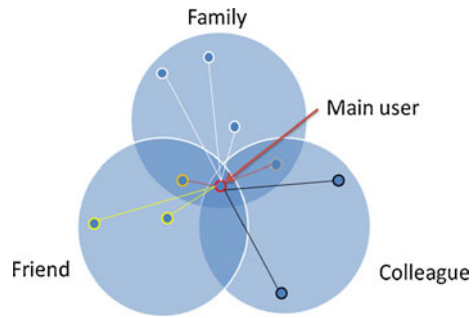
Photo: (pid, meta, caption, so)*

where:

- *pid* represents the photo Universal Resource Identifier (URI).
- *meta* or metadata is the set of technical descriptions embedded in the photo that takes the form of data about data. Here, metadata can be defined as information not related to the semantic content of the photo (author, location, date, etc.). Adding metadata at the creation time requires no effort. Actually, recent digital devices provide such functionalities. This is

⁴ <http://xmlns.com/foaf/spec/>

Fig. 1 A social network of a main user showing how one or more relationship types can exist with her contacts



considered as a best practice if all the information related to the date, the place, and other technical characteristics are available when capturing the photo. Here, we assume that photos use the Exif⁵ standard to extract the following metadata in a photo: *Date*, *Time*, and *Location*. We choose Exif because most of digital cameras store automatically such information in the header of the images using this standard. In Exif, captured images are described using a predefined set of attributes in terms of the context, the environment, and the technical characteristics (exposure time, resolution, focal length, shutter speed, etc.). The value of the *Location* metadata is extracted from the following attributes: *exif:gpsLatitude*, and *exif:gpsLongitude*. The values of the *Date*, and the *Time* metadata are extracted from the following attribute: *exif:dateTime*.

- *caption* is the textual description provided by the main user as an image caption. Usually, an image caption gives some relevant information about the image in terms of places, objects, people, etc. which could be useful and helpful in the process of discovering the type of relationship between users. In this study, the caption of a photo is represented using the Dublin Core (DC) standard,⁶ which is a compact metadata standard used for cross-domain information resource description. It is simple to use and easy to implement thus allowing the non-specialist to use it (librarians, researchers, and music collectors, etc.).
- *so** is the set of salient objects depicted in the photo. □

Definition 4 Salient Object: represents a person (face) in a photo. Each salient object in our study must have a match with one of the profiles (and precisely with images) of the social network so it can be identified. Faces are particularly important in our model not only to identify the related people (when applying face detection and recognition algorithms) but also to extract other interesting information related to the age and gender since several visual (age and gender) estimators can be applied [3, 19]. Two main methods are possible to get *so**: 1) **Manual:** some social networks provide users with the possibility to tag salient objects in photos (by drawing rectangles on the related regions and filling the names of corresponding contacts/people), 2) **Automatic:** using face detection algorithms to detect faces in the photos.

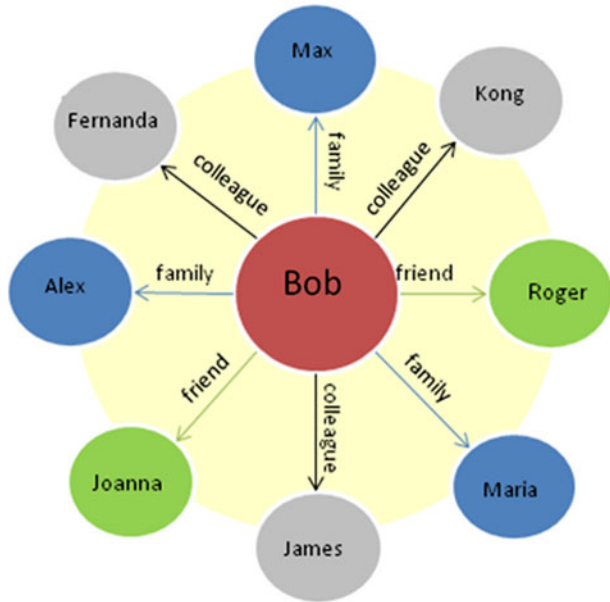
In this work, we assume that *so** are already identified and tagged by users (method 1) since the collected photos are from a social network site which provides such functionality. Formally, a salient object is defined as:

so: (*uid*, *pid*, *Date/Time*, *location*, *tag*)

⁵ <http://www.exif.org>

⁶ DC is an XML/RDF based syntax used to describe textual as well as audiovisual documents (see <http://dublincore.org/>).

Fig. 2 A sample star social network with three relationship types



where:

- *uid* is the identifier of the person who created the given *so*. It can be the owner of the photo or one of her contacts.
- *pid* is the identifier of the related photo
- *Date/Time* is the date/time value when *so* was created
- *Location* is the region containing the salient object in the photo. The salient object is not assigned a fixed width and height. Its coordinates determine the location and the size of

Fig. 3 FOAF attributes

FOAF Core	Social Web
<ul style="list-style-type: none"> • Agent • Person • name • title • img • depiction (depicts) • familyName • givenName • knows • based_near • age • made (maker) • primaryTopic (primaryTopicOf) • Project • Organization • Group • member • Document • Image 	<ul style="list-style-type: none"> • nick • mbox • homepage • weblog • openid • jabberID • mbox_sha1sum • interest • topic_interest • topic (page) • workplaceHomepage • workInfoHomepage • schoolHomepage • publications • currentProject • pastProject • account • OnlineAccount • accountName • accountServiceHomepage • PersonalProfileDocument • tipjar • sha1 • thumbnail • logo

the salient object region. We represent a salient object as a region in the photo using the Image vocabulary⁷

- $tag \in \{text, photo, video, audio\}$ is the tag assigned to this salient object. The tag can be in the form of a text, or another multimedia object used to tag this *so*. In this work, the tag is only limited to a text that contains either the name of a person (and assumed to be in the social network) or an adjective (cousin, wife, etc.).

Its creator, date/time of creation, and description are represented using the DC standard. □

Definition 5 Photo Album: It is a set of photos grouped together. The title and the description of an album are given by the creator of the photo album. A user, u_x , may create several photo albums according to several criteria. Other users may share with u_x their photo albums. An album is represented as follows:

PhotoAlbum: (*uid, aid, title, desc, photo**)

where:

- *uid* is the identifier of the creator of the album
- *aid* represents the photo album Universal Resource Identifier (URI)
- *title* is the title of the album
- *desc* is the description assigned to the whole album
- *photo** is the set of related photos.

Its creator, title and description are represented using the DC standard. □

Definition 6 User Preferences: enable the user to tune the settings of attributes by assigning them some values. It is represented as a set of attributes: ($pref_1:s_1, \dots, pref_n:s_n$) where $pref_i \in \{relationship\ label, work\ time, workdays, work\ place\ name, work\ place\ GPS\ location, work\ keywords, holidays\ dates, home\ place\ name, home\ place\ GPS\ location, family\ keywords, friend\ place\ name, friend\ place\ GPS\ location, friend\ keywords, age\ range\ categories\}$, and each s_i is the value associated to the related preference attributes. Users can choose to fill manually the values of any attribute or to leave the system trying to obtain these values from the users' profiles. Some of the attributes are related to each relationship type (e.g., *work time* is related to the Colleague type, *home place* is related to the Family type, etc.). □

To illustrate the proposed data model used in this study, let us consider the shared photo of Alice in Fig. 4. Her FOAF social profile is described in Fig. 5. For the sake of space reduction, we only display a small set of information available in her profile, one album photo, and one of her contact list.

3.2 Framework

Our framework is composed of 5 main modules: 1) **Profile Parser**, 2) **Photo Parser**, 3) **User Preferences**, 4) **Rule Analyzer**, and 5) **Relationship-type Discoverer**. The star social network of the main user is the input of the whole process. The framework is detailed in Fig. 6 and the rule analyzer is shown in Fig. 7.

1. **Profile parser:** extracts the profile of the main user as well as the profiles of all available contacts in the star social network (e.g., *age, gender, working place name, school name*, etc.). The values of all the selected attributes are stored in the profiles database component.

⁷ http://www.bnowack.de/w3photo/pages/image_vocabs

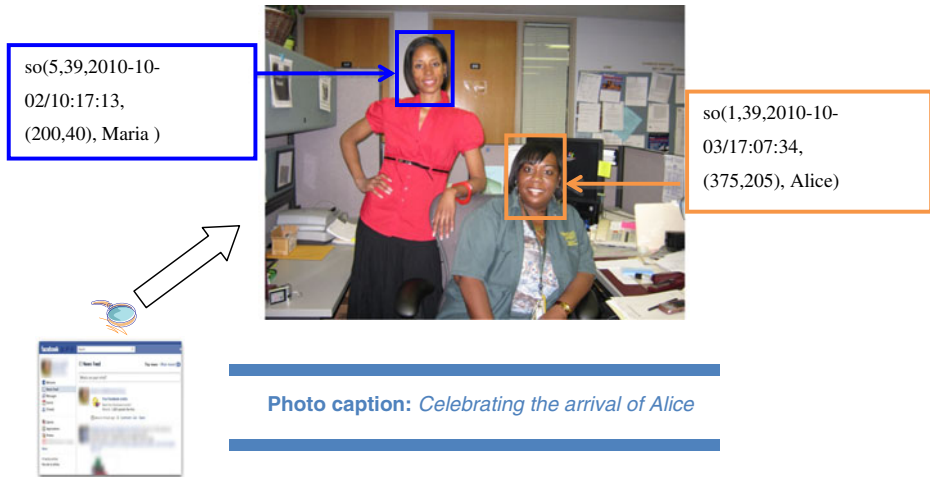


Fig. 4 Sample photo (with its salient objects) shared on the social network of Alice

2. **Photo parser:** retrieves, analyzes, and stores the photos of the user's profile (e.g., *img* and *image*) as well as all the photos shared by her contacts in order to extract available *metadata*, *captions*, *descriptions*, *tags*, and other information defined in our data model.
3. **User Preferences:** stores user's preferences (e.g., labeling relationship types to be discovered, assigning a list of keyword to each category [8], choosing valid predefined basic rules, giving work place and home place details, etc.) and feedback. Users can define sub-categories within the default relationship types. For instance, users can choose to differentiate between different types of colleague such as "former colleague" and "current colleagues".
4. **Rule analyzer:** tests, via its *rule validator* component, the validity of the set of selected predefined basic rules⁸ over the stored user profiles and photos in order to come up with "valid basic rules" (that match with at least one photo) and "invalid basic rules". Thus, persons depicted in photos corresponding to a valid rule are assigned the relationship type indicated in the rule. Of course, a contact can be assigned to more than one category (e.g., a person, who appears in two photos having different relationship types, is assigned to the two relationship categories). "Valid basic rules" and "invalid basic rules" are relaxed and derived with respect to the main user's social network content and preferences by the *rule miner* component of the rule analyzer. This component applies mining techniques on input rules and consequently generates a set of derived (or relaxed) valid rules and invalid rules to be applied later on the remaining set of photos and profiles. In this work, we chose to use the association rule mining methods (See Appendix A for a short description about them). Other methods could have been used such as: Decision Tree [27], Bayesian classifiers [26], SVM [33], etc. However, we chose to adopt association rule mining methods because it is found to be more accurate than some traditional classification methods, such as C4.5 [13]. These

⁸ The basic rules, which are based on common sense, are considered as being the more trustworthy. They will be detailed later on.

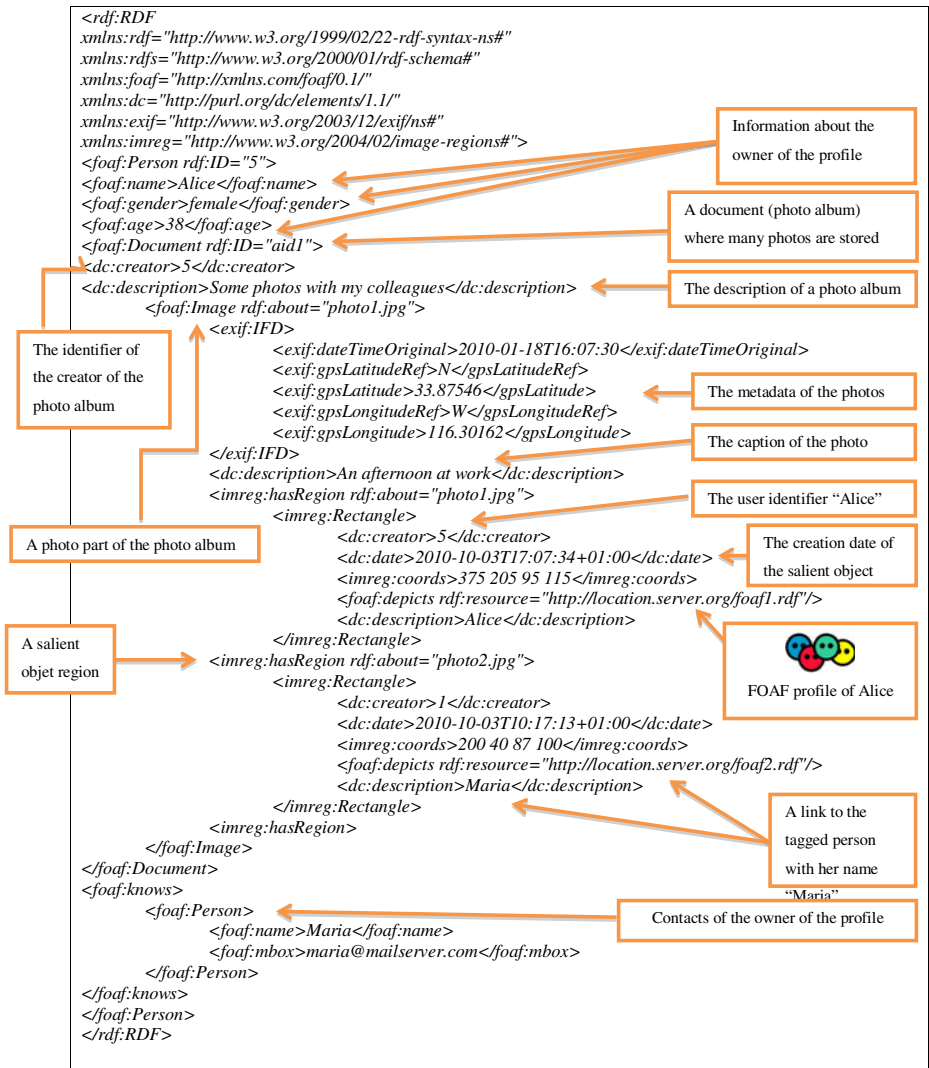


Fig. 5 Representation of Alice FOAF profile

methods yield new rules that explore highly confident associations among multiple attributes. Thus, they overcome some constraints encountered by decision-tree induction approaches, which consider only one attribute at a time [38].

5. **Relationship-type Discoverer:** takes as input the “derived valid rules” and the remaining⁹ set of photos and profiles of corresponding users in order to assign them matched relationship types. If multiple derived valid rules match a same photo, the rule with the highest confidence score is used. After, if other photos remain un-typed, the “derived invalid rules” are applied over them. This does not mean, however, that the

⁹ The photos that remain without any matching with a basic rule.

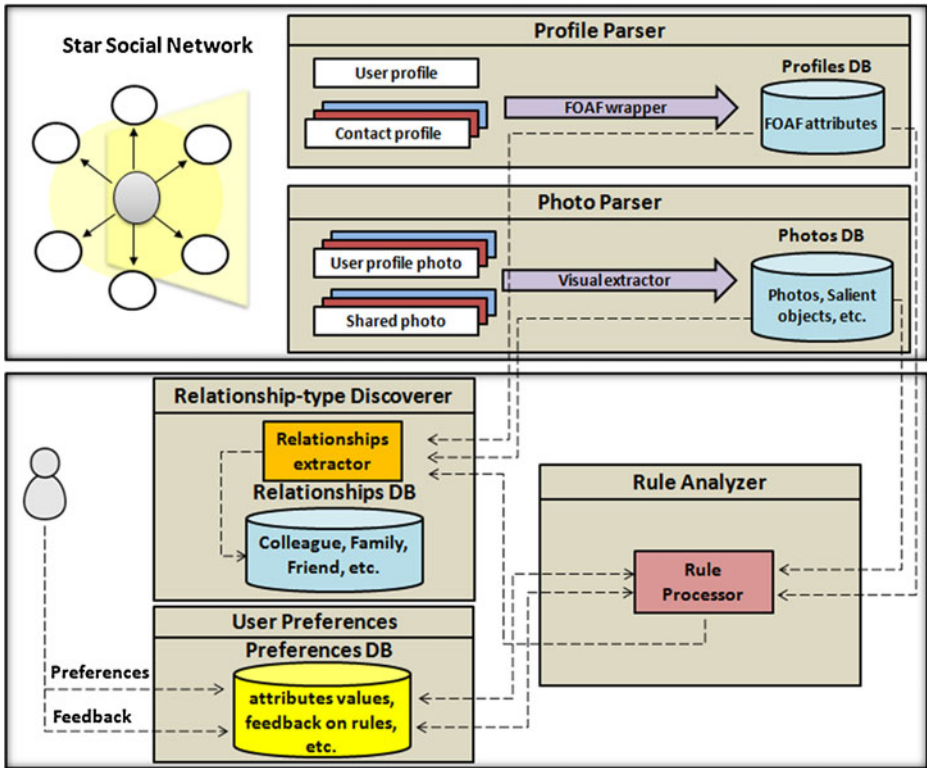
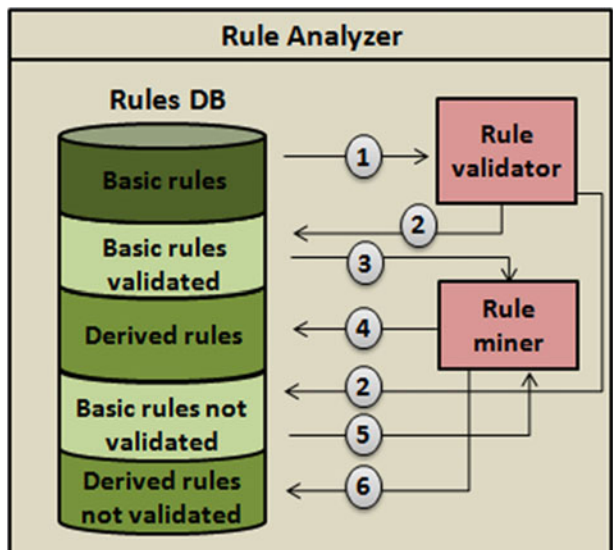


Fig. 6 Architecture of our framework for relationship type discovery

Fig. 7 Rule analyzer



missing relationship types are predefined and/or belong to the remaining types. In this case, the system would ask the user to provide new relationship type label(s).

The pseudo-code of our algorithm of relationship type discovery is provided here. The functions used in this algorithm are listed in Appendix B.

Algorithm: Relationship-type discovery	
Input:	
u_0	// The profile of the main user
$Pref(u_0)$	// The preferences of the main user (e.g., work place, confidence, etc.)
$Pref(u_0).relType[]$	// The labels of the relationship type
$Pref(u_0).keywordsRelType[]$	// The list of keywords associated to each relationship type
$Pref(u_0).placeRelType[]$	// The data related to the place name and to the place
	// GPS location associated to each relationship type
$Pref(u_0).ageRange[]$	// The ranges and differences of age of each relationship type.
	// By default, 3 categories are provided: baby, kids, and adult
$Pref(u_0).basicRules[]$	// The set of basic rules selected by the user
Output:	
$u_0.relType$	// The typed star social network of the user u_0
BEGIN	
$photoAlbum[] = RetrieveMO(G(u_0))$	// Gets all shared photos of the user star network.
	// Using the set of basic rules to assign relationship types to contacts
	// The validity of each rule is tested over the stored set of profiles and photos
FOREACH $bRule, IN Pref(u_0).basicRules[]$	
FOREACH $album, IN photoAlbum$	
FOREACH $photo_k, IN album,$	
IF $(CheckRuleValidity(bRule,)) = True)$	// Test the validity of the rule
$ValidBasicRule[] \leftarrow bRule,$	// Stores the basic rule in the Rules DB as valid
$Contacts[] \leftarrow GetContacts(photo_k)$	// Retrieves the names of the persons in the photo
$u_0.relType(Contacts) \leftarrow bRule.Type$	// Assigns the type indicated in the valid rule to the
	// relationship between u_0 and the persons in the photo
ELSE	
$InvalidBasicRule[] \leftarrow bRule,$	// Stores the rule in the Rules DB as invalid
$RemainingPhotos[] \leftarrow photo_k$	// Stores the set of photos that didn't match any basic rule
ENDIF	
ENDFOR	
ENDFOR	
ENDFOR	
	// Applies mining technique on the set of valid basic rules to generate a set of derived (or relaxed) rules
$derivedRules[] \leftarrow MineRules(ValidBasicRule[])$	
	//Using the set of derived valid basic rules to assign relationship types to contacts over the remaining collection of photos
FOREACH $dRule, IN derivedRules[]$	
FOREACH $album, IN photoAlbum$	
FOREACH $photo_k, IN album,$	
IF $photo_k, IN RemainingPhotos[]$	
IF $(CheckDerivedRules(dRules,)) = True)$	
$Contacts[] \leftarrow GetContacts(photo_k)$	
$u_0.relType(Contacts) \leftarrow dRule.Type$	
ENDIF	
ELSE	
$RemainingPhotosAfterMining[] \leftarrow photo_k$	// Stores the set of photos that didn't match any derived valid rule
ENDIF	
ENDFOR	
ENDFOR	
ENDFOR	
	// Applies mining technique on the set of invalid basic rules to generate a set of derived (or relaxed) invalid rules
$derivedInvalidRules[] \leftarrow MineRules(InvalidBasicRule[])$	
	// Using the set of derived invalid basic rules to propose to the user a grouping of contacts to which she can assign a relationship type
FOREACH $dInvalidRule, IN derivedInvalidRules[]$	
FOREACH $album, IN photoAlbum$	
FOREACH $photo_k, IN album,$	
IF $photo_k, IN RemainingPhotosAfterMining[]$	
IF $(CheckDerivedRules(dInvalidRules,)) = True)$	
$Contacts[] \leftarrow GetContacts(photo_k)$	
$u_0.RelType(Contacts) \leftarrow USER.Type$	//The user is asked to assign manually the relationship type
ENDIF	
ENDIF	
ENDFOR	
ENDFOR	
ENDFOR	
Return $relType(u_0)$	
END	

3.3 Default basic rules

Here, we present the basic set of rules that we use to discover the three following types of relationships that link a user to her contacts: *Colleague*, *Family* and *Friend* relationship. In order to check the validity of a rule, the algorithm extracts different kinds of information from photos and profiles:

- 1) the metadata of the photo (day, time, and description),
- 2) the photo cardinality (photoCard),
- 3) the maximum difference of age between persons depicted in the photo (ageRange),
- 4) the gender of the persons depicted in the photo (gender),
- 5) the place depicted in the photo (its description and its GPS location).

The following algorithm shows how to discover the *colleague* relationship type.

Algorithm: Colleague relationship type default basic rules	
1.	Colleague(u,c) IS TRUE WHEN // Detecting colleagues based on the work place description and on the work place location
2.	((GetDate(photo) IN {GetWorkDays(u), GetNotWorkDays(u)}) && (GetTime(photo) IN {GetWorkTime(u), GetNotWorkTime(u)})) && ((GetDescription(photo) IN { Pref(u), keywordsRelType[Colleague], Pref(u), placeRelType[Colleague]} (distance(GetPlaceGPS(photo), GetPlaceGPS (u, Colleague)) = 0)) // Detecting adult colleagues together on a working day at a working time
3.	(GetCard(photo) >= 2) && (CheckAgeRange(photo, Contacts) IN {smallDiff, bigDiff}) && (GetDate(photo) IN {GetWorkDays(u)} && (GetTime(photo) IN {GetWorkTime(u)} && (CheckPlace(photo) = UnknownPlace) // Detecting adult colleagues of the same gender
4.	(GetCard(photo) = 2) && (GetGender(c ₀) = GetGender(c ₁)) && (CoAppear(c ₀ , c ₁) <= 3) && (GetDate(photo) IN {GetWorkDays(u)} && (GetTime(photo) IN {GetWorkTime(u)} && (CheckPlace(photo) = UnknownPlace)

The rules state that persons depicted in a photo are assigned to the relationship type *colleague* in the following cases:

1. When a photo description matches with the place name description of the *colleague* relationship type (from user preferences) or with the set of keywords assigned to the *colleague* relationship type (from user preferences). This is true regardless of the daytime and regardless whether it is a working day or not (line 2).
2. When the distance between the GPS location of the *colleague* relationship type (from user preferences) and the GPS location of the photo is equal to 0. This is true regardless of the daytime and regardless whether it is a working day or not (line 2).
3. When a group of persons (e.g., the number is greater than 2 persons) is depicted in a photo (line 3). The age difference between all these persons can be small (less than 20 years) or big (greater than 20 years). No babies, no kids are depicted in the photos. The age difference value, which is 20 years by default, is a threshold that can be modified in the user preferences. The day must be a working day, and the time must be a working time. The place depicted in the photo is unknown (an unknown place is a place having a description different from all the descriptions and from all the keywords assigned to relationship type, and a GPS location different from all the GPS location in the user preferences).
4. When two persons of the same gender are depicted alone in some photos (line 4). Here, the number of photo in which they co-appear alone is set to be less than 3 photos (default threshold in the user preferences). The day must be a working day, and the time must be a working time. The place depicted in the photo is unknown.

The *family* relationship type discovery algorithm is provided in the following pseudo-code. The rules state that persons depicted in a photo are assigned to the relationship type *family* in the following cases:

Algorithm: Family relationship type default basic rules	
1.	Family(u,c) IS TRUE WHEN // Detecting family members based on the home place description and the home place location
2.	((GetDate(photo) IN {GetWorkDays(u), GetNotWorkDays(u)}) && (GetTime(photo) IN {GetWorkTime(u), GetNotWorkTime(u)})) && ((GetDescription(photo) IN {Pref(u),keywordsRelType[Family], Pref(u),placeRelType[Family]}) (distance(GetPlaceGPS(photo), GetPlaceGPS (u, Family)) == 0))
3.	// Detecting parents with their kids (GetCard(photo) == 2) && (CheckAgeRange(photo, Contacts) IN {kidsAdultDiff}) && (GetDate(photo) IN {GetWorkDays(u), GetNotWorkDays(u)}) && (GetTime(photo) IN {GetWorkTime(u), GetNotWorkTime(u)}) && (CheckPlace(photo) == UnknownPlace)
4.	// Detecting spouses of different genders (GetCard(photo) == 2) && (CheckAgeRange(photo, Contacts) IN {smallDiff,bigDiff}) && (GetGender(c ₀) != GetGender(c ₁)) && (CoAppear(c ₀ ,c ₁) >=3) && (GetDate(photo) IN {GetWorkDays(u), GetNotWorkDays(u)}) && (GetTime(photo) IN {GetWorkTime(u), GetNotWorkTime(u)}) && (CheckPlace(photo) == UnknownPlace)
5.	// Detecting adult family members with a big age difference e.g. grandparents with their adult kids (GetCard(photo) >= 3) && (CheckAgeRange(photo, Contacts) IN {bigDiff}) && (CheckPlace(photo) == UnknownPlace)

1. When a photo description matches with the place name description of the *family* relationship type (in the user preferences) or with the set of keywords assigned to the *family* relationship type. This is true regardless of the daytime and regardless whether it is a working day or not (line 2).
2. When the distance between the GPS location of the *family* relationship type (user preferences) and the GPS location of the photo is equal to 0. This is true regardless of the daytime and regardless whether it is a working day or not (line 2).
3. When two persons, one of them is an adult and the other is a baby or a kid, are depicted alone in a photo. The place depicted in the photo is unknown. In this case, the relationship type is *family*; regardless of the daytime and regardless whether it is a working day or not (line 3).
4. When two persons of different gender are depicted alone in some photos (line 4). Here, the number of photo in which they co-appear alone is set to be greater than 3 photos. The age difference between all these persons can be small or big. Neither babies nor kids are depicted in the photos. The place depicted in the photo is unknown. The relationship type is *family*; regardless of the daytime and regardless whether it is a working day or not (line 4).
5. When a group of person is depicted in the photo. The age difference is big and the place depicted in the photo is unknown (line 5).

The pseudo-code of the *friend* relationship type discovery is below.

Algorithm: Friend relationship type default basic rules	
1.	Friend(u,c) IS TRUE WHEN // Detecting friends based on the friend place description and on friend place location
2.	((GetDate(photo) IN {GetWorkDays(u), GetNotWorkDays(u)}) && (GetTime(photo) IN {GetWorkTime(u), GetNotWorkTime(u)})) && ((GetDescription(photo) IN { Pref(u),keywordsRelType[Friend], Pref(u),placeRelType[Friend]}) (distance(GetPlaceGPS(photo), GetPlaceGPS (u, Friend)) == 0)) // Detecting adult friends of different genders
3.	(GetCard(photo) == 2) && (CheckAgeRange(photo, Contacts) IN {smallDiff,bigDiff}) && (GetGender(c ₀) != GetGender(c ₁)) && (CoAppear(c ₀ ,c ₁) <=3) && (GetDate(photo) IN {GetWorkDays(u), GetNotWorkDays(u)}) && (GetTime(photo) IN {GetWorkTime(u), GetNotWorkTime(u)}) && (GetPlace(photo) == UnknownPlace)
4.	// Detecting friends having small age difference (GetCard(photo) <= 3) && (CheckAgeRange(photo, Contacts) IN {smallDiff}) && (CheckPlace(photo) == UnknownPlace)

The rules state that persons depicted in a photo are assigned to the relationship type *friend* in the following cases:

1. When a photo description matches with the place name description of the *friend* relationship type or with the set of keywords assigned to the *friend* relationship type.

- This is true regardless of the daytime and regardless whether it is a working day or not (line 2).
2. When the distance between the GPS location of the *friend* relationship type and the GPS location of the photo is equal to 0. This is true regardless of the daytime and regardless whether it is a working day or not (line 2).
 3. When two persons of different gender are depicted alone in some photos (line 4). Here, the number of photo in which they co-appear alone is set to be less than 3 photos. The age difference between all these persons can be small or big. Neither babies nor kids are depicted in the photos. The place depicted in the photo is unknown. The relationship type is *friend*, regardless of the daytime and regardless whether it is a working day or not (line 3).
 4. When a group of person is depicted in the photo. The age difference is small and the place depicted in the photo is unknown (line 4).

4 Prototype and experiments

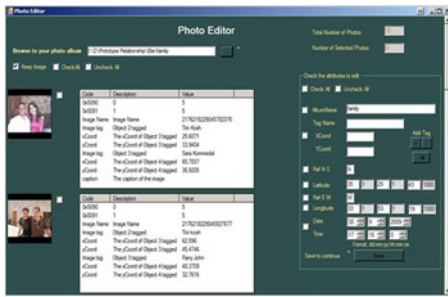
In this section, we present the prototype and the experiments that we conducted to assess the efficiency and the relevancy of our approach. First, we detail the prototype that we designed before describing the dataset used in the tests and the results obtained for the different parameters calculated in order to quantify the contribution of the proposed method.

4.1 Prototype

Implemented using C# programming language, our prototype encompasses 5 main components:

1. **Profile and photo retriever:** is used to retrieve the set of (contacts) profiles and shared photos of a given user. Photos are retrieved with their metadata, captions, descriptions, and salient objects.
2. **Photo editor:** provides a user with the possibility to edit the photos. She may modify metadata attribute values, add a tag, give a description, etc.
3. **Preference manager:** allows a user to define her own settings. She can set the relationship types that she wishes to discover. For each relationship type, she has to fill the label, the keywords, and the values of the related attributes.
4. **Relationship finder:** is the main module. It allows to assign a relationship type to the persons that appear in users' photos. This component uses a set of basic rules whose attribute values are taken from the preference manager component. All photos are analyzed by these rules.
5. **Rule Miner:** extracts new rules derived from the basic set of rules. In this component, we implement the **Apriori** rule mining algorithm [2]. This algorithm generates frequent itemsets for a given dataset and then scans them to select the most frequent items in the dataset. In a given dataset, association rule mining finds all rules that satisfy the minimum support and the minimum confidence.

The user profile, his preferences, and the output are stored as XML files. In Figs. 8 and 9, we show some snapshots of the user interfaces of this prototype below.



Snapshot of the photo editor interface



Snapshot of the user preferences module

Fig. 8 Input interfaces

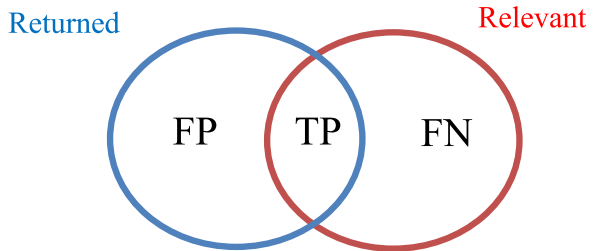
4.2 Experiments

To validate our approach, we conducted a set of tests using a collection of real personal photos that belongs to 10 users on Facebook.¹⁰ We used a dataset consisting of 1000 photos equally distributed between 10 users. In our study, we limited the discovery process to only the three default relationship types: *Colleague*, *Family*, and *Friend*. We assume that each user knows the relationship type to each of her contacts list. Each user had 60 contacts where 25% are colleagues, 40% are family members, and 35% are friends. In our experiments, 22% of the photos contained one person, 50% contained two persons, and 28% contained more than two persons. Note that in this study, we assume that the photos are already tagged. The string comparison was done with exact matching. To measure the relevance of our approach in detecting and discovering correct relationship types, we used three popular information retrieval measures: Precision (P), Recall (R), and F-score (F) calculated as follows:

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$

$$F = 2 \times \frac{P \times R}{P + R}$$



where *TP*, *FP*, and *FN* denote respectively True Positive (number of returned correct relationships), False Positive (number of returned incorrect relationships), and False Negative (number of not discovered correct relationships).

The aim of the set of experiments conducted in this study was to evaluate: 1) the relevance of our proposed method before and after the rule mining, 2) the correctness and the scope of our set of basic rules, 3) the relevance of our proposed approach while varying the percentage of annotated photos, 4) the impact of varying the confidence and the support values on the results after the rule mining, 5) the execution time, and 6) the capacity to potentially discover undefined relationships.

In the following experiments (except Test 3), the dataset of photos was annotated of 50% (e.g., some photo attributes were assigned empty values). Experiments were

¹⁰ <http://www.facebook.com>

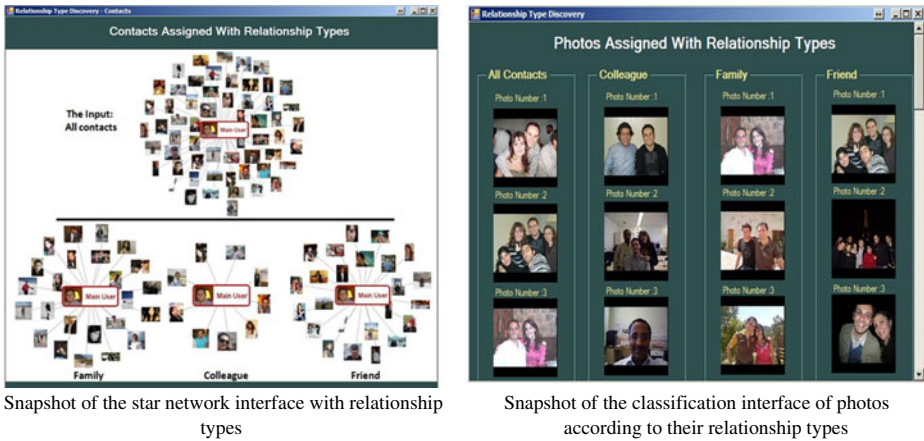


Fig. 9 Output interfaces

carried out on an Intel Core Duo processor machine with 2.50 GHz processing speed and 2GB of RAM.

4.2.1 Test 1: relevance of our proposed approach before and after rule mining

The first test aimed at showing that our approach was able to discover the correct relationship type between a user and her contacts, with and without using the rule mining algorithm. For the mining approach, we varied the value of the minimum confidence score. As a result, only rules with a higher confidence value than this minimum score were selected; the ones that have a smaller value were eliminated from the new set of rules.

The obtained results are shown in Table 2. One can see that the score of the precision varied in different manners with and without mining based on the relationship type, while for the recall the score after mining was greater than or equal to its score without mining. Indeed, the interpretation of these results must jointly consider the two results. Here, we emphasize on these results in 3 main points:

- 1) When having a minimum confidence score of 0%: The recall value without mining was less than the recall value for a minimum confidence score of 0%. This means that the system was able to discover a bigger number of correct relationship types. In other words, a minimum confidence of 0% means that all the mined rules obtained from the Apriori algorithm were used. This explains why the system with mining was able to discover 34% more correct relationship types for the *Colleague* category, 24% more for

Table 2 Precision and Recall values with different confidence scores

	Colleague		Family		Friend	
	P	R	P	R	P	R
Without mining	53%	53%	52%	52%	53%	53%
With mining: confidence 0%	53%	87%	55%	76%	67%	67%
With mining: confidence 50%	60%	87%	42%	57%	69%	73%
With mining: confidence 100%	53%	53%	52%	52%	53%	53%

- the *Family* category, and 14% more for the *Friend* category. However, since all the obtained rules were applied, this means that the system might also have returned false positive results. For instance, as shown in Table 2, the precision of the *Colleague* category after mining was the same with and without mining. In fact, the number of true positive increased as well as the number of all returned results with mining. This explains why the precision measure didn't increase. However, the number of true positive relevant relationship types has increased since the recall value increased by 30%. As for the precision value of the Family category, the precision after mining was 3% better, like the precision for the Family category, which also was better after mining with 14% higher precision than before mining
- 2) When having a minimum confidence score of 50%: only rules with higher confidence value were taken into consideration. This filtered out the initial set of rules and gave the following result: the precision value, for the *Colleague* and *Friend* categories, increased respectively of 7% for the first and 2% for the second category. For the *Family* category, the precision dropped down. Although rules with confidence score less than 50% were capable to identify a number of *Family* relationships, they were eliminated (for a minimum confidence score of 50%). So, a number of *Family* relationship types previously identified with 0% confidence score was not identified with 50% confidence score. As a proof to this interpretation, we can see that the recall score dropped down for the Family category with a minimum confidence of 50%. This means that some true positive results were not identified. In the cases of *Colleague* and the *Friend* categories, the increase of the recall percentage was due to a decrease in the number of the returned false positive results
 - 3) When having a minimum confidence of 100%: only rules, which had a confidence of 100%, were selected. This was the uppermost value that could be applied and the obtained sets of mined rules were very similar to the basic set of rules. The relaxed rules in the Apriori algorithm were filtered by this confidence value. As a result, similar precision and recall values were obtained with and without mining.

Obviously, these results confirm the capacity of our approach to discover relationships between persons in the context of social networks. In addition, it shows that obtained results when applying derived rules allowed to improve the results.

4.2.2 Test 2: a comparative study of the used rules

In this test, we compared our set of basic rules presented in Section 3.3 with the set of rules used in the work of Singla et al. [30] where the rules aimed to discover two big categories of relationship types: family (relative, parents) and friends (acquaintances). However, our basic set of rules considers three relationship types: family, friend, and colleague. Consequently, in order to be able to compare the two sets of rules we filtered our basic set of rules so they only refer to family and friends relationship types.

We used the same dataset for both tests as the one described in this experiment section. We conducted this test using only basic rules without mining. The following table summarizes the F-scores results obtained in this test.

Table 3 F-score results when comparing the basic rules

	Family	Friend
Singla et al. [30]	58,46%	60%
Our approach	73,87%	75,5%

As observed in the Table 3, for both relationship type (Family and Friend) our approach achieved a higher F-score value than the one of the other approach. Indeed, while the rules in [30] are limited to a subset of attributes (age, gender, co-appearance), our rules consider a wider set of attributes extracted from photos such as date, time, location, etc.

4.2.3 Test 3: relevance of our proposed approach while varying the percentage of annotations/descriptions in photos

In this test, we varied the percentage of annotations/descriptors in photos by assigning empty values to one or several photo attributes. The aim of this test was to study the effect of available annotations/descriptors of the photos on discovering the relationship types using our approach. To do that, we conducted 4 experiments with different photo annotation percentages varying them from 25% to 100%. The confidence value varied from 0% to 100% in each of these experiments. The obtained results in terms of F-score on *Colleague*, *Family*, and *Friend* categories are provided in Fig. 10. One can easily observe that the more the dataset is annotated, the more the result is accurate. This observation was true for all the categories and for all the confidence values. Another important observation to note is related to the tendency of F-score value which drops down when reaching 90% of confidence value. This is due to the limited number of rules with a confidence value between 90% and 100%. In other terms, since the set of used rules was restricted to the basic rules, the obtained F-score value was similar to the one without mining. This

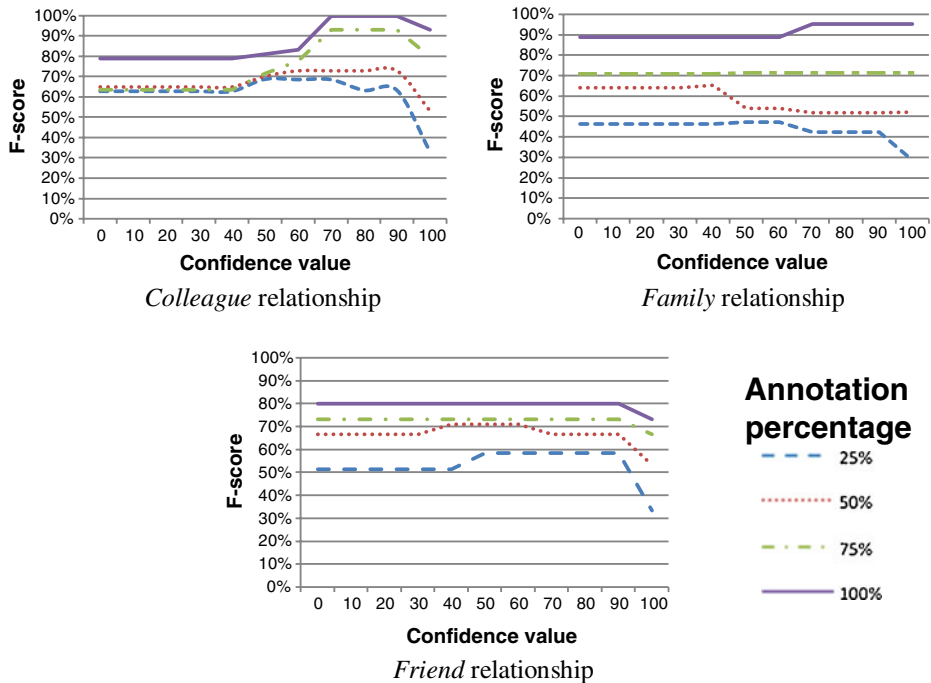


Fig. 10 Comparing the results of F-score while varying the percentage of photos annotations for each relationship type

experiment showed also that for a 100% set of annotated photos, a high F-score value was obtained:

- for *Colleague* relationship: between 80% and 100%,
- for *Family* relationship: between 90% and 100%,
- for *Friend* relationship: between 75% and 80%.

For a percentage of 25% of annotated photos (this was the lowest percentage), while varying the confidence value between 0% and 90%, the F-score value achieved a score between 61% and 69% for *Colleague* relationship, between 41% and 49% for *Family* relationship, and between 51% and 59% for *Friend* relationship. This means that accurate and efficient results can be obtained even with partly annotated sets of photos are in hand.

4.2.4 Test 4: impact of varying the confidence value and the support value on the results after rule mining

Through this test, we investigated the accuracy of our approach while varying support and confidence scores of the mined rules. The F-score values for all of the three relationship types were computed. Results of the three categories showed the same pattern. The support score was varied from 0% to 100%. Support scores with a percentage greater than 35% gave the same results, and consequently we don't show them on the result obtained in Fig. 11.

Based on the computed results shown in Fig. 11, we were able to pin down two main observations:

- When the confidence value is equal to 100%: For different support values, regardless of the F-score value, all reported the same F-score value. In such cases, the generated rules were filtered out and eliminated from the new set of rules. Consequently, the set of rules was reduced to the set of basic rules available before applying the mining algorithm
- The lower the support value, the higher the value of the F-score: Rules with low support value use a wider set of rules to discover relationships, this is why they achieved greater F-score values than rules with higher support values.

4.2.5 Test 5: execution time

We analyzed the execution time of our proposed approach with different parameters. We measured the execution time before and after applying the rule mining. It was obvious that without applying the rule mining, the overall time to scan and process the collection of photos was higher than the time needed to process the remaining photos after mining. As mentioned previously, the rules obtained after mining were only applied to the remaining

Fig. 11 Comparing the F-score values while varying the support and the confidence values of the mined rules

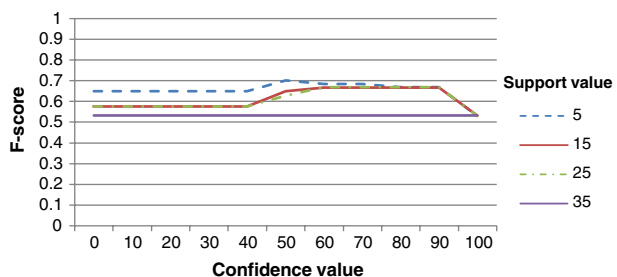
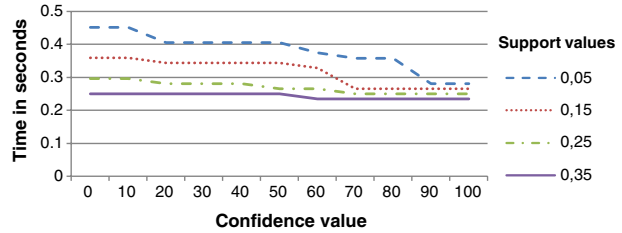


Fig. 12 Measuring the execution time while varying the support and the confidence values of the mined rules



photos that the basic set of rules haven't discovered previously. In Fig. 12, we show the time execution after mining with different support and confidence values.

We varied the confidence value that is represented on the x-axis. We considered different support values ranging from 5% to 35%. Above 35%, the reported results were the same. The measured time is drawn as a line for each support value and for each confidence value. We can obviously see that the execution time was decreasing when the confidence value was increasing. In fact, the number of used rules decreased since rules having a confidence value less than the minimum one were eliminated. The same reasoning was true for the different support values. In both cases, fewer applied rules required less execution time.

In Fig. 13, we show the effect of varying the number of photos in a collection. We used different album photo collection ranging from 100 to 500 photos per album and per user. Different measures are shown for each collection evaluated based on different confidence values. As observed on the graph, the execution time needed to process all the photos increased rapidly but this is acceptable when having reasonable social network size and/or number of shared photos.

4.2.6 Test 6: a recommendatory system using the non-used hard rules

The aim of this experiment was to show that our system was able to detect the existence of undefined relationship types and allows the user to label them. As shown in Fig. 14, when the confidence values range between 0% and 40%, the basic set of rules and their extended rules were enough to discover all the relationship types. However, when a confidence score is greater than 40%, they did not succeed in identifying the relationship type(s) for the set of remaining photos. Accordingly, the proposed approach was able to suggest the possibility of having a set of photos that share common patterns. In fact, as we notice in Fig. 14, the system started to return its recommendation from a minimum confidence of 40%. This is due to the number of extended rules eliminated because their confidence value was lower than the required one. Therefore, contacts depicted in these photos were left without a defined relationship type. This was the trend for values whose confidence value ranged between 40% and 90%. This series reached a peak for the confidence value of 70% and then it decreased. For a confidence equals to 100%, there was no returned results by the system because no rule had a confidence 100%.

Fig. 13 Measuring the execution time while varying the number of photos

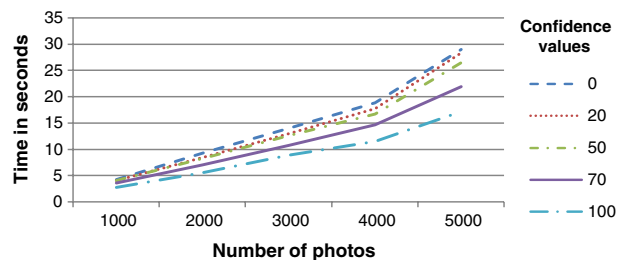
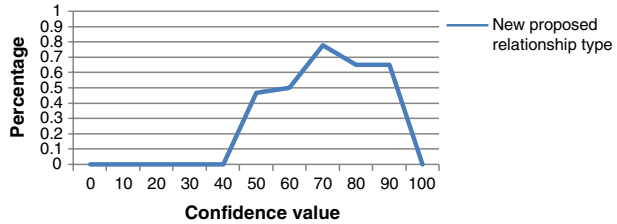


Fig. 14 Percentage of persons having an undefined type of relationship



In this test, we showed how the system is able to detect undefined relationships from a set of photos that share common characteristics. The user can assign any relationship type of her choice to persons depicted within the set of photos. The ability to discover new persons appearing in photos was based on common characteristics derived from relaxing/deriving the unused basic rules.

5 Conclusion

In this work, we have addressed the issue of enriching relationship types between users in social network sites. The main objective is to protect the privacy of a user by identifying the type of the relationship between her and her contacts. Our approach extracts information from users' profiles, in particular photos and their attributes, in order to identify the types of relationships. A set of basic rules based on observations and common sense is used. These rules are extended in a way to take into consideration the context of each user. We have developed a prototype to verify the potential and the capability of our approach. We have elaborated a set of experiments and tested them on real data. The results were satisfactory and encouraging.

When processing real world data, the issue of information completeness arises. There are, of course, social network profiles with missing information and photos that have not been previously tagged. This would prevent the system to discover the relationship type with people who appear in those photos. Therefore, we must consider not only previously tagged photos by users, as it is done in this current work, but also photos that have not been tagged. We are currently working on this issue and we intend to apply image-processing techniques, such as face detection and face recognition algorithms, to improve the obtained results. In addition, users may assign persons with tags that refer to adjectives, words, nicknames rather than their real names contacts' names. To solve this issue, we plan to include efficient algorithms and some semantic similarity measures to allow tag disambiguation. We also further wish to fully integrate users' preference by letting users propose their own rules.

Acknowledgments We would like to thank Parag Singla, Henry Kautz, Jiebo Luo, Andrew Gallagher for providing us the set of rules they used in their work [30].

Appendix A—Outline on association rule mining

Association Rule Mining (ARM) is part of the descriptive classification category in the domain of data mining. It was introduced by [2] in order to identify relationships among a set of items in a database. To this end, it seeks to find frequent itemsets (attribute-value pair) then to derive association rules. An association rule has the form of: $X \Rightarrow Y$ where the left hand part is called the rule body and the right hand part is called the rule's head. Let us suppose a database D of k instances having n attributes $\{a_1, a_2, \dots, a_n\}$. A record, d , of the

database is called an itemset, and an itemset of k elements is called k -itemset. In an association rule: $X, Y \subseteq \{a_1, a_2, \dots, a_n\}$, $Y \neq \emptyset$ and $X \cap Y = \emptyset$. In our case, an itemset contains always the set of attributes in addition to the defined class. The rule's head or right hand part is an attribute that represents the class type while the left hand part is the set of attributes associated to this rule's head with a certain measure. The two classical and most fundamental rule interestingness measures used are:

- The *support* s : indicates the relative occurrence of X and Y within the overall dataset and is defined as the number of records satisfying both X and Y over the total number of records: $s(X) = |X \cup Y|/k$
- The *confidence* c : is the probability of Y given X measured by computing the ratio of the number of records satisfying both X and Y over the number of records satisfying X : $c(X \Rightarrow Y) = |X \cup Y|/X$

Appendix B—List of functions used in this study

Several functions are used in our approach to exploit information coming from user preferences, user profiles, shared photos, salient objects and albums. We list them in Tables 4 and 5.

A multimedia object, which is a photo in our context, can be described using its metadata, its description, and the persons depicted in the photo. The main attributes describing a photo that we use here are: *Date*, *Time*, and *Location*. To extract their values, we use the following functions respectively: *GetDate(photo)*, *GetTime(photo)*, and *GetPlaceGPS(photo)*. Another important facet for the image is the visual part that is composed of the faces of the persons who appear in the photo. We can count the number of persons that appear in a photo using the *GetCard(photo)* function and retrieve their names *GetContacts(photo)*. The main related functions are listed in Table 6. *GetName(so)* recognizes the name of a salient object in a photo. However, a user may tag the face of a person in the photo with a text different from real name, e.g. nickname (Alex instead of Alexander), with the social relationship type (my dad, my boss, etc.), or she can even mistype or misspell it (e.g. Alexandra instead of Aleksandra). So when a tag doesn't match any user's contact real name, similar candidate real names are extracted using the *CompareText(str1, str2)* function where *str1* is the available tag and *str2* represents the real name (or nicknames) of all the other user's contacts. Different similarity measures are used in this case depending on the user's choice. Syntactic similarity measures with a defined threshold are useful when used to compare attributes like person names [7, 15]. Therefore, taking into consideration the similarity score between is of great importance. For other attributes (place name, description, etc.) semantic similarity measures can be used [9]. For instance, having two photos one described with the name of a city (e.g., "Paris") and another one with the name of a

Table 4 Main user profile functions

Used function	Description
<i>GetProfile(uid)</i>	Extracts a profile of a given user
<i>GetContacts(uid)</i>	Extracts the list of contacts of a given user
<i>GetAge(uid)</i>	Gets the age of a person
<i>GetGender(uid)</i>	Gets the Gender of a person
<i>RetrievePhotos(uid)</i>	Gets all the main user photos and albums as well as those shared by her contacts

Table 5 Main user preferences functions

Used function	Description
<i>GetRelationshipTypes</i> (uid)	Gets the list of relationship types given by a user
<i>GetRelationshipKeywords</i> (uid, relType)	Retrieves the list of keywords associated by a user to a relationship type
<i>GetAgeRange</i> (uid)	Gets the age distribution as defined by the user: the minimum and the maximum age for considering a person as baby, kid, adult, or senior
<i>GetHolidays</i> (uid)	Gets the calendar of holidays (summer vacation, Christmas, etc.). This can be obtained manually by the user or by accepting the default holidays defined within a country, a region, etc.
<i>GetPlaceName</i> (uid, relType)	Gets the name of a place based on the relationship type (work place name, home place name, friend place name) provided manually by the user or automatically from the user profile
<i>GetPlaceGPS</i> (uid, relType)	Gets the GPS longitude and latitude of the place (home or work) related to a given relationship type
<i>GetWorkTime</i> (uid), <i>GetNotWorkTime</i> (uid)	Get the working hours and non-working hours respectively of a given user
<i>GetWorkDays</i> (uid), <i>GetNonWorkDays</i> (uid)	Gets the weekly working days and non-working days (weekends) respectively of a given user

famous place of this city (e.g., “Eiffel Tour”). These two descriptions are syntactically different, however semantically they are related since “Eiffel Tour” is in “Paris”.

Table 7 shows some other functions used in the discovery process. The *CheckRuleValidity (Rule)* tests the validity of a rule function. For a given photo collection that belongs to a user, each photo is processed by the rule validator component. When a photo matches with a rule, the rule related relationship type is consequently assigned to the persons that appear in the photo. In this case, the applied rule is considered as “valid”. It is possible that, one or more rules don’t match with any of the photo in the collection. Thus, those rules are considered as “invalid”.

Table 6 Main photo, salient object and albums functions

Used function	Description
<i>GetTime</i> (photo)	Gets the time of image creation
<i>GetDate</i> (photo)	Gets the date of image creation
<i>GetContacts</i> (photo)	Retrieves the names of the persons in the photo
<i>GetCard</i> (photo)	Returns the number of persons in a photo
<i>GetCaption</i> (photo)	Retrieves the keywords describing a photo
<i>GetDescription</i> (photoalbum)	Retrieves the keywords describing a photo album
<i>GetDescription</i> (photo)	Retrieves the keywords describing a photo and its corresponding photo album
<i>GetPlaceGPS</i> (photo)	Gets the GPS Longitude and Latitude from the photo
<i>CheckPlace</i> (photo)	Retrieves the description and the location of the place depicted in the photo. Then it checks if the extracted information corresponds to any information from the user preferences. If a correspondence exists the function returns its value, if not the place is indicated as <i>UnknownPlace</i>
<i>GetPersons</i> (photo)	Returns people names in a photo by recognizing their faces
<i>GetName</i> (so)	Returns the name of a salient object

Table 7 Other functions

Used function	Description
<i>CoAppear</i> (uid ₁ ,uid ₂)	Check how many times two persons appeared together in a photo
<i>CheckAgeRange</i> (uid ₁ ,uid ₂)	Computes the difference of age between two persons
<i>CheckAgeRange</i> (photo, Contacts)	Computes the difference of age between persons depicted in the photo and it returns the maximum difference
<i>CompareText</i> (str1,str2)	Computes the similarity score between two texts. Different syntactic and semantic similarity measures can be used.
<i>Distance</i> (point1, point2)	Computes the distance between two GPS points
<i>CheckRuleValidity</i> (Rule)	Tests the validity of a rule over the information extracted from a photo and from the depicted user's profile in the photo. Rules are stored as "valid basic rules" or "invalid basic rules". Persons depicted in the photos are assigned the type indicated in the rule if the rule is valid.

References

- Adamic L, Adar E (2003) Friends and neighbors on the web. *Social Networks* 25(3):211–230
- Agrawal R, Srikant R (1994) Fast algorithms for mining association rules in large databases. *Proceedings of the 20th International Conference on Very Large Data Bases*:487–499
- Baluja S, Rowley H (2007) Boosting sex identification performance. *Int J Comput Vis* 71(1):111–119
- Bilgic M, Namata G, Getoor L (2007) Combining collective classification and link prediction. *Proceedings of the Seventh IEEE International Conference on Data Mining Workshops*:381–386
- Bojars U, Heitmann B, Oren E (2007) A prototype to explore content and context on social community sites. *Proceedings of the International Conference on Social Semantic Web* 47–58
- Boyd D (2004) Friendster and publicly articulated social networking. *CHI '04 extended abstracts on Human factors in computing systems*: 1279–1282
- Cohen WW, Minkov E (2006) A graph-search framework for associating gene identifiers with documents. *BMC Bioinforma* 7(1):440
- Elliott B, Özsoyoglu Z (2008) Annotation suggestion and search for personal multimedia objects on the web. *Proceedings of the 2008 international conference on Content-based image and video retrieval*:75–84
- Gabrilovich E, Markovitch S (2007) Computing semantic relatedness using Wikipedia-based explicit semantic analysis. *Proceedings of the 20th international joint conference on Artificial intelligence*:1606–1611
- Golbeck J, Rothstein M (2008) Linking social networks on the web with foaf: A semantic web case study. *Proceedings of the 23 rd national conference on Artificial intelligence - Volume 2*:1138–1143
- Golder S (2008) Measuring social networks with digital photograph collections. *Proceedings of the nineteenth ACM conference on Hypertext and hypermedia*:43–48
- Green T, Quigley A (2010) *Perception of online social networks. Mining and analyzing social networks*, vol 288. Springer Publishing Company, Incorporated
- Han J, Kamber M (2006) *Data mining: concepts and techniques*. Morgan Kaufmann
- Hogg T, Wilkinson D, Szabo G, Brzozowski M (2008) Multiple relationship types in online communities and social networks. *Proceedings of the AAAI Symposium on Social Information Processing*:30–35
- Jaro M (1989) *Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida*. *J Am Stat Assoc* 84(406):414–420
- Jin Y, Matsuo Y, Ishizuka M (2007) Extracting social networks among various entities on the web, vol 4519. *LNCIS The Semantic Web: Research and Applications*. Springer Berlin/Heidelberg
- Kaplan AM, Haenlein M (2010) Users of the world, unite! The challenges and opportunities of social media. *Bus Horizons* 53(1):59–68
- Kautz H, Selman B, Shah M (1997) The hidden web. *AI Mag* 18(2):27–36
- Lanitis A, Taylor C, Cootes T (2002) Toward automatic simulation of aging effects on face images. *IEEE Trans Pattern Anal Mach Intell* 24(4):442–455
- Lin M, Chen H (2008) Labeling categories and relationships in an evolving social network. *Proceedings of the IR research, 30th European conference on Advances in information retrieval*:77–88

21. Matsuo Y, Mori J, Hamasaki M, Nishimura T, Takeda H, Hasida K, Ishizuka M (2007) POLYPHONET: an advanced social network extraction system from the web. *Web Semant Sci Serv Agents World Wide Web* 5(4):262–278
22. Mika P (2005) Flink: Semantic web technology for the extraction and analysis of social networks. *Web Semant Sci Serv Agents World Wide Web* 3(2–3):211–223
23. Miki T, Nomura T, Ishida T (2005) Semantic web link analysis to discover social relationships in academic communities. *Proceedings of the The 2005 Symposium on Applications and the Internet*:38–45
24. Min J, Jang S, Cho S (2009) Mining and visualizing mobile social network based on Bayesian probabilistic model. *Ubiquitous Intelligence and Computing*:111–120
25. Mori J, Ishizuka M, Matsuo Y (2007) Extracting keyphrases to represent relations in social networks from web. *Proceedings of the 20th international joint conference on Artificial intelligence*:2820–2825
26. Poole D (2000) Logic, knowledge representation, and Bayesian decision theory. *Computational Logic—CL 2000*:70–86
27. Quinlan J (1986) Induction of decision trees. *Mach Learn* 1(1):81–106
28. Rowe R, Creamer G, Hershkop S, Stolfo S (2007) Automated social hierarchy detection through email network analysis. *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*:109–117
29. Salton G, McGill MJ (1983) Introduction to modern information retrieval
30. Singla P, Kautz H, Luo J, Gallagher A (2008) Discovery of social relationships in consumer photo collections using markov logic. *Computer Vision and Pattern Recognition Workshops*:1–7
31. Stefanone M, Gay G (2008) Structural reproduction of social networks in computer-mediated communication forums. *Behav Inform Technol* 27(2):97–106
32. Tyler J, Wilkinson D, Huberman B (2005) E-mail as spectroscopy: automated discovery of community structure within organizations. *Inf Soc* 21(2):143–153
33. Vapnik V (1998) The Nature of statistical Learning Theory. *Data mining and knowledge discovery*:1–47
34. Wang C, Han J, Jia Y, Tang J, Zhang D, Yu Y, Guo J (2010) Mining advisor-advisee relationships from research publication networks. *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*:203–212
35. Wu P, Tretter D (2009) Close & closer: social cluster and closeness from photo collections. *Proceedings of the seventeen ACM international conference on Multimedia*:709–712
36. Yu K, Chu W, Yu S, Tresp V, Xu Z (2006) Stochastic relational models for discriminative link prediction. *Adv Neural Inf Process Syst* 19:1553–1560
37. Zhang T, Chao H, Willis C, Tretter D (2010) Consumer image retrieval by estimating relation tree from family photo collections. *Proceedings of the ACM International Conference on Image and Video Retrieval*:143–150
38. Zhao M, Cheng X, He Q (2009) An algorithm of mining class association rules. *Proceedings of the 4th International Symposium on Advances in Computation and Intelligence*:269–275



Elie Raad received the B.S. degree and the M.S. degree in Networks and Telecommunications engineering from the Antonine University in Lebanon in 2007. He is currently pursuing the Ph.D. degree at the LE2I laboratory – University of Bourgogne, France. From 2008 to 2011, he occupied teaching positions at the Antoine University and the ESIREM Department – University of Bourgogne. His research interests include automatic annotation of multimedia content applied on social networks.



Richard Chbeir received his PhD in Computer Science from the University of INSA-FRANCE in 2001. The author became a member of IEEE since 1999. He is currently an Associate Professor in the Computer Science Department of the Bourgogne University, Dijon-France. His research interests are in the areas of multimedia information retrieval, XML and RSS Similarity, access control models, multimedia document annotation. Dr. CHBEIR has published in international journals and books (IEEE Transactions on SMC, Journal of Methods of Information in medicine, JDIM, etc.), conferences (ACM SAC, Visual, IEEE CIT, FLAIRS, PDCS, etc.), and has served on the program committees of several international conferences (IEEE SITIS, ACM SAC, IEEE ISSPIT, EuroPar, etc.). He is currently the Chair of the French Chapter ACM SIGAPP.



Albert Dipanda received his Phd degree in computer science in 1990 from the University of Bourgogne, France. He is currently Professor in the Engineering School of Materials, Computer Sciences and Electronics and Deputy Director of the LE2I laboratory (Laboratoire d'Electronique, Informatique et Image). His research interests include motion estimation, MRF modeling in Image Processing, Image analysis, 3-D reconstruction and Genetic Algorithm applications in image processing.