Towards Automatic Privacy Management in Web 2.0 with Semantic Analysis on Annotations

Nitya Vyas*, Anna C. Squicciarini†, Chih-Cheng Chang*,
and Danfeng Yao*
* Department of Computer Science
Rutgers University
Email: {nityav, geniusc, danfeng}@cs.rutgers.edu
† College of Information Science and Technology
The Pennsylvania State University
Email: acs20@psu.edu

Abstract— **Sharing personal information and documents is pervasive in Web 2.0 environments, which creates the need for properly controlling shared data. Most existing authorization and policy management systems are for organizational use by IT professionals. Average Web users, however, do not have the sophistication to specify and maintain privacy policies for their shared content. In this paper, we aim to utilize personal and social annotations to develop automatic tools for managing content sharing, and demonstrate a new application of social annotations in access control. We use annotation data to predict privacy preferences of users and automatically derive policies for shared content. We carry out a series of user studies to evaluate the accuracy of our predicted techniques. We also perform extensive analysis on static and dynamic approaches of analyzing semantic similarities of tags, which is of independent interest. Our analysis gives encouraging results on the feasibility of using annotations for privacy management in Web 2.0.**

Keywords: information sharing, privacy, authorization, semantic similarity, annotation

I. Introduction

Web 2.0 revolutionizes how people store and share personal data and contents today. Desktop applications are being more and more replaced by Web services. Digital documents such as photos used to be kept on the owners' hard disks, whereas today sharing of personal information and documents on the Web is pervasive, from `flickr.com` for photo sharing to `myspace.com` for profile sharing and `facebook.com`, which has the highest image uploading rate among all social network sites. The change in how people share information is profound and has multi-facet implications, among which privacy is the most important aspect, i.e., how to ensure that the shared contents (e.g., pictures, videos, hypertexts) are not used against the owner's will. If contents are stored on a user's PC, the control of sharing can be done explicitly, e.g., the owner shows certain photos to trusted guests and the owner decides what to share and what not. The equivalent technique in the digital world is *access control*, which is the art of defining and determining the privileges of users to certain resources. The focus of conventional access control literatures are more on the security and robustness of the authorization systems and less on the usability [30]. As conventional authorization policies are designed for use by trained professionals (e.g., system administrators), they are complex to manage and use [5], [30].

As a result, users are exposed to a number of privacy threats [34]. A significant privacy threat is raised by an increasing amount of media content posted by users on Web 2.0 platforms. User-provided digital images are an integral and exceedingly popular part of profiles on social network sites. For example, Facebook hosts 10 billion user photos (as of 14 October 2008), serving over 15 million photo images per day [4]. Pictures are tied to individual profiles and often either explicitly (through tagged labeled boxes on images) or implicitly (through recurrence) identify the profile holder [1]. Such pictures are made available for other SN users, who can view, add comments and, by using content annotation techniques, can add hyperlinks to indicate the users who appear in the pictures.

Web 2.0 users have to take the responsibility to manage the access of their shared contents. Although social networking and photo sharing websites provide mechanisms and default configurations for data sharing control, they are usually not intuitive, and many users do not take the appropriate time to configure their privacy preferences [2]. This type of sharing control mechanisms do not effectively protect users' content, and have resulted in privacy breaches of shared data in Web 2.0. As documented in the public news media [29], user-provided content can be stolen, sold, used for blackmailing and have serious consequences, such as stolen identities and financial losses.

Directly borrowing conventional access control approaches to Web 2.0 is not a suitable solution, as both paradigms have drastically different requirements for the authorization model. In Web 2.0, the emphasis for such models is on the *usability and manageability*. In traditional information systems, resources are owned by an organization and controlled by a team of trained professionals, whereas in Web 2.0 environments, content owners are individuals who may not be technology-

This work has been supported in part by NSF grant CCF-0728937, CNS-0831186, and the Rutgers University Computing Coordination Council Pervasive Computing Initiative Grant. The work from Squicciarini has been supported by NSF Grant CNS 08-31247.

savvy. A personalized, quantified, and easy-to-use method for users to manage their shared contents in Web 2.0 environments is highly desirable in order to protect the personal information of participants.

In this paper, we take the first step to address the challenge of automated privacy management by presenting an *automatic policy generator based on the semantic analysis of annotations and social communities*, referred to as *APPGen* (standing for Automatic Privacy Policy Generator). Our approach takes advantages of user-specified annotations, i.e., tags. The purpose of tagging is to help users organize and maintain their own contents – profiles, photos, blogs, or videos – with free-form keywords, i.e., tags. We leverage personal and social group annotations to develop automatic tools for managing content sharing.

Our technique utilizes folksonomy [21] and semantic similarity analysis for automatically inferring policies in content-based access control. Folksonomy is different from traditional taxonomy in that tags used to label and classify Web 2.0 contents are generated by users, not by certain authorities. Specifically, the APPGen system draws knowledge from two main sources: i) the similarity of users in a group of related users; ii) a pre-defined privacy profile of the user. We demonstrate the potential of our new approach by experimental evaluation and user study, which show promising initial results. Our contributions are summarized as follows.

1) We describe a new framework for automatically inferring the privacy policies for personal Web 2.0 contents, which is to improve the privacy, usability, and manageability of personal contents. The framework produces privacy policies for the content owner based on a small amount of annotation information.
2) We design privacy inference mechanisms based on the relatedness of new contents to existing knowledge by utilizing a k-means clustering method for discrete objects. Specifically, we implement three independent privacy inference techniques:
   - *social group analysis*
   - *personalization with static tag classification*
   - *personalization with dynamic tag clustering*
3) We carry out a Web-application based user study to evaluate the accuracy and usability of the privacy inference system. Our experiments show that the majority of the participants think that the framework is accurate in inferring the privacy policies. 94% of the participants voted the policy generated using our tag clustering technique as the best policy in terms of both accuracy and closeness with their "ideal policy".

The rest of the paper is organized as follows. We give formal definitions for concepts used in our framework in Section II. Our privacy management framework is presented in Section III. In Section IV, we describe our approaches of computing similarity among tags and clustering similar tags for dynamic classification, respectively. Our experiments are described in Section V. Related work is given in Section VI. Conclusion and future work are in Section VII.

II. Model and Definitions

In this section, we provide the fundamental notions underlying our solution. We cast our techniques for managing content sharing in the context of a social application, called *APPGen*. APPGen helps social-network users or bloggers predict privacy policies of their shared content. The framework allows users to annotate their content (hypertext, pictures, or videos) using *tags*. A *tag* (τ), or social annotation, is a single English word, freely chosen. The APPGen framework predicts a privacy policy for the content just added, based on semantics of the tags, leaving the user the option to accept or decline the predicted policy. In an *initialization phase*, APPGen requires the user to explicitly indicate some general topics of her interest, along with her privacy preferences, as she creates a Web space in the considered domain. This initial set of topics is then dynamically updated by APPGen as new content is added to the user's Web space. A simple use scenario of APPGen is as follows.

Example 1: Suppose Alice is a new blogger, and she wishes to create her blog within the *TheSpotToBlog* social network, to reach out to old friends, and share her pictures taken while working on her favorite hobbies and activities. In the initialization phase, Alice generates a simple *privacy profile* where she indicates her topics of interests and *sensitivity values* possibly associated with the topics. This setup is a one-time process.

Alice updates her Web space over time, and annotates the added contents using tags. As she adds new content, APPGen predicts a privacy policy to be applied to the uploaded material. Alice can choose to accept it or modify it as she wishes. In this paper, we assume one tag per content for our application but this can be easily generalized to more than one tag.

A. Definitions for Social Network and User Profile

In the rest of the paper we cast the presentation of APPGen's features in the context of a social network site. We notice that social networks represent only one of the possible social computing platforms where APPGen could be successfully used. The requirements for APPGen to guarantee accurate predictions, are the use of annotations and, as discussed later in the paper, the existence of users who are *similar* to the user in the same domain. Hence, policy predictions can be applied in other Web 2.0 platforms, such as blogs, wikis, etc.

We begin our formal presentation by defining social networks, tags, and users profiles.

- A *social network* is denoted by the tuple $\langle U, \mathcal{R} \rangle$, where U denotes a collection of users U, connected by social relationships $\mathcal{R}$ of different types $\{R_1, \ldots, R_k\}$. (e.g., family, friends, colleagues, school network). We assume relationships to be explicit and mutually accepted by the involved users. For simplicity we focus on binary user relationships, and denote a relationship as $u : R : u'$, being u and u' users' unique identifiers, and R the relationship that connects them. By assumption, each user is connected by at least one relationship to another user in U.

- Each user $u \in U$ has one associated Web space or *profile*, $prof$. Each $prof$ is related to one or more topics $\gamma_1, \gamma_2, \ldots, \gamma_k$ indicated by the user at the time of registration. A *topic* or a *subject* is a word that represents an area of interest or a concept. For example, a topic may be: *alcohol, adult, religion, schoolwork, sport, technology, travel, food, animal*, or *gathering*. We assume the existence of a pre-defined set of general topics Γ, which can be dynamically expanded. We assume set of topics to be universal i.e., they are known to everyone.

Users populate their profiles (or Web spaces) by adding content of different types, and content can be annotated with tags. Users profiles offer a large amount of information, which, if well correlated, can be leveraged to do accurate predictions regarding users' attitudes. Such groups, referred to as *Social Group* represent cluster of users, sharing certain properties, such as their relationships, their interests, etc.

Definition 1 (Social Group): Let $SocG$ be a subset of users in U. $SocG$ is a *social group* if and only if at least one of the following condition is satisfied: group of friends of a user who

1) $\forall u' \in SocG$ there exists $\gamma \in \Gamma$ s.t. γ is associated to $prof'$,
2) $\forall u, u' \in SocG$ there exists a relationship u:R:u'.

The definition identifies social groups as groups of users who share a topic of interest (condition 1), are connected through a social relationship (condition 2) or both. This notion is useful to identify correlated users, in case not enough user information is available to accurately predict a policy. Social groups are also important to infer whether users with similar features are predictive of certain privacy preferences. We further elaborate on the notion of social groups in the next sections.

B. User's privacy policies

Expressing privacy preferences with APPGen is a simple task. The user simply has to assign a sensitivity score to the topics of interest, and indicate her privacy preferences. A *sensitivity value* for a topic γ is a non-negative numerical value w that a user u assigns to γ to indicate the degree of reluctance to share the contents related to it. We model the indication of users' preferences by means of a *user expression*.

Definition 2: (User Expression) A user expression is an expression of the form ($\{R_1, \ldots, R_k\}$, Cond); where:

- $\{R_1, \ldots, R_k\}$ is a list of relationship kinds, and $R_i, i \in [1, k]$ is a relationship in $\mathcal{R}$.
- Cond is a boolean formula, against user profile attributes.

Example 2: Suppose that Alice indicates her preferred topic as 'photography', at time of registration. As part of the registration process, she indicates that photography is an interest she is willing to share with friends and relatives. This preference is summarized by the expression ({Friends, Colleagues}, $\emptyset$), since no further conditions are enforced. If sensitive content is added regarding the photography, she wants only friends which High School is 'Art School of London' to access her profile portion. In the latter case the expression used will be of the form $(\{Friends\}, HighSchool = ArtSchoolLondon)$.

User expressions represent the building blocks for both privacy profiles, and privacy policies. The collection of sensitivity values along with related user expressions for the topics in $prof$ define the *privacy profile* of a user.

Definition 3: (Privacy Profile) Let u be a user in U, and $prof$ be her profile. The privacy profile p of u is the list $[tup_1, \ldots, tup_n]$, where each tup$_i$, $i \in [1, n]$ is a tuple of the form $\langle \gamma_i, w_i, UExpr \rangle$, where γ_i is a topic, w_i the associated sensitivity value and $UExpr$ a user expression, specified according Definition 2.

A compact representation of the privacy profile of a user u is synthesized as a vector $\overrightarrow{p_i} = [w_1, \ldots, w_n]$, where w_j is the sensitivity score for topic γ_j. As well as topics, tags are also coupled to a sensitivity score w, which value is subjective to the individual's privacy inclination. As we return later in the paper, this score is not manually input by the user, unless she wishes to do so, but inferred by APPGen.

Having introduced users expressions, we are now ready to discuss privacy policies. In our context, a *privacy policy* (or policy for short) controls the access of a user's content. Given a Web space composed of multiple objects, the privacy policy applies to only one of these contents. The privacy policy specifies the scope of sharing, i.e., who is allowed to access the object/s posted in the profile.

We provide a simple representation of privacy policy.

Definition 4: (Privacy Policy) Let $prof$ be the profile of a user u, and let c be some content in $prof$. A privacy policy *pol* is modeled as a predicate AccessTo(UExpr, Mode), where $UExpr$ is a user expression specified according to definition 2, *Mode* is a subset of admitted access modes that consists of view, modify, execute and delete.

According to the definition, a policy constrains the set of users who can access certain content, based on the content sensitivity (namely, the w component) and on the viewers' properties (i.e., the user expression). The mode component indicates the granted access privilege.

Example 3: Examples of policies are: *AccessTo(({U2Fans}, ϕ), read)*, and *AccessTo(({}, ϕ), read;write, { pet $\in$ $prof$})*. The first policy is an example of policy with no access condition, while the second policy allows read and write operations to users who indicated *pet* in their preferred topics.

III. APPGen Privacy Policy Inferencing

The main goal of APPGen is to provide a semi-automated approach to privacy protection. A central technical question is, given the annotation of a content, how to infer the intended privacy policy for the user, while minimizing her intervention as possible.

As introduced, a privacy policy essentially specifies which users are allowed to view the tagged content (say, s) of a user's profile. We can identify several approaches according to which a policy for some content s can be selected. The trivial approach would be to simply apply default policies according to the broad topic the tag falls into, and use the user's specified policy for the topic. Clearly, this approach would not allow fine-grained specification of policies, nor it would capture the user's inclinations with regards to content sharing. The opposite approach would require the user to continuously add policies each time new content is added, failing to provide any automation. APPGen overcomes the limitations of these approaches by using inferencing techniques to identify the *best* policies for some newly added content. Specifically, the system draws knowledge from two main sources: i) the similarity of users in a group of related users; ii) the sensitivity values of the content specified by users.

We describe three main approaches, **i)** *personalization with static classification of tags*, **ii)** *personalization with dynamic clustering of tags*, and **iii)** *social-group based analysis*, for the privacy policy inference. The inference mechanisms are complementary to each other and can be integrated to yield a hybrid approach.

A. APPGen Policy Personalization

Given the inputs of a tag and a set of pre-defined topics or the user's previous tag history, the personalization process outputs an appropriate privacy policy for some annotated content. The personalization component will first utilize semantic analysis techniques to discover the most similar tag in the topics or the user's profile, and then apply the appropriate policy accordingly. We present two different approaches for policy personalization, namely *static classification* and *dynamic clustering*. The two approaches are independent of each other.

- **Static Classification of Tags** utilizes a set of pre-defined topics (typically around 20), and aims to assign the tag τ to a topic γ that is semantically most similar to τ. Semantic similarity analysis is presented in more details in the next section. Once the topic γ is chosen, the user expression $UExpr$ associated to γ is used as the policy for the content tagged with τ.
- **Dynamic Clustering of Tags**. The analysis is between tag τ and all the previously annotated contents in the user's profile $Prof$, in order to identify the most similar content. In particular, we aim to discover a tag τ' in the user's history that is semantically most similar to tag τ. When such a tag τ' is found, the policy associated with τ' is applied to τ and the content associated with τ. In the dynamic clustering approach, the analysis is between tag τ and all the previously annotated contents in the user's profile $prof$, in order to identify the tag most similar to τ. In particular, we aim to cluster the tags in user's personal profile $prof$ into several groups based on tag semantic similarities, and then discover a cluster c whose cluster center is semantically most similar to tag τ. The cluster center is a tag in $prof$. When such a cluster and its center tag are found, the policy associated with the center tag is applied to τ and the content associated with τ.

The above approaches are called *personalization* because the analysis is based on the user's unique personal profile, as opposed to a set of uniform and generic rules defined by the system for every user.

B. APPGen Social Group Analysis

Social groups analysis is an alternative, yet equally powerful approach, to automatically generating privacy policies for annotated contents. The main idea is to leverage those users who have similar privacy preferences as the focal user (i.e., the user whose policy needs to be predicted), and to derive privacy policies based on their policy records and profiles. The users who have similar privacy preferences as the user are called *reference points* by us. We require the users who serve as reference points in this analysis to belong to the social group of the user. The purpose of this requirement is two-fold: to restrict the scope of reference points and to speed up the computation.

Once users have performed the one-time registration and we have obtained their privacy profiles that contain their specified sensitivity values for a set of pre-defined topics, a social group for a focal user can be identified. Precisely, given a certain user u and some content s tagged with τ, we identify u's social group $SocG$ (see Definition 1), as indicated by the users' specification. Users may specify how to select a social group that they belong to, by joining existing groups (aka. networks), or by indicating their own. Subsequently, the similarity of the user u with the users in $SocG$ can be computed.

In order to infer policies based upon the user's social group information, we first compute the *similarity of profiles*, that is, the similarities between a user's profile and group members' profiles. Formally, we denote $sim(p_u, p_v) \in [0, 1]$ as the similarity between user u and v where p_u and p_v are the privacy profiles of user u and v, respectively. Cosine similarity in Equation 1 (or more complex Pearson correlation coefficient) can be used as the similarity function. We use cosine similarity mainly because of its simplicity. The similarity is commutative, i.e., $sim(p_u, p_v) = sim(p_v, p_u)$.

$$sim(\overrightarrow{p_i}, \overrightarrow{p_j}) = cos(\overrightarrow{p_i}, \overrightarrow{p_j}) = \frac{\overrightarrow{p_i} \cdot \overrightarrow{p_j}}{|\overrightarrow{p_i}||\overrightarrow{p_j}|} \quad (1)$$

We then sort the similarity scores and identify the most similar user (i.e., the most similar reference point). To obtain the privacy policy for tag τ, we directly apply the policy existed in this reference point's profile. For example, if the reference point, say Bob, has given the policy *pol* to tag τ, then policy *pol* is returned at the end of the social group analysis. This method can be generalized to consider top-k similar reference points. This generalized top-k method will increase the chance of locating tag τ in the reference points profiles. We do not handle the situation when there are more than one reference points for the user in the application. In case the tag τ cannot be found in the top-k profiles, the aforementioned personalization and semantic analysis techniques can then be incorporated, which are not limited to the syntax of words.

Note that in order for the inference to be feasible, the users' profiles in the social groups must be already populated with content, and users must have posted tagged content. As such, there is a necessary training phase during which the users cannot enjoy the advantages of the social groups'. This problem is well known in recommendation systems as the *cold start problem*. Essentially, the problem arises in case of lack of historical data to use for inferencing. To solve the cold start problem, our personalization approach with static tag clustering can be used in combination with the social group analysis. Due to space limit, we omit the details of this discussion in this paper.

IV. Semantic Similarity Analysis

The semantic similarity analysis among tags plays an important role in our automatic privacy management framework described in Section III for APPGen personalization. The user-user similarity described in Section II is for comparing users' privacy profiles and utilizes well-known metrics presented in equation 1. In comparison, the semantic similarity of tags is more challenging and requires developing and evaluating new methods beyond the existing semantic analysis tools.

Our semantic similarity analysis problem is as follows. *Given a tag τ associated with a new content, how to find the tags that are semantically most similar to τ among the tags associated with existing contents.* Once the most similar tags are located, our privacy policy inference method described in Section III can be used to derive the sensitivity score of the tag τ and thus privacy policy for the new content. This inference process does not require user's participation and is automated.

The building block of all our semantic analysis is the pair-wise word similarity metric. Given two words w_1 and w_2, a similarity metric computes the words' semantic similarity or relatedness $sim(w_1, w_2)$ based on certain measurement. There exist several proposals on how to measure the semantic similarity of two words, including Jiang-Conrath [13], Resnik [28], Lin [19], Banerjee-Pedersen [3], and Pirro'-Seco method [26]. All of these above-mentioned metrics use Wordnet [32] as the dictionary, which is a large lexical database of English. We refer readers to artificial intelligence literature for detailed methods of computing semantic similarity [28], [26]. An online Wordnet similarity tool implementing several measures is available [24]. In our implementation, we evaluate different similarity metrics with the focus on the most recent approach by Pirro' and Seco [26]. Their metric has been demonstrated to have good prediction accuracy by human users.

A. Static Classification of Tags

As described earlier, the static classification of a tag involves assigning the tag to one (or more) pre-defined topics based on the computed semantic similarity of the tag-topic pairs – the topic that is semantically most similar to the tag is chosen. Then, based on the chosen topic, we can derive an appropriate policy for the tag. To evaluate whether semantic similarity measures can be used to map a tag to one of the pre-defined topics, we manually choose a set of topics (20), each representing a general category. The topics are *alcohol, adult, religion, schoolwork, sport, politics, news, business, culture, technology, gathering, food, animal, pet, people, travel, relationship, entertainment, nature*, and *family*. We obtain 1544 tags from Flickr.com using the Flickr API. The tags were from the most popular photos on August 29, 2008. Some of the tags are non-English words. We use Wordnet to filter out these non-English words, by keeping the ones that can be found in Wordnet. We further remove identical words, which leaves 914 distinct tags. Our evaluation procedure is given in Appendix B.

The static classification relies on a set of pre-defined topics and thus is limited in its ability of locating the most suitable topic for a given tag. For example, if the tag represents a new concept that is not yet incorporated by the topics, the static classification may give inaccurate result. To improve the classification of tags and to group similar tags with high accuracy, we utilize a new clustering method for words, which is presented and analyzed next.

B. Dynamic Clustering of Tags

To accommodate the dynamic aspect of folksonomies, we apply a machine learning technique, namely k-means clustering, to cluster tags based on their pair-wise semantic similarity. Dynamic classification of tags does not require pre-defined topics, instead, the method needs a large number of tags as inputs. Given a new tag τ, the method outputs a cluster of tags that is semantically most similar to τ. Then, based on the cluster information, we can derive an appropriate policy for τ. We carry out a set of experiments to investigate whether we can *automatically* group tags into clusters, each of which may represent a topic. Reclustering the tags periodically may be necessary as the cluster size gets bigger to improve the fine granularity of categorization.

Next, we briefly explain k-means clustering algorithm. Integer k in k-means clustering specifies the number of clusters being sought. We do not attempt to find a generalized value of k in this algorithm. Once k is determined, k data points are chosen at random as cluster centers, and all instances are assigned to their nearest cluster center according to a certain distance metric, e.g., typical Euclidean distance. At the next iteration, the centroids, or the means of the points in each cluster are computed that are taken as the new cluster centers for their respective clusters. The iteration terminates until an equilibrium is reached, i.e., the cluster assignments stop changing. k-means algorithm is simple and finds a local minimal, i.e., with respect to the cluster centers, the total distance of the instances to their cluster centers is minimized. We refer readers to machine learning literature for details about k-means clustering algorithm [22].

Conventional k-means algorithm does not work for discrete objects, and only works for numerical data. In order to use k-means to cluster words, the cluster recenter step of the algorithm needs to be modified. Instead of choosing the

Cluster	fruit	indian	motion
Tags	flower	american	play
	cinnamon	persian	bw
	nature	iranian	crossing
	hair	barrage	reentry
	whiskers	aussie	jump
	seed	czech	art
	beard	irish	morning
	shoot	cuban	tilt
	wool	chinese	flying
	saskatoon	creek	flight
	delicious	italian	surprise
	cane	european	reflection
	europa	chin	drop
	chameleon	russian	travel
	watermelon	japanese	flare
		inca	kill
		inka	laugh
		tongue	buzz ......

TABLE I

EXAMPLES OF CLUSTER OUTPUTS.

cluster center (i.e., means) as the new cluster center, we choose the object (i.e., tag) that is closest to the centroid. For completeness, we describe the k-means clustering for discrete objects (discrete k-mean for short) in the Appendix A.

Privacy inference using clustered tags For our privacy inference purpose, clustering is done on existing tags of the user or his social group. Each of the existing tags is already associated with a sensitivity score as we defined in our framework in Section II. Given a new tag associated with a new content, we need to decide (1) which cluster c^* this new tag τ^* belongs to, and (2) what is the inferred sensitivity score w^*. To locate c^*, we compute the average distance from τ^* to members of a cluster and choose the cluster that gives the minimal distance value as in Equation 2, where $|c_i|$ is the size of cluster c_i. Then, the sensitivity value w^* is computed as the average sensitivity score of the cluster as in Equation 3. This clustering and new tag assignment operations can be updated and carried out dynamically. We do not describe here in details of how the dynamic data analysis is realized.

$$c^* = argmin_{c_i} \sum_{\tau_j \in c_i} sim(\tau_j, \tau')/|c_i| \tag{2}$$

$$w^* = \sum_{\tau_j \in c^*} w_{\tau_j}/|c^*| \tag{3}$$

Our clustering analysis is run on the same set of 914 Flickr tags with k being 50 and the k-means running for 10 iterations. We have also experimented clustering runs with 30 and 50 iterations that produce different clusters with similar quality. Table I gives examples of cluster outputs. Compared to classification, clustering provides a holistic picture of pairwise similar tags, rather than based on a single point of computation. The words grouped into one cluster must be similar to one another, thus creating a web of inter-connected words. As the inter-connectivity among tags are based on multiple similarity values, misclassifying a tag into a wrong cluster is less likely. On the other hand, for pre-defined topics, classification *solely* depends on single tag-topic similarity values, which is less robust. may not be accurate. Therefore, *clustering is a more robust method for finding semantic related tags than assignment to pre-defined topics.*

V. IMPLEMENTATION AND EVALUATION

We evaluated our approach by implementing a APPGen prototype and conducting a user study involving 50 participants. Our goal was to examine the accuracy of the APPGen techniques, in inferring users' most appropriate privacy policies based on the input provided both at the time of registration and during the users' lifetime within the social network.

A. Experiment Setup and Methodology

The implementation of our prototype consists of a Web server and a backend database that run on a Fedora 8 Linux machine. We used Apache Tomcat 5.5.27 as the Web server to run JSP and servlets. We also used MySql 11.18 Distrib 3.23.58 for redhat-linux-gnu (i386) as the database. All the JSP and servlets are implemented in Java and HTML/CSS. For the Wordnet similarity, we use the Pirro' and Seco implementation Java Library [26]. We use MySql java connector library for the data insert/update/retrieval. Clustering based inference uses 914 Flickr tags. Finally, we use `Surveymonkey.com` to host the survey. For simplicity, we assign all participants into the same arbitrary social group; the social group analysis is based on *the most similar user* among all the participants. In the setup of the clustering method, we assign synthetic sensitivity scores to 914 Flickr tags (See also Section VII).

In user study, we asked each participant to register to a fictitious social network, where they could create their own blog. At registration, we asked them to provide their privacy preferences of 20 pre-defined topics (listed in Section IV-A) on a sensitivity scale ranging from 1 (least sensitive) to 10 (most sensitive). Then, we asked the participants to tag three pictures (about cocktail parties, traveling in London, and party drinking, respectively). The selected pictures were purposely very different from one another, and with content that could potentially be interpreted sensitive. The tool, each time a picture is tagged, produces three types of policies based on our three privacy inference techniques (i.e., (1) *social group analysis*, (2) *personalization with static tag classification*, and (3) *personalization with dynamic tag clustering*). The policies resembled the policies described in Section III, although they do not include conditions, for simplicity. Specifically, the policies can include one or more of 10 pre-defined relationships, such as *Public, School/University Network, Friends of Friends, Local Community, Colleagues, Friends, Good Friends, Relatives, Best Friends*, and *Family*. We selected these groups as they reflect the most common relationships, and are general enough to summarize all possible relationships among social network users. We plan to explore more expressive relationships in the future work.

Participants were then asked to complete a post-session questionnaire. In order to evaluate the most effective technique we formulated questions using two different methodologies, namely *vertical comparison* and *horizontal comparison*. For the horizontal methodology we required each participant to evaluate *individually* each policy generated by a specific technique. For each prompted policy we asked the participants three separate questions: to rate the overall perceived sensitivity of the picture, to state whether they thought it was a policy similar to their privacy inclinations, and to indicate whether the policy was appropriate for the content. The *vertical comparison* approach, instead, required the participants to compare the policies generated for a same picture. For each picture, we asked the participants to evaluate the three prompted policies and select the one that they perceived as the most adequate in terms of closeness with their thoughts, the most conservative in terms of privacy, and the most adequate with respect to the content. At the end of this procedure, the participants had to compile an exit questionnaire, where we asked some biographic information.

Notice that an alternative design to the one described above would be to ask the participants' to manually input policies and compare them with the system's suggested ones. However, this approach is error-prone, as it depends on analysis of policies' similarity. Also, participants' would have the burden of commenting on their choices in order to make such approach effective.

Technique	Adequacy	Closeness
Social group analysis	19%	26%
Static tag classification	38%	26%
Dynamic tag clustering	43%	48%

TABLE II

RATING OF POLICIES IN PERCENTAGE. POLICIES RATING REFERS TO ADEQUACY FOR THE CONTENT AND SIMILARITY WITH USER'S PRIVACY PREFERENCE

B. Experimental Results

Our initial sample consisted of 50 participants recruited using fliers. 15 participants had an age of under 20, 20 were aged between 20 and 25, and 15 were older than 25. Out of the 50 participants, 8 of them were not social network users. While 8 participants did not have their own blog the number of readers were higher, roughly 44 out of 50 participants declared they were blog readers, with varying degree of frequency. Data were discarded for all respondents who completed less than 80% of the tasks. For participants with modest amounts of missing data, we used a simple data imputation method that has been found to be quite effective for factor analysis [7]. Specifically, we substituted item means (rounded to their integer value) for missing responses if a respondent omitted 1 item on a short scale (10 items or less) and up to 2 items on longer scales (more than 10 items). No imputation was used when 2 or more items were missing on short scales or 3 or more items were missing on long scales; rather, those participants were dropped from analyses involving these scales. Our final sample included answers of 42 participants.

1) Analysis Techniques and Participants' Preferences: According to the responses collected under the vertical comparison methodology, the policies were rated in terms of closeness with user's inclinations and adequacy of the policy with respect to the content. On a Likert scale from 1 (strongly agree) to 5 (strongly disagree) on the questions on both similarity, both personalization techniques (static tag classification and dynamic tag clustering) are rated equally well with a negligible difference, average 2.22 (agree) with standard deviation of 0.65 for static classification and 2.29 and sd=0.84 for dynamic clustering[1]. The policy returned by the social group method is rated at 2.5 (sd=0.721). Similar results were reported for the answers on adequacy of the policy with respect to the content. The lack of popularity for the social group technique can be motivated by the following considerations. First, in about 44 % of the cases, static tag classification produced a very similar policy to the social group technique. Users may select the static classification based policy for convenience, as it is listed at the top of the Web page (we did not scramble the ordering of policies when prompted to users). Second, we had to generate some synthetic data to bootstrap the social group technique. The synthetic data may have skewed the actual results, in that the randomly generated records may not be realistically significant for the similarity analysis.

The results from the horizontal comparison are reported in Table II. Interestingly, the results do not exactly reflect the responses obtained using the vertical comparison. Thanks to this latter set of questions, we can clearly disambiguate the attitude of respondents' with respect to the prompted policies. When it comes to comparing the policies and select one policy over another, respondents preferred the clustering technique.

As reported, in fact *the personalization with dynamic tag clustering technique outperforms the others, both in terms of perceived adequacy and closeness*. 94% of the participants voted the policy generated using the dynamic tag clustering technique as the best policy in terms of both accuracy and closeness with their privacy preferences. Votes were differentiated for policies generated by the other techniques, where the participants paired the answers about 80% of times. When they differentiated the answers, it was in most cases (90% of the cases) to indicate a more stringent policy as the most adequate one.

On top of the analysis of above, we analyzed further our data, by running regression analysis for all three techniques. We used as independent variables age, social networks, and pictures' sensitivity as rated by the participants. These regression coefficients for our independent variable summarize the effects of the independent variable on the dependent variable when the effects of the other independent variables included in the regression analysis are controlled for or held constant.

The bivariate relationships obtained were inverse: as the sen-

[1] The mean and standard deviation can only be calculated for interval and ration data. Many researchers argue that it is unclear whether Likert scales have interval properties. Nevertheless, Likert scales are often assumed to have interval properties (some researchers even refer to them as quasi-interval) and the mean and standard deviation are often reported[23].

sitivity variable increased (that is, as participants perceived the pictures to be less sensitive) perceptions of policy generated by static tag classification decreased. So the policy generated with the static method was evaluated more positively when the pictures were more sensitive. The older people were, the less positively they evaluated the policy by static tag classification. Likely, this result can be justified by the fact that we noticed a tendency of *younger participants of rating the same pictures less sensitive than the elder observers*. Therefore, young users do not perceive stringent policies as useful.

We report the results for this technique in Table III. The table Coefficient gives results of the regression analysis while the model summary table reports the summary of results. In the Unstandardized Coefficients part of the Coefficient table, two statistics are reported: B, which is the regression coefficient, and the standard error. Notice that there are few statistics reported under B: one labeled as (Constant), age, soc_net, blog, sens_mod. These statistics are the regression coefficients. The t-test (labeled as t) tests the significance of each b coefficient. The *sig* value indicates the confidence level. A *sig* below 0.05 indicates that the predictor is significant.

No other predictors for the other techniques were found, although there are some clear tendencies for the clustering technique. The regression analysis showed that sensitivity of the picture is close to be a predictor variable (the significance variable is slightly below the threshold). The more sensitive is the picture, the more participants appreciated the policy.

The lack of other significant predictors can be due to the relatively small sample size we had available. In light of the overall positive feedback obtained by the study, we interpret this as an encouraging sign. To certain extent, it implies that *no technical understanding of tags and blogs is required to appreciate our approach*. However, no stronger claims can be done at this time, and we reserve this investigation for future studies.

2) Increased privacy awareness: As part of our study, we asked users at the end of the experiment an overall opinion on this type of predictor tool and whether they thought this would be beneficial to them. The results obtained by these answers are extremely satisfactory, and clearly justify our efforts. *92 % of the respondents embraced the idea of a tool being able to adaptively provide privacy protection with little effort from the user end.* As this ideology is the goal of our APPGen framework, we feel that this outcome confirms our hypotheses of the need of APPGen type of tools. 86% of the respondents felt that tools like ours will increase their privacy awareness, and better protect their privacy. Finally, 83% of the respondents expressed a positive opinion over the intention of using APPGen as a predictor tool for their current blog.

VI. Related Work

Several solutions related to the access control management in Web 2.0 environments [10], [6], [11] have been proposed. Carminati, Ferrari, and Perego proposed a rule-based access control model for online social networks [6]. Their solution requires data owners to issue digital certificates to participants in their social relationships. The certificates are then used for enforcing the access control rules that the data owners define. Digital certificate is an important security primitive that has been demonstrated useful in numerous e-commerce settings, e.g., online banking and online shopping. However, the process of generating and verifying digital certificates requires a relatively high degree of sophistication from the users, which may not be appropriate in Web 2.0 settings. In comparison, our framework is easier for average Web 2.0 users to learn and use, as access control policies are automatically generated based on social annotations rather than specified by the data owners. Compared to the work [6], a more practical but coarse-grained solution for enforcing social relationship was proposed by Mannan and van Oorschot [20]. Their idea is to leverage the existing circle of trust in Instant Messaging (IM) networks.

Gates [9] has described relationship based access control as one of the new security paradigms that addresses the requirements of the Web 2.0, whilst [11] proposed a content-based access control model, which makes use of relationship information available in SNs for denoting authorized subjects. However, those frameworks yet rely on the users input indicating their access control policies for each protected object, in order to effectively protect users' privacy.

There has been much work on the customization and personalization of tag-based information retrieval [16], [27], [17], [33]. Several techniques involved in exploring social annotations include association rule mining [17] and EM-based probabilistic learning approach [16], [27], [33].

The ability to evaluate the semantic similarity of words has important applications in many research fields such as psychology, linguistics, cognitive Science, and biomedicine. Semantic similarity measures and tools are mostly developed by the natural language community. Most of the word similarity measures make use of Wordnet [32], and these include Jiang-Conrath [13], Resnik [28], Lin [19], Banerjee-Pedersen [3] and Pirro'-Seco method [26]. The above metrics cannot be applied to phrases, as Wordnet does not contain general phrases. To address this limitation, a solution for assessing phrase similarity is proposed by measuring the edit distance of parse trees and single term similarity [31]. Sentence similarity has also been studied using corpus statistics and lexical databases [18]. Motivated by the need of Web 2.0 privacy management, our work studies the categorization and clustering properties of a large number of words based on their semantics, which differs from the existing word-word semantic analysis.

Clustering methods have previously been used to cluster documents for information retrieval purpose [14], or group contexts in a large corpus of text, for example, Kulkarni and Pedersen developed SenseCluster by analyzing the lexical features and co-occurrence of phrases [15], [25]. Our clustering method differs from the existing bisecting spherical clustering approach in that we leverage the quantified distance (i.e., similarity) values provided by Wordnet, and are able to significantly simplify the k-means algorithm to meet our needs.

Model	Unstandardized Coefficients		Standardized Coefficients	t	Sig
	B	StD Error	Beta		
(Constant)	3.645	.479		7.602	.000
age	-.428	.174	-.304	-.2455	.016
blog	-0.45	.106	.046	-.420	.675
soc_net	.102	.110	.118	.934	.353
sens_mod	-.197	.082	-.265	-2.418	.018

TABLE III

REGRESSION ANALYSIS RESULTS FOR THE PERSONALIZATION WITH STATIC TAG CLASSIFICATION.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we utilize personal and social group annotations to develop automatic tools for managing content sharing. Our APPGen is a privacy policy generation framework that enables automatic generation of access control policies for users' contents. Our main approaches are to utilize static and dynamic semantic similarity analysis and social group structures for automatically inferring policies in content-based access control. Using tags and social networks in Web 2.0, we demonstrate a new security application of social annotations beyond conventional knowledge discovery and personalized information retrieval. We show the feasibility of our new approach by experimental evaluation and user study.

Although promising, our system has several limitations, that we plan to investigate in the near future. First, our approach on social group analysis needs to be refined to achieve its full potential. One may argue that similar users do not have similar privacy preferences. Hence, inferring from social groups may not always be accurate. We will further explore this issue by carrying out some comparative analysis between social groups that take into account users' privacy inclinations against groups that are purely based on other similarity features. Second, the semantic similarity analysis can be certainly improved. We plan on doing so by using the Wikipedia based explicit semantic analysis by Gabrilovich and Markovitch [8]. Instead of using synthetic sensitivity scores for clustering, we will explore how to use social annotation and personal profile to infer the *average* sensitivity scores for clustered words and its impact on the score of the new tag. Also, how a selection of random policy affects experimental results needs to be measured. The semantic analysis could include multiple tags, rather than a single tag for picture. Finally, users studies in larger scale are certainly desirable, to confirm our findings on a larger population. As part of this extension, it remains to be investigated whether, for legal purposes, certain levels of privacy are to be guaranteed, regardless of the user's actual input.

APPENDIX

A. Discrete k-means algorithm

1) Arbitrarily choose k tags to be cluster centers and denote them as $\tau_{c_1}, \ldots, \tau_{c_k}$. Denote the k clusters by $c_1, \ldots, c_k$.
2) **Cluster assignment** For each tag τ_i: Add tag τ_i to the nearest cluster c_j, $j \in [1, k]$ according to a distance metric defined as the inverse of $sim(\tau_i, \tau_{c_j})$.
3) **Cluster update** Choose the new cluster center as the tag that is closest to the centroid of the cluster. If the new cluster center is the same as the previous one, then an equilibrium is reached and the algorithm terminates. Otherwise, repeat from Step 2.

B. Evaluation on Static and Dynamic Classification of Tags

1) Static Classification of Tags: We evaluate the Pirro'-Seco similarity metric on the aforementioned 914 Flickr tags [26]. The pair-wise semantic similarity is a numerical value between 0 and 1, with more similar words giving higher score. Our analysis is as follows.

1) For each tag τ_i and each topic γ_j, compute their semantic similarity $sim(\tau_i, \gamma_j)$ using Pirro'-Seco metric.
2) For each tag τ_i, sort the values $sim(\tau_i, \gamma_j)$ for all j from high to low; select the top three highest ranking topics and denote them as the set $\Omega = \{\gamma^1, \gamma^2, \gamma^3\}$.
3) For each tag τ_i, a human judge evaluates the following:
   a) Semantic similarity of τ_i and the topics in Ω: If Ω contains at least one topic semantically similar to τ_i, then the human judge sets variable $counter1_i$ to 1, otherwise 0.
   b) How well topics are selected: If $counter1_i = 0$ and Γ contains at least one topic semantically similar to τ_i, then the human judge sets variable $counter2_i$ to 1, otherwise 0.

Then, we compute the sums $C_1 = \sum_i counter1_i$, and $C_2 = \sum_i counter2_i$, respectively in Table IV. If $counter1_i = 1$ or $counter2_i = 1$ for all i's, then each of the tags can find at least one topic that is semantically similar. For tag τ_i with nonzero $counter2_i$, value $counter1_i$ represents how well the semantic similarity measure is in finding the most similar topic(s).

Table IV shows that classification correctly identifies the most suitable topic among the top-3 hits for 53% of tags. However, for 31% of the tags studied, none of three most similar topics returned by the Pirro'-Seco algorithm are considered related by the human judge. We also evaluate the tags using Jiang-Conrath [13], Resnik [28], and Lin [19] metrics, which do not provide significantly better results. We do not report the analysis results here. In tag-topic classification, the assignment is computed based on a *single similarity value* between the tag and the topic, which may not be accurate for certain words. In addition, static and arbitrary choice of topics limits the accuracy of finding the suitable topic for a given tag.

Static Tag Classification		Dynamic Tag Clustering	
C_1	C_2	C_X	C_Y
496	278	564	252

TABLE IV

COUNTER VALUES IN OUR SEMANTIC SIMILARITY ANALYSIS PERFORMED BY A HUMAN JUDGE FOR BOTH TAG CLASSIFICATION AND TAG CLUSTERING METHODS. THE ANALYSIS IS DONE ON A TOTAL OF 914 TAGS RETRIEVED FROM FLICKR.

2) Dynamic Classification of Tags: To analyze the clustering quality, we let the same human judge (as in the previous section) to manually look into each tag and count the number of tags that are semantically related to their cluster centers. The human judge reports the following two counters, $counterX$ and $counterY$, which are defined as follows. For each tag τ_i in a cluster c_j with center τ_{c_j}, if τ_i is semantically related to the cluster center τ_{c_j}, then $counterX_i = 1$, otherwise, 0. If $counterX_i = 0$ and there exists at least one cluster center (among the rest of 49 centers) that is semantically related to the tag τ_i, then $counterY_i = 1$. Then, we compute $C_X = \sum_{i=1}^{n} counterX_i$ and $C_Y = \sum_{i=1}^{n} counterY_i$. We compare the performance of clustering method with the static analysis in Table IV.

Table IV shows that dynamic tag clustering gives better results than the static tag-topic classification. Both C_1 and C_X represent the number of correctly assigned tags (either into a topic or into a cluster of tags). Out of 914 tags, the human judge finds 68 more tags (8% more) that are properly assigned by clustering than by classification. Counters C_2 and C_Y represent the number of tags that are mis-assigned while there exists a different topic or a cluster to which the tag should belong. Clustering gives us 26 fewer such misclassification cases. We plan to extend this analysis to a larger scale in future.

ACKNOWLEDGMENTS

We thank Brian Thompson for sharing his clustering code.

REFERENCES

[1] A. Acquisti and R. Gross. Imagined communities: Awareness, Information Sharing, and Privacy on the Facebook. In *In Proc. of Privacy Enhancing Technologies*, pages 36–58, 2006.

[2] A. Acquisti and J. Grossklags. Privacy and rationality in decision making. *IEEE Security and Privacy (Jan/Feb)*, pages 26–33, 2005.

[3] S. Banerjee and T. Pedersen. Extended gloss overlaps as a measure of semantic relatedness. In *In Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 805–810, 2003.

[4] D. Beaver. 10 billion photos, October 2008. `http://www.facebook.com/note.php?note\_id=30695603919`.

[5] M. Blaze, J. Feigenbaum, and A. D. Keromytis. KeyNote: Trust management for public-key infrastructures. In *Proceedings of Security Protocols International Workshop*, 1998.

[6] B. Carminati, E. Ferrari, and A. Perego. Rule-based access control for social networks. In R. Meersman, Z. Tari, and P. Herrero, editors, *OTM Workshops (2)*, volume 4278 of *Lecture Notes in Computer Science*, pages 1734–1744. Springer, 2006.

[7] C. Finkbeiner. Estimation of the multiple factor model when data are missing. pages 409–420, 1979.

[8] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In M. M. Veloso, editor, *IJCAI*, pages 1606–1611, 2007.

[9] C. Gates. Access control requirements for Web 2.0 Security and Privacy. In *IEEE Web 2.0 Privacy and Security Workshop*, 2007.

[10] K. K. Gollu, S. Saroiu, and A. Wolman. A social networking-based access control scheme for personal content. In *Proceedings of the 21st ACM Symposium on Operating Systems Principles (SOSP '07), Work-in-Progress Session*, 2007.

[11] M. Hart, R. Johnson, and A. Stent. More content - less control: Access control in the web 2.0. In *In Proc. of Web 2.0 Security and Privacy (in conjunction with IEEE Symposium on Security and Privacy)*, 2007.

[12] J. Huai, R. Chen, H.-W. Hon, Y. Liu, W.-Y. Ma, A. Tomkins, and X. Zhang, editors. *Proceedings of the 17th International Conference on World Wide Web, WWW 2008, Beijing, China, April 21-25, 2008*. ACM, 2008.

[13] J. Jiang and D. Conrath. In *Proceedings of ROCLING X*, Semantic similarity based on corpus statistics and lexical taxonomy.

[14] D. Jiménez, E. Ferretti, V. Vidal, P. Rosso, and C. F. Enguix. The influence of semantics in IR using LSI and K-means clustering techniques. In *Proc. of Workshop on Conceptual Information Retrieval and Clustering of Documents*, pages 286–291, 2003.

[15] A. Kulkarni and T. Pedersen. Senseclusters: Unsupervised clustering and labeling of similar contexts. In *ACL*. The Association for Computer Linguistics, 2005.

[16] K. Lerman, A. Plangprasopchok, and C. Wong. Personalizing image search results on flickr. *CoRR*, abs/0704.1676, 2007.

[17] X. Li, L. Guo, and Y. E. Zhao. Tag-based social interest discovery. In Huai et al. [12], pages 675–684.

[18] Y. Li, D. McLean, Z. A. Bandar, J. D. O'Shea, and K. Crockett. Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge and Data Engineering*, 18(8):1138 – 1150, 2006.

[19] D. Lin. An information-theoretic definition of similarity. In *Proceedings of Conference on Machine Learning*, page 296304, 1998.

[20] M. Mannan and P. C. van Oorschot. Privacy-enhanced sharing of personal content on the web. In Huai et al. [12], pages 487–496.

[21] A. Mathes. Folksonomies: cooperative classification and communication through shared metadata, 2004.

[22] A. W. Moore and D. Pelleg. X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the Seventeenth International Conference on Machine Learning*, page 727734. Morgan Kaufmann, 2000.

[23] E. J. S. Norm O'Rourke, Larry Hatcher. *A step-by-step approach to using SAS for univariate and multivariate statistics*. SAS.

[24] T. Pedersen. WordNet::Similarity. `http://www.d.umn.edu/~tpederse/similarity.html`.

[25] T. Pedersen and A. Kulkarni. Selecting the "right" number of senses based on clustering criterion functions. In *EACL*. The Association for Computer Linguistics, 2006.

[26] G. Pirro' and N. Seco. Design, implementation and evaluation of a new semantic similarity metric combining features and intrinsic information content. In *Proc. of On the Move to Meaningful Internet Systems*, 2008.

[27] A. Plangprasopchok and K. Lerman. Exploiting social annotation for automatic resource discovery. *CoRR*, abs/0704.1675, 2007.

[28] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *In Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 448–453, 1995.

[29] D. Rosenblum. What anyone can know: The privacy risks of social networking sites. *IEEE Security and Privacy*, 5(3):40–49, 2007.

[30] R. Tamassia, D. Yao, and W. H. Winsborough. Role-based cascaded delegation. In *Proceedings of the ACM Symposium on Access Control Models and Technologies (SACMAT '04)*, pages 146 – 155. ACM Press, June 2004.

[31] M. Vilares, F. J. Ribadas, and J. Vilares. Phrase Similarity through the Edit Distance. In *Database and Expert Systems Applications*, volume 3180 of *Lecture Notes in Computer Science*, pages 306–317. Springer, 2004.

[32] Wordnet - a lexical database for the English language. `http://wordnet.princeton.edu/`.

[33] X. Wu, L. Zhang, and Y. Yu. Exploring social annotations for the semantic web. In *WWW*, pages 417–426, 2006.

[34] E. Zheleva and L. Getoor. To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 531–540, New York, NY, USA, 2009. ACM.