Preventing Automatic User Profiling in Web 2.0 Applications

Alexandre Viejo, David Sánchez[1], Jordi Castellà-Roca

Departament d'Enginyeria Informàtica i Matemàtiques,
UNESCO Chair in Data Privacy, Universitat Rovira i Virgili,
Av. Països Catalans 26, E-43007 Tarragona, Spain

Abstract

The rise of the Internet and Web 2.0 platforms have brought very accessible publishing techniques that have transformed users' role from mere *content consumers* to fully *content consumers-producers*. Previous works have shown that user-generated content can be automatically analyzed to extract useful information for the society. Nevertheless, researchers have also shown that it is possible to build individual user profiles automatically. This situation may provoke concerns to the users worried about their privacy. In this paper, we present a new scheme that effectively obfuscates the real user's profile in front of automatic profiling systems, while maintaining her publications intact in order to interfere the least with her readers. The proposed system generates and publishes fake messages with terms semantically correlated with user posts to distort and, hence, hide the real profile. Our method has been tested using Twitter, a very well-known Web 2.0 microblogging platform. Evaluation results show that this new scheme effectively distorts user profiles, producing uniform (i.e. balanced) profiles that hardly characterize users and outperforming simpler methods based on random distortions. In addition to that, the presented system is adaptive, capable of profiling and anonymizing users with a quite limited number of publications and it reacts quickly to any variation in their interests.

Keywords: Profiling, Privacy, Knowledge-based system, Web 2.0.

[1] Corresponding author. Address: Departament d'Enginyeria Informàtica i Matemàtiques. Universitat Rovira i Virgili. Avda. Paisos Catalans, 26. 43007. Tarragona. Spain. Tel.:+034 977 556563; Fax: +034 977 559710. E-mail: david.sanchez@urv.cat

1. Introduction

The Internet is continuously evolving and creating new possibilities of interaction for its users. Nowadays, the rise of the Web 2.0 and its related technologies have transformed *passive users* into *active* ones. The term Web 2.0 is associated with web applications (*e.g.*; social networking sites, blogs, wikis, video sharing sites, etc) that allow users to interact and collaborate between them in a social dialogue using highly accessible publishing techniques. These new technologies enable users to both generate and consume content. As a result, they have evolved from *content consumers* to *content consumers-producers*.

From the different publishing methods which are offered in any Web 2.0 application, this paper focuses on the text-based posts. This service is known as *blogging* when there is no limitation in the length of the posts or *microblogging* when they are limited to a certain (and usually small) number of characters.

Users publish text-based posts containing opinions and information about a broad range of topics. These topics can be general (*e.g.,* films, music, sports, etc) but also personal (*e.g.,* current activity, current localization, current feeling, etc). Regarding what motivates people to share information on the Internet, (Rui and Whinston, 2011) states that people contribute in the Web to seek attention from others. By getting attention, they obtain publicity, vanity or ego gratification from peer recognition.

Researchers have shown that the information shared by the users of Web 2.0 technologies can be automatically analyzed and relevant data can be retrieved. For example, user-generated content has been used to forecast flu trends (Achrekar et al., 2011), develop earthquake warning systems (Sakaki et al., 2010), compile product recommendations (Garcia Esparza et al., 2012), build interest-based recommendation systems (Nocera and Ursino, 2011; Li et al., 2012) or help business to predict future sales (Rui et al., 2009) among others.

Tools which are used to automatically analyze user-generated content can extract useful knowledge in an *aggregated* way or in an *individual* way. The former groups the information gathered from several users and, hence, it weakens the link between the data and the person who has generated it. On the contrary, the latter approach builds a *complete profile* of each analyzed user, explicitly linking all the user characterization with her identity.

Since some of the topics included in the gathered data may refer to *personal* data, user profiling clearly puts at risk the privacy of the users of the Web 2.0 (Islam and Brankovic, 2011). More specifically, the *Consumer Reports'2010 State of the Net analysis* (Consumer Reports National Research Center, 2010) states that more than half of users of social networks and similar applications share private information about themselves online. As explained in Zhang et al. (2010), user characterization and leakage of personal data may invite malicious attacks from the cyberspace (*e.g.*; personalized spamming, phishing, etc) and even from the real world.

In order to alleviate these problems, Web 2.0 applications offer *privacy settings* to allow users to control who has access to certain contents generated by them. However, this basic privacy-preserving tool suffers from the following problems:

- These privacy settings are generally not sufficiently understood by the average user who seldom changes the default configuration which is provided by the company that owns the web application (Van Eecke and Truyens, 2010). To make it worse, these companies offer privacy configurations that make most of their information public by default and they require the users to make a choice if they wish to keep information private (Bilton, 2010).

- A recent study from Barracuda Labs (a security company) states that 30% per cent of Twitter users are worried about how this company protects their privacy (Wilson, 2011). Specifically, Twitter offers access to its archive of billions of posted user messages dating back to January 2010 (Brown, 2012; Crimes, 2012; Hotonline.net, 2012) and it has been reported that individuals are concerned about their data being exploited by advertisers in order to target them (Crimes, 2012). It can be assumed that these privacy concerns can be extended to other similar Web 2.0 platforms.

- A privacy-preserving method that is based on restricting the visibility of the user-generated content compromises as well the capability of the users to gain attention from others. This situation can represent a strong limitation to the use of those privacy settings because, as explained above, seeking attention is the main motivation of the users of Web 2.0 applications.

As a result, users who are willing to decrease the chances of privacy breaches should use privacy-preserving mechanisms deployed and managed by themselves to prevent data extraction techniques from profiling them in an accurate way. In addition to that, these methods must protect the privacy without limiting the visibility of the user-generated data.

1.1. Contribution and plan of this paper

Text-based posts published on Web 2.0 sites can contain relevant and useful information for the society. Therefore, developing text-based data extraction methods for gathering this data in an effective way is an important field of research. However, those techniques can be also used to automatically profile content generators and jeopardize their privacy. As a consequence, measures to prevent automatic user profiling in Web 2.0 applications should be proposed.

In this paper, we offer three contributions to this research field:

1. We propose a new knowledge-based profiling approach grounded in the Information Theory that dynamically quantifies the amount of information provided by terms contained in the text messages published by users to accurately characterize their profile according to a set of categories.
2. We present an adaptive method that, according to the characterized profile, distorts it to prevent automatic data analysis techniques from profiling the user, while maintaining her published data intact. Current profilers mainly characterize users according to the distribution of terms appearing in their messages. Those general profilers cannot detect the fake queries generated by our proposal.
3. We use Twitter (the leading social network based on text-based messages (McMillan, 2011)) to test both the profiling and profile obfuscation proposals. Results show that our profiler characterizes users faster and more accurately than methods based solely on absolute term frequencies, whereas our profile distortion method effectively balances user profiles in front of automatic profilers based on information distribution.

Section 2 introduces the state of the art related to text-based profiling techniques used in the Web 2.0 environment; it also reviews privacy-preserving approaches found in the field. Section 3 details our new proposals.

Section 4 evaluates the accuracy of our profiling method and the obfuscation level provided by our masking scheme. Finally, Section 5 reports some concluding remarks.

2. State of the art

As explained above, this section first covers automatic profiling techniques applied in Web 2.0 applications. Then, it reviews methods which can be used to prevent automatic user profiling from textual data.

2.1. Text-based automatic profiling methods

In the work presented by Ebner et al. (2010), the authors try to categorize different users in an unsupervised manner according to the overlapping keywords found in their published messages. In order to extract the keywords of the messages, the *Yahoo Term Extraction Web Service* is suggested. The authors test this approach using the messages which were published on Twitter by the attendants to a Conference. The results of their evaluations show that user posts present high diversity and nearly no overlapping keywords that prevents from achieving an accurate profiling. Authors then argue that a knowledge-based semantic analysis is needed to deal with the high keyword diversity. Authors propose to manually linking each keyword with its related category.

Zoltan and Johann (2011) present a knowledge-based framework that builds user profiles from text messages shared in social platforms. To do so, authors extract Named Entities and keywords and match them to categories found in a knowledge base. Specifically, they exploit Linked Data vocabularies (such as DBpedia (DBpedia, 2012)) as knowledge base. Authors leverage the contribution of extracted information to the user profile according to their degree of occurrence (*i.e.,* TF-IDF (Salton and Buckley, 1988)) with respect to the linked categories. As a result, user profiles are characterized according to a set of weighted categories.

Michelson and Macskassy (2010) present a similar approach, discovering users' topics of interest by examining the Named Entities they mention in their posts in Twitter. First, the entities in each message are found by using simple capitalization heuristics. The author argues that this can be challenging because tweets are generally ungrammatical and noisy. Then, each entity is disambiguated and categorized using Wikipedia as a knowledge base. The process is as follows: the terms contained in the publication are considered to be the context for that entity. Then, this context is compared to the text of each candidate entity's page from Wikipedia. The entity from Wikipedia which have more term occurrences is selected. Finally, the tree of Wikipedia categories related to the selected entity is retrieved. Due to the complexity of Wikipedia category trees, in the last step, the proposed scheme filters them by selecting the categories (nodes in the trees) that occur frequently to generate useful topic profiles.

Bernstein et al. (2010) discuss the problems of relying on term occurrence and co-occurrence to identify topics of messages published in a social network like Twitter. They argue that the current best practices for topic identification assume that user posts are of a decent length. Since, messages in Twitter (and other Web 2.0 applications) are at most 140 characters long, the authors explain that this assumption fails almost by definition. Additionally, users usually compress similar words to gain space in order to insert their opinions. To tackle these problems, Bernstein et al. (2010) present a novel approach based on transforming noun phrases found in each user message in a set of web search queries. These queries are executed in search engines (*e.g.,* Google, Bing, etc) to retrieve documents that help to expand the context of the original message. Then, authors apply term co-occurrence techniques to assign the most likely topic for the original published message. This proposal is interesting but suffers from two main problems: (i) the use of search engines introduces a serious overhead to the scheme in terms of execution time; and (ii) it does not consider ambiguity issues derived from querying keyword-base search engines.

The proposal presented by Abel et al. (2011) also notes the necessity of expanding the context of user posts to properly categorize them. In this case, the authors expanded posts by linking messages published in Twitter with related news articles. Following the same idea that Bernstein et al. (2010), this proposal also uses methods based on term occurrences to classify each message and associate topics.

As a conclusion, most methods rely on a knowledge base (*e.g.,* Wikipedia or DBPedia, news repositories, the Web) that bring the semantic background needed to properly analyze and characterize user profile/posts. All authors rely on term occurrences/co-occurrences to leverage their contribution to the profile. As it will be discussed in section 3, methods solely based on quantifying term appearances consider them as equally important to the profile. However, it is coherent to consider some terms (*e.g.,* iPhone) as more informative than others (*e.g.,* cell-phone) when characterizing the user profile due to their different degrees of specificity. Likewise, some potential profiling categories (*e.g.,* cell-phones) may be more specific than others (*e.g.,* electronics). To properly capture these dimensions that are implicitly considered in human reasonings (Resnik, 1995), our knowledge-based profiling method will exploit the Information Theory and the concept of Information Content (IC) to accurately quantify extracted evidences and to better characterize the user profile. According to (Ross, 1976), the IC of a term is computed as the inverse of its probability of appearance in a corpus. This allows to quantify the amount of information that a certain concept provides when it appears in a certain context.

2.2. Privacy-preserving methods

In the literature, there are some mechanisms which try to prevent the automatic profiling of users of Web 2.0 applications. This section focuses on those schemes which can be deployed and managed by the users themselves without the interaction of the company that supports the Web 2.0 application.

A straightforward solution to prevent automatic profilers from accessing the user data which can be found in a Web 2.0 application is using cryptography primitives to cipher the text before publishing it. Applying this method, only the individuals with the correct cryptographic keys will be able to access the protected content. Nevertheless, this solution is quite problematic: usually, registering on well-know social networks under a pseudonym, or obfuscating the personal information in any way is forbidden by the terms of service. In this way, Facebook has banned users in cases where it identified violations of those terms (Scoble, 2008). According to that, instead of being encrypted, data should be *distorted*. This is, the information to be published should look real but it should not be completely correct (or correct at all).

Luo et al. (2009) present a Firefox extension that allows users to specify which data or activity need to be kept private. The sensitive data is substituted with fake one, while the real data is stored in a third party server that can be only accessed by the allowed users. One of the main shortcomings of this scheme is that it relies on a centralized infrastructure which must be honest and always available.

A similar proposal is presented by Guha et al. (2008). This scheme first divides the private data into atoms and then it replaces each atom with a corresponding atom from another randomly selected user who uses the same scheme. Some shortcomings of this proposal are: (i) it requires a certain number of users to provide anonymity; (ii) it requires some infrastructure that keeps the relations between the users and the atoms; and (iii) it has been designed to protect personal data like gender, age, etc. Some work is required in order to use it to protect text-based messages.

Both proposals above use complete distortion of the published information to protect it from profilers and their use jeopardizes the capability of the users to gain attention from others. As stated previously, getting attention (which in turn provides publicity, vanity and ego gratification) is the main motivation for Web 2.0 users (Rui and Whinston, 2011), hence, users might not be willing to sacrifice their level of attention for preserving their privacy.

A slightly different approach would be to partially distort the data to achieve a certain level of homogeneity, making difficult the profile characterization. Then, the distortion level introduced can be fixed according to a trade-off between the privacy level achieved and the reliability of the published information. There are some works in the literature that propose the use of similar techniques in Web 2.0 applications but only on those which are based on localization data (Freni et al., 2010) or friends topology data (Hay et al., 2008). However, methods for location-based data are not directly applicable to text-based content generated by the users themselves. For this specific purpose, authors dealing with document anonymization (Abril et al., 2011) propose three different techniques which can be applied to text-based sources:

- *Named entity generalization.* Entities can be generalized (*e.g.,* iPhone → cell phone) to achieve some degree of privacy while preserving some of their semantic meaning in the document. Thanks to this behavior, this technique does not introduce erroneous information. This approach is usually used in location-based services like Freni et al. (2010).

- *Entity swapping.* This method is based on swapping relatively similar entities between documents of the same set, or within the same document depending on the concrete case. Note that this technique may inject erroneous information.

- *Entity noise addition.* This method is based on introducing new entities to user documents that may aid to hide the original information. Similarly to the latter technique, this one may also introduce erroneous information.

The main problem of the first method is that it introduces an information loss derived from the degree of generalization introduced in user data. Regarding the *entity swapping* technique, this approach suffers from two drawbacks when trying to apply it to Web 2.0 messages: (i) it is designed to work with documents of a decent length and properly structured. For example, messages published in Twitter do not follow these requirements; and (ii) Web 2.0 users may not be comfortable with certain messages generated and published by a scheme that uses this approach. Note that the content-producer user will be, from the point of view of any independent reader, the legitimate author of the published message. Finally, if applied directly in Web 2.0 applications (like Twitter), the *entity noise addition* method may also generate uncomfortable publications but this approach might be successful if it is implemented allowing independent readers to distinguish between legitimate and distorted publications. As it will be described in the next section, this is precisely the premise of our privacy-preserving method.

3. Proposed method

The main goal pursued by our method is to hide the user profile inherent to her published messages, so that an automatic profiling system could gain no knowledge about her preferences. Considering the analysis of related works conducted in the previous section, our method is based on the following premises.

First, automatic profiling systems (discussed in Section 2.1) mainly characterize users according to the distribution of terms appearing in their messages. In consequence, it could be possible to prevent automatic user profiling by altering the term distribution of published messages, so that profiling topics become so homogeneously distributed that none of them seems to prevail.

Human readers, on the contrary, interpret user messages according to their semantics, and they expect that human-constructed messages would be semantically coherent. To modify term distribution so that it interferes the least with user's readers, this should be done in a way that human beings can easily distinguish fake/altered terms/posts from original ones. Since the the semantic interpretation of messages is what distinguishes human readers from automatic profilers, we propose maintaining user messages *intact* (so that their semantics are not altered), while *adding* fake messages with specially tailored contents that would help alter term distribution towards an uniformly distributed profile. Fake messages will be constructed as a concatenation of terms, lacking semantically-coherent discourse. Thanks to this lack of coherency, human readers would be able to intuitively discern between user messages and fake ones. On the other hand, profilers based on counting term occurrences/co-occurrences would not be able distinguish original and fake terms, so that all of them will equally contribute to characterize the user. As discussed in section Section 2.2, *adding* new information to hide user profile is preferable to *modify* her posts, a measure that could mislead human readers and, hence, negatively affect the user.

To create this profile-balancing fake messages, our system automatically and unsupervisedly learns the user profile following similar principles as those of related works (Zoltan and Johann, 2011; Michelson and Macskassy, 2010; Abel et al., 2011) but with a more accurate characterization of profile categories.

As stated in Section 2.2, the strategy of adding new information to hide sensible data (the user profile in our case) can be seen as adding *Noise* to original data (Brand, 2002). Numerical additive noise is based on distributional statistics of data (*e.g.,* average, standard deviation), whereas when dealing with textual data, some authors refer to the fact of adding new terms as *Semantic Noise (SN)* (Abril et al., 2011). In our case, noise (*i.e.,* fake terms/messages) is semantically *correlated* with respect to the original data so that the user profile can be balanced and, hence, hidden.

Our method consists of two modules which behave complementarily to dynamically hide user profile as new messages are added. The first one is the *Profiler*, which is in charge of dynamically learning the user profile according to uploaded messages. The second one is the *Noise Generator*, which considers the learned profile to adaptively introduce semantically correlated messages (*i.e.,* SN). In this section, these two modules are described in detail.

3.1. Profiler

The *profiler* is a knowledge-based module that aims to mimic the common way to build a user profile according to published messages. Inspired by systems like TweetPsych (2012) and Peerindex (2012), it considers a set of well-defined categories $C = \{c_1, \ldots, c_k\}$ (*e.g.,* science, health, society, sports, etc) and analyzes user posts to extract evidences that may be classified in one of these categories. In this manner, a user profile is characterized by the amount of evidences retrieved for each category.

As a contribution, and on the contrary to approaches which base the profile characterization solely on term occurrences (Ebner et al., 2010; Zoltan and Johann, 2011; Abel et al., 2011), we ground the profile construction in the Information Theory in order to better quantify the contribution of extracted evidences according to the *amount of information* they provide (Ross, 1976).

Three consecutive steps are performed: message analysis, semantic classification and user profile characterization.

3.1.1. Message analysis

The *profiler* receives a message m as input to be analyzed. As discussed in Section 2, in many environments like microblogging sites, messages are usually slightly-grammatical short texts which are difficult to syntactically and semantically analyze (Ebner et al., 2010; Zoltan and Johann, 2011; Bernstein et al., 2010; Abel et al., 2011). Hence, we opted, as done in some related works (Zoltan and Johann, 2011; Michelson and Macskassy, 2010; Bernstein et al., 2010), to implement a shallow linguistic parsing that, instead of trying to analyze well-formed sentences, focuses on extracting pieces of text with semantic content that can contribute to characterize the user profile: *noun-phrases (NPs)*. NPs are built around a noun whose semantics can be refined by adding new nouns or adjectives (*e.g., iPhone* → *new iPhone*). Each NP either refers to a generic concept (*e.g., water sports*) or it could be considered as a proper noun that is an instance of a concept (*e.g., iPhone* is an instance of a *cell-phone*). The latter ones are usually considered as *Named-Entities (NEs)*, and some authors (Michelson and Macskassy, 2010; Sánchez et al., 2011b) use them as concrete and less ambiguous evidences from which the profile can be characterized. In our work, considering the potentially short contexts analyzed (*e.g.,* tweets from a Twitter account) and the need of characterizing the profile as soon as possible to start introducing SN, we consider all NPs (and NEs in particular) to improve the throughput (*i.e.,* amount of extracted evidences) of the message analysis.

The analysis of user posts relies on several natural language processing tools[2] *sentence detection*, *tokenization*, *part-of-speech tagging* and *syntactic parsing* (*i.e.,* chunking) so that verbal, prepositional or nominal phrases are detected. From these, only NPs are extracted.

As output of this analysis, from the input message m, we obtain the set $M = \{< NP_1, w_1 >, \ldots, < NP_p, w_p >\}$ in which NP_i is each Noun Phrase extracted from m and w_i is its number of appearances. This set will

[2] OpenNLP Maxent Package: `http://maxent.sourceforge.net/about.html`

be the structured input for the profile characterization step, in which both the term frequency (TF) and their contribution according to the amount of information provided estimated from large corpora (which, as discussed below, is similar to the notion of IDF) will be considered.

3.1.2. Semantic classification

The next step consists in semantically analyzing extracted NPs in order to classify them, if possible, in one of the profile categories (in C). By doing this, the number of NPs corresponding to each category can be evaluated to characterize the user profile in a later stage.

To enable this classification from a semantic point of view (*i.e.,* to associate each NP to its conceptual abstraction), we rely on a predefined knowledge base. This knowledge base can be a taxonomy, folksonomy or a more formal ontology (Guarino, 1998) as long as it offers a structured conceptualization of one or several knowledge domains expressed by, at least taxonomic relationships. To improve the recall of the semantic analysis and due to the proliferation of proper nouns in user posts, a large base that potentially includes up-to-date NEs is desirable. As stated in Section 2, several related works (Zoltan and Johann, 2011; Michelson and Macskassy, 2010) rely on the *Wikipedia/DBPedia category structure* as the taxonomy used to classify extracted evidences. In our work we rely on the *Open Directory Project (ODP)* hierarchy of categories. ODP is the largest, most comprehensive human-edited directory of the Web[3], constructed and maintained by a vast community of volunteer editors. The purpose of ODP is to list and categorize web sites. Manually created categories are taxonomically structured and populated with related web resources. Nowadays, it classifies almost 5 million web sites in more than 1 million categories (considering also up-to-date Named Entities).

We preferred ODP to Wikipedia/DBPedia since the former offers comprehensive and well-structured hierarchical classifications, whereas the later typically result in cyclic taxonomical relationships and excessively complex taxonomical trees that should be cut or pruned to keep it manageable (Michelson and Macskassy, 2010). Both of them offer a large coverage of NEs (on the contrary to more formal repositories like WordNet (Fellbaum, 1998), which focus only on general concepts) and they can be downloaded and queried

[3] Open Directory Project: `http://www.dmoz.org/docs/en/about.html`

locally, avoiding the overhead of on-line accessing.

To semantically classify NPs in M, we query each NP_i to ODP. ODP matches the queried expression with all its categories, returning the hierarchy of categories in which NP_i is included (*e.g.,* if we query the NP *"iPhone"*, it will return: *iPhone* → *Smartphones* → *Handhelds* → *Systems* → *Computers*). To improve the retrieval recall, ODP applies stemming (to detect equivalent morphological constructions, *e.g., "cell phones"*=*"cell phone"*) and omits punctuation marks. Note also that, since ODP categories are created by heterogeneous users, almost equivalent or even synonym categories have been defined; this also helps to improve the retrieval recall. From the returned hierarchy, we pick up the hierarchy H_i ($H_i = h_{i,1} \rightarrow \ldots \rightarrow h_{i,l}$) in which the lowest node is the queried NP_i and the most general one corresponds to one of the general categories in ODP (*e.g.,* computers, arts, shopping, society, science, sports, etc.). If NP_i is not found "as is" in ODP, we iteratively try with simpler forms of the NP by removing its adjectives/nouns, starting from the word localed most on the left (*e.g., "new fancy iPhone"* → *"fancy iPhone"* → *"iPhone"*). This will improve the recall while maintaining the core semantics. The fact that NPs incorporate qualifiers is quite common in texts but these are hardly covered in knowledge structures which try to model them in a generic way.

3.1.3. Profile characterization

In this last stage, categories to which extracted NPs belong are taken into consideration to characterize the user profile. Similar to systems like Zoltan and Johann (2011); Abel et al. (2011); TweetPsych (2012); Peerindex (2012), a user profile, Π, consists of a weighted category set $\Pi = \{< c_1, v_1 >, \ldots, < c_k, v_k >\}$, where v_j states the sum of *information* provided by the terms (NPs) retrieved and classified for the j-th category.

First, for each NP_i we look if any of the profile categories (*e.g.,* c_j in C) is a taxonomical subsumer of it (*i.e.,* if c_j is included in H_i). In the affirmative case, we state that NP_i *is-a* c_j (*i.e.,* NP_i is a taxonomical specialization of c_j).

In the affirmative case, we compute and add the contribution of NP_i to c_j, following the same principles as the well-known TF-IDF (Salton and Buckley, 1988) score. First, we consider the number of repetitions/term frequency (TF), w_i, of NP_i; that is, the larger the amount of times NP_i appears in user messages, the higher its contribution to the user profile will be. Moreover, instead of considering a constant contribution for all equally frequent terms to the corresponding category, we normalize it according to the *amount of information* they provide. In this manner, NPs referring to general and abstract terms will have a lower influence in the profile characterization than other ones referring to very specific ones. The amount of information is computed as the Information Content (IC) of NP_i.

In general, the IC of a concept is computed as the inverse of its probability p of appearance in a corpus (Ross, 1976), quantifying the amount of information provided when appearing in a context (in our case, the user message). In this manner, infrequent terms are considered as more informative than common ones and, hence, the former ones will contribute more than the later to the corresponding category. Applied to NP_i, its IC is computed as follows:

$$IC(NP_i) = -\log p(NP_i)$$

Note that the $IC(NP_i)$ represents, in essence, the same dimension as the Inverse Document Frequency (IDF) of NP_i, which is also used by related works (Zoltan and Johann, 2011; Bernstein et al., 2010; Abel et al., 2011) to normalize the contribution of terms according to their relative distribution in a corpus. To gather robust IC values, appearance probabilities should be computed from large, heterogeneous and ideally disambiguated corpora (Resnik, 1995; Sánchez and Batet, 2011; Sánchez et al., 2011a) so that they approximate, in an accurate way, the *real* distribution of data at a social scale. Nowadays, the largest electronic repository is the Web. Compared to the more reduced copora used by related works to compute IDF, the Web offers an scope several orders of magnitude higher, which is beneficial to obtain robust statistics and IC values. Several authors have used web search engines to gather web-scale statistics about term occurrences according to the search engine hit count (Turney, 2001; Sánchez, 2010; Sánchez and Moreno, 2008; Cilibrasi and Vitányi, 2006). However, they suffered from the ambiguity caused by the fact that keyword-based search engine only look for the presence or absence of the searched query, regardless its semantics (Sánchez et al., 2010, 2011a). To minimize this problem, we rely again on ODP. Since ODP indexes millions of web resources which have been manually categorized (*i.e.,* disambiguated) according to their topics, it provides accurate yet representative hit count results.

In this manner, being hit_count_{ODP} the number of web sites returned by ODP for a given query, we estimate the probability of appearance of NP_i

in ODP (where *total_webs* is total amount of indexed web sites, around 5 millions) as:

$$p(NP_i) = \frac{hit\_count_{ODP}(NP_i)}{total\_webs}$$

Then, the contribution, φ, of NP_i to its corresponding profile category (c_j) is computed as the product between the number of times it appeared in user posts (w_i) and the amount of information it provides ($IC(NP_i)$), as follows:

$$\varphi(NP_i) = w_i \times IC(NP_i) = w_i \times -\log \frac{hit\_count_{ODP}(NP_i)}{total\_webs}$$

By taking w_i into consideration, we increase the contribution of terms frequently mentioned in user messages during the process of profile characterization. Moreover, by considering $IC(NP_i)$ (which is the non-linear inverse to $hitCount_{ODP}(NP_i)$), we penalize usual terms (those with a large $hitCount_{ODP}(NP_i)$) since, due to their commonness, they are more likely to appear in a discourse.

Finally, φ values for all NPs that are taxonomic specializations of c_j are added to the category weight (v_j):

$$v_j = \sum_{\forall NP_i \ is-a \ c_j} \varphi(NP_i)$$

Once all NPs are considered, the user profile, Π, corresponding to the analyzed message/messages is defined by a ranked list of categories, according to their weights.

Analyzing the profile Π from the Information Theory perspective, profile categories may also present different degrees of generality (*i.e.,* Business is more general than Shopping). This means that, in a uniformly distributed text, the chance to extract a NP belonging to a more concrete category would be lower than for a more general one (because the latter cover more terms). To consider this potential differences in the profile, we rely on the IC of profile categories (*i.e.,* $IC(c_j)$). For each profile category c_j, its $IC(c_j)$ is used to normalize its weight v_j, as follows:

$$v'_j = v_j \times IC(c_j)$$

In this manner, general categories would require from more numerous (*i.e.,* higher amount of occurrences) and more concrete (*i.e.,* higher IC values) extractions than more specific ones. As a result of this process, we obtain the normalized profile (Π') consisting of a set of normalized category weights $\Pi' = \{< c_1, v'_1 >, \ldots, < c_k, v'_k >\}$ that numerically quantifies the preferences of the user according to the analyzed messages.

Note that the final weights for each category are obtained from the IC of both terms and categories, which are computed according to their distribution in a large information source (ODP, with more than 5 million web resources). Since term occurrences have been normalized according to their degree of generality/concreteness in a large and representative community (thousands of heterogeneous and independent contributors that try to objectively model and classify information sources in ODP), they capture, up to a degree, the *social* distribution of information (Cilibrasi and Vitányi, 2006). This is desirable both from the profiling and profile distortion perspectives, since it will enable more natural and coherent results.

3.2. Noise generator

Given a message or a set of messages posted by a user for which a profile has been characterized, the objective of the *noise generator* module is introducing additional terms as new *fake messages* that would balance the user profile towards a uniformly distributed one (according to the considered categories) while maintaining the original tweets unaltered. For example, given a user posting about *"computers"* (so that his profile will show a prevalence on this category), the *noise generator* will add new messages with terms belonging to other categories (such as *"health"*, *"sports"* or *"society"*, if those are categories considered during the profiling process) so that the normalized weights associated to all categories (*"computers"* but also *"health"*, *"sports"* and *"society"*) would tend to balance, approximating to the most prevalent category.

The *noise generator* performs two steps: profile analysis and fake message construction.

3.2.1. Profile analysis

Given a user profile Π' constructed by the *profiler* as described in the previous section, the *noise generator* analyzes the set of weighted categories and selects the one with the maximum normalized weight (*i.e.,* $< c_{MAX}, v'_{MAX} >= argmax_{\forall < c_i, v'_i > \in \Pi}(v'_i)$). Then, for the rest of the categories c_j in Π', it computes the difference with respect the maximum one in the profile ($\Delta(< c_j, v_j >; \Pi')$) as follows:

$$\Delta(< c_j, v'_j >; \Pi') = v'_{MAX} - v'_j$$

This difference quantifies the normalized amount of information needed to balance each non-maximal category c_j with respect to the most prevalent one c_{MAX}.

3.2.2. Fake message construction

As a final stage, for all non-maximal categories, and starting from the c_j for which Δ is the largest (*i.e.,* the one with the least prevalence in the user profile), the *noise generator* estimates the number of terms of the c_j category ($\Psi(c_j)$) that should be added to future messages so that the resulting normalized weight of c_j would equal the maximal one c_{MAX}. These terms will be randomly retrieved from the subcategories sc_{jk} that are specializations of c_j in ODP (*i.e.,* $sc_{jk}\ is-a\ c_j$). For each c_j, the *number of terms* to be added (Ψ) is a function of the *amount of information* that it is needed (Δ). To obtain an estimation of Ψ from Δ, we compute the inverse of the expressions presented in Section 3.1.3. Δ is divided both by the IC of the category ($IC(c_j)$) (to undo the category normalization introduced in the last equation of Section 3.1.3) and by IC of the terms to be added from ODP (to undo the second equation in Section 3.1.3). Since these last terms will be randomly retrieved from the c_j category in ODP, we estimate their IC as the *average IC* of all subcategories sc_{jk} of c_j in ODP (*i.e.,* $\overline{IC(sc_{jk})}, \forall sc_{jk} \in ODP$ where $sc_{jk}\ is-a\ c_j$).

$$\Psi(< c_j, v'_j >; \Pi') = \frac{\Delta(< c_j, v'_j >; \Pi')}{IC(c_j) \times \overline{IC(sc_{jk})}}$$

For example, if Ψ for the non-maximal category *"health"* is three, the *noise generator* may randomly retrieve from ODP: *"cancer"* , *"allergy"* or *"herpes"*, being all of the taxonomical specializations (*i.e., is-a*) of *"health"* .

Retrieved subcategories for all non-maximal categories are then put together in the form of a new message (*i.e.,* a fake message fm) to be published together with user posts. This represents the semantically correlated noise added to balance the user profile. Due to this added noise is based on accounting the IC of message terms, we refer it as *IC-based semantic noise addition.*

Since fake messages are raw lists of terms of different domains put together without a narrative thread, human readers would easily distinguish them from those created by the original user. On the contrary, an automatic profiling system processing messages as a whole according to term distribution (as done in related works) would obtain an homogeneously distributed profile that would reveal no new knowledge about user characteristics.

As it will be detailed in the next section, these fake messages are added to the user's account progressively as the user introduces new posts (which remain unaltered).

3.3. System's life cycle

In this section we describe the life cycle of the *profiler* and *noise generator* modules, detailing how they act in an automatic, unsupervised and adaptive way from the initial point in which the user creates an account and starts posting messages to the moment in which the user profile is considered well-balanced. Figure 1 shows the overall flow of the proposed system.

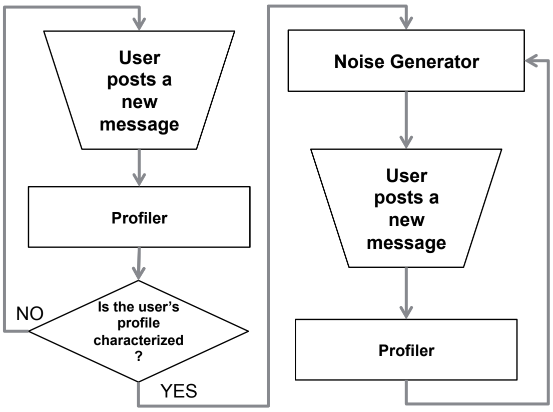

Figure 1: System's life cycle

System's iterations are controlled by the message publication rate of the user u. For each user message, the system will execute the *profiler* and, if necessary, the *noise generator* module.

At the beginning, when no messages have been published by u, our system has no knowledge about her profile. So, given a set of predefined categories C to be considering during the profiling, the profile of the user u is initialized as $\Pi' = \{< c_1, 0 >, \ldots, < c_k, 0 >\}$. To be able to learn the user profile in an unsupervised manner, the system waits for the user to post messages. For each message m_i, the profiler module will iteratively compute and update Π', as described in Section 3.1.3. Note that each new message results in a system iteration updating the accumulated information gathered for each category, so that the profile computed after uploading the i-th message will reflect the aggregation of all previous ones $(t_0, \ldots, t_i)$.

The noise generator, on the other hand, waits until the user profile is characterized before starting introducing fake messages (fm). The idea is delaying the introduction of semantically correlated noise until the user profile is, at least, roughly characterized, so that the added noise will help to hide that profile. The profiler considers that Π' is characterized when the most prevalent category (c_{MAX}, *i.e.,* the one with the highest normalized weight) is stable. To detect this, the profiler looks for s consecutive user messages/iterations (where s is a constant, *e.g.,* 10), for which c_{MAX} remains the same. To characterize profiles, we focus on a stable c_{MAX} rather than on a stable category window (*e.g.,* the first three are stable, according to their weights), because profiles will be characterized faster (and hence, the noise generator will act sooner), and because c_{MAX} is precisely the category that will guide the noise addition, so that its detection is most influential in that stage. Moreover, since the profile will be updated as new messages are posted, the system will adapt the profiling/noise addition to the characteristics of user posts, iteratively refining the profile and its balancing. According to the consistency and homogeneity of user posts with respect to the mentioned topics, the profiler would require more or less messages/profiling iterations to characterize the user profile. In any case, the profile detection rule (*i.e.,* s) should be set up in a way that the noise addition module could act before general profilers, considering that some of them use message windows of several months or hundreds of messages (Peerindex, 2012).

Let us consider that the profiler concludes that Π' is stable after the i-th profiling iteration (*i.e.,* after the m_i user message) obtaining the profile characterization Π'_i. At this moment, the *noise generator* will act as described in Section 3.2.2, computing and retrieving, according to the amount of information required to balance c_{MAX} in Π'_i, a number of terms from ODP. If the number of required terms fit in a single message, the system will create a single fake message (fm) to be posted to the user's account. At this point, restrictions regarding the maximum length of the published message could be taken into consideration according to the publishing environment (*e.g.,* for Twitter only 140 characters per message are allowed). This states the *maximum noise* that can be added for each fm, a parameter that can be configured for each Web 2.0 application or according to user preferences. If the fm surpass the *maximum noise* length, the *noise generator* will only publish the first terms in fm that fit in the maximum, leaving the rest for future fake messages. Hence, the *maximum noise* parameter influences the response time of the system with regards to profile balancing and also the degree of intrusion (*i.e.,* amount of fake information added at each iteration).

From this moment, a new iteration of the *noise generator* will be executed for each new message published by the user. Concretely, for the $(i+1)$-th post, the $(i+1)$-th system iteration will be executed, recomputing Π'_{i+1} considering the fake message (fm) created by the *noise generator* and the new user message (m_{i+1}). In this manner, Π'_{i+1} reflects the complete user message history, including the added *semantic noise*. The noise generator computes again the number of terms to be added according to Π'_{i+1} and creates, if necessary, a new fm. As new fake messages are created, the user profile will tend to balance while the system dynamically adapts its behavior to the new user messages. The reaction time for profile balancing will depend on the homogeneity of user messages according to the computed profile and the *maximum noise* allowed for each user message. Moreover, the fact that the system dynamically re-computes the user profile at each iteration allows adapting to changes in user preferences or topics of interest, considering also the past history.

Regarding this last aspect of profile characterization, considering the potential dynamicity of user posts (especially in microblogging sites) and the fact that the user profile may change through time, some profiling systems only consider the recent user history to characterize the user. For example, in systems like Peerindex (2012) a window of 4 months of user messages is considered to create a Twitter profile. The fact that older entries are omitted when profiling is very common (Marin et al., 2011; Salamó and López-Sánchez, 2011) and allows increasing the reaction time in front of changes in user preferences or interests.

This aspect has been also considered in our system, defining a profiling window that can be configured according to time (*i.e.,* only messages in the x immediate months are evaluated) or number of messages (*i.e.,* only the x most recent posts are evaluated). Formally, considering a profiling windows of x posts, when the system reaches the m_{x+1} message, the contribution in Π' of the oldest post in the profiling window (m_1) is removed (*i.e.,* the amount of information added by m_1 is subtracted) and the profiling window is shifted. Note that fake messages do not count in the updating of the profiling window.

Finally, note that the fact that the profiler/noise addition modules act for each new user message creating a fake one, does not necessarily implies that original and fake messages must be published consecutively in a real scenario. Fake messages could be stored in a queue and their publication rate could follow a random distribution, so that sometimes user messages are not followed by a fake one and, other times, several fake messages are published together. This would prevent an straightforward detection of published fake messages.

4. Evaluation of the proposed scheme

In this section we detail the evaluation of the proposed system considering both the *profiler* and *noise generator* modules. For the first one, we test the accuracy of the adaptive profile characterization detailed in Section 3.1.3; for the second one, we quantify the degree of profile balancing achieved according to the metrics detailed in Section 3.2.1.

We focus on Twitter as evaluation environment. As discussed in the introduction, Twitter is especially challenging from the profile-prevention point of view, due to the high dynamicity of user interactions, and the fact that it is a heavily user-centered environment, where users commonly publish personal data in an open manner (Rui and Whinston, 2011). This has motivated the creation of many Twitter profiling systems like those discussed in Section 2. However, Twitter is also challenging from the profiling point of view due to the short and many times ungrammatical *tweets* (in Twitter, user publications are referred as *tweets*), which makes difficult the extraction (Ebner et al., 2010; Zoltan and Johann, 2011; Bernstein et al., 2010; Abel et al., 2011), and also from the noise generation perspective due to limitations in the amount of terms that can be added per tweet.

4.1. Evaluation of the profiler module

By profiling categories in C we defined eight well-differentiated general ones corresponding to root categories in the ODP hierarchy with the minimum overlap. As shown in Table 1, even though all categories have similar degrees of generality, they can be distinguished according to the amount of web resources indexed by ODP for each one, normalizing their contribution to the user profile (see Section 3.1.3).

As evaluation data, we picked up tweet sets of real users with well-distinguished profiles. For each of the eight profile categories in C, we picked

Table 1: Categories considered in the user profile. Numbers correspond to the amount of web resources classified per category in ODP (last accessed: November 30th, 2011)

Category	Number of web resources in ODP
Arts	213839
Health	54673
Shopping	83164
Science	105834
Computers	102481
Sports	91472
Society	213630
Business	216293

the most relevant Twitter user from *wefollow* (Wefollow, 2012)[4], a web site devoted to classify the most relevant and influential Twitter users. Moreover, in order to evaluate a less defined profile, an extra user covering heterogeneous topics (*i.e.,* general news) and, hence, not belonging to any of the well-defined profile categories in C, was also selected. Concretely, the most relevant user among the delivers of general news and information was picked from *wefollow* (*i.e.,* CNN breaking news). Table 2 shows the set of nine users to be used in the evaluations and their description.

An advantage of using *wefollow* is that these users have been manually classified in the same categories considered for profiling. By comparing this humanly-tailored classification with the profile automatically inferred by our system, we can evaluate the accuracy of the *profiler* module. For each user, their 100 most recent tweets have been taken as evaluation data. The *profiler* module has been configured with a profiling window of 100 posts (*i.e.,* the whole set) and $s = 10$ (see Section 3.3), where s states the amount of analyzed tweets for which the most prevalent category (c_{MAX}) should not change in order to consider the user profile stable and start adding noise. Table 3 compares the profile detected for each user according to: (i) a human-made classification (*i.e., wefollow*); (ii) the profiler module presented in this paper (which considers the normalized amount of information provided by the terms of each category); and (iii) a profiler based solely on the absolute frequency

[4] `http://wefollow.com` (last accessed: November 16th, 2011)

Table 2: Twitter users considered for evaluation, together with their description.

User	Description
@MuseumModernArt	Museum of Modern Art, NewYork
@goodhealth	Health beauty magazine
@zappos	Tony Hsieh, CEO of zappos.com (shoes, clothing)
@CERN	European Organization for Nuclear Research
@thurrott	Webmaster of the Windows SuperSite
@London2012	Official Olympics and Paralympics channel
@celebritygossip	What are the celebrities doing?
@exectweets	Information for business executives
@cnnbrk	CNN breaking news

of terms belonging to each category (instead of their normalized IC), but using the same profiling procedure and parameters as our method.

For most users, *our profiling method* was able to accurately detect the most appropriate category compared to the human classification. Only for the user *@celebritygossip*, our scheme got *Arts* as the main category (c_{MAX}) instead of *Society*, even though *Society* appeared second in the ranking of categories. In this case, only 10 tweets were required to -wrongly- characterize the user. This happened because of the dominance of art-related topics in the first published tweets. This shows the importance of dynamically re-computing the user profile for each new tweet as our method does, so that a potential miss-profiling could be redirected. The case of *@zappos* should be discussed too. This user is classified as *Shopping* but the *profiler* has assigned *Business* as the main category. Even thought the profiler has not been accurate in this case, this result is also reasonable because the tweets of this user are written by the CEO behind the shop. Regarding the *frequency-based profiler*, it has not properly classified *@thurrott* and *@exectweets*. Finally, both approaches have profiled the general user @cnnbrk as *Society*. This is normal because this user does not belong to any of the eight profile categories in C (it is tagged as *News* in the human-made classification) and *Society* is the most general category in C.

Considering the number of tweets needed to get stable profiles, in general, we observe that when both approaches assign the same category to a certain user, our profiler requires less tweets than the frequency-based one. In

Table 3: Twitter users considered for evaluation, together with their profile detected according to three different mechanisms: a human-made classification (Wefollow, 2012), the profiler module presented in this paper and profiler based solely on absolute frequencies of terms. For the last two mechanisms, the number of tweets analyzed before considering the profile stable (with $s = 10$) is also provided.

User	Human-made classification	Our method	# tweets	Freq.-based profiler	# tweets
@Museum-ModernArt	Art	Arts	10	Arts	11
@goodhealth	Health	Health	14	Health	17
@zappos	Shopping	Business	45	Business	50
@CERN	Science	Science	10	Science	11
@thurrott	Computers	Computers	22	Business	18
@London2012	Sport	Sports	12	Sports	16
@celebrity-gossip	Society	Arts	10	Arts	10
@exectweets	Business	Business	22	Society	12
@cnnbrk	News	Society	20	Society	21

addition to that, it is worth to mention that in the cases where the frequency-based profiler has been faster (*i.e., @thurrott* and *@exectweets*), this method has missed the proper category. This is interesting since the goal of our profiler is to achieve accurate profiling as soon as possible to start introducing noise in front of external profilers.

Focusing on our profiling mechanism and the number of tweets required to profile each user, we observe notable differences. Some users with heavily biased profiles towards a certain topic (like *@MuseumModernArt* for *Arts* or *@CERN* for *Science*) are easily characterized analyzing only s tweets (*i.e.,* the main category was correctly detected and maintained from the first tweet to the s-th one). On the other hand, we find users whose tweets cover different topics (*@zappos* is the clearest example), and for which the profile is more diffuse and variable. Table 4, compares two extreme users (*@CERN* and

@zappos) according to the ranking of profile categories in function of their accumulated weights. For *@CERN*, the *Science* category weights almost a 70% in the profile. For *@zappos*, the *Business* category weights less than a 25%, while the first four categories constitute more than a 75%. In this last case, the fact that the first ranked categories were very close with respect to their weights caused continuous variations in the profile characterization, which are reflected in the fact that 45 tweets were needed to detect it.

Table 4: User profile characterization according to accumulated normalized category weights for *@CERN* and *@zappos*

@CERN profile after 10 tweets		*@zappos* profile after 45 tweets	
Category	Weight	Category	Weight
Science	1005.75	Business	1019.31
Shopping	112.91	Arts	940.72
Computers	112.15	Society	703.46
Arts	67.86	Computers	620.42
Health	60.13	Sports	461.94
Society	54.64	Shopping	323.03
Sports	39.81	Science	200.23
Business	35.22	Health	100.41

4.2. Evaluation of the noise generator module

This section discusses the evaluation results of the *noise generator* module from the moment in which it starts adding fake tweets (*i.e.,* when the user profile is characterized as shown in Table 2).

To numerically quantify the balance θ of a user profile Π' after each published tweet (which results in a system iteration, as described in section 3.3) and, hence, to test the influence of the *noise generator* module, we sum the differences Δ in the amount of information v'_j for each category c_j in Π' with respect to the maximum one, c_{MAX} (see Section 3.2.1). Then, we normalize it by the total amount of information needed to balance a profile with respect to c_{MAX} in the worst case (*i.e.,* when the contribution of the other non-maximal categories is zero). The normalizing factor corresponds to the product of the number of non-maximal categories in Π', this is $|\Pi'|-1$, by the amount of information of c_{MAX}, that is v'_{MAX} as discussed in Section 3.2.1.

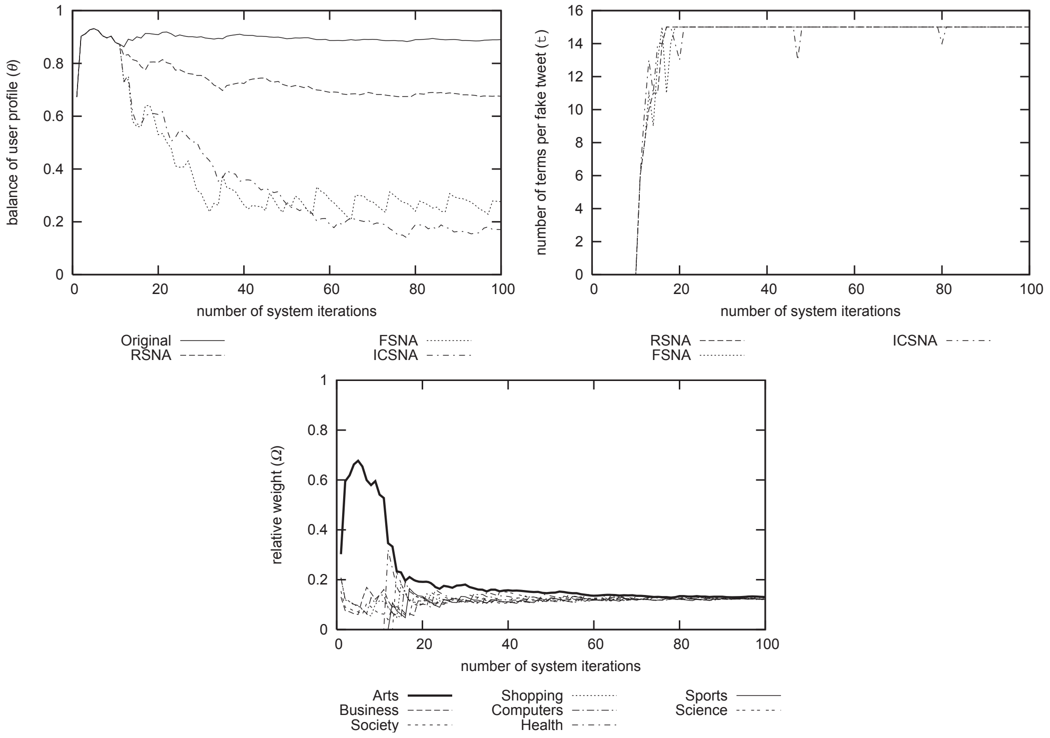

Figure 2: Results for @MuseumModernArt

$$\theta(\Pi') = \frac{\sum_{\forall < c_j, v'_j > \in \Pi'} (\Delta(< c_j, v'_j >; \Pi'))}{v'_{MAX} \times (|\Pi'| - 1)}$$

The numerical interval of θ goes from 0 to 1, where 0 means a perfectly balanced profile (*i.e.,* zero difference between all non-maximal categories with respect to the most prevalent one) and 1 means maximal difference (*i.e.,* the contribution of all categories except the maximum one is zero).

As described in Section 3.2.2, the *noise generator* computes and retrieves new terms from ODP creating, if necessary, a fake tweet per user tweet. Considering the limitations imposed by Twitter regarding message lengths (a maximum of 140 characters), a maximum number of terms will be added to the fake tweet so that its length (including spaces) fulfills the Twitter restriction. Considering that the average length of terms in ODP is 8, up to 15 terms can be fitted with separators within a tweet, hence, this is the *maximum noise* parameter (as defined in Section 3.3). Note that, analyzing

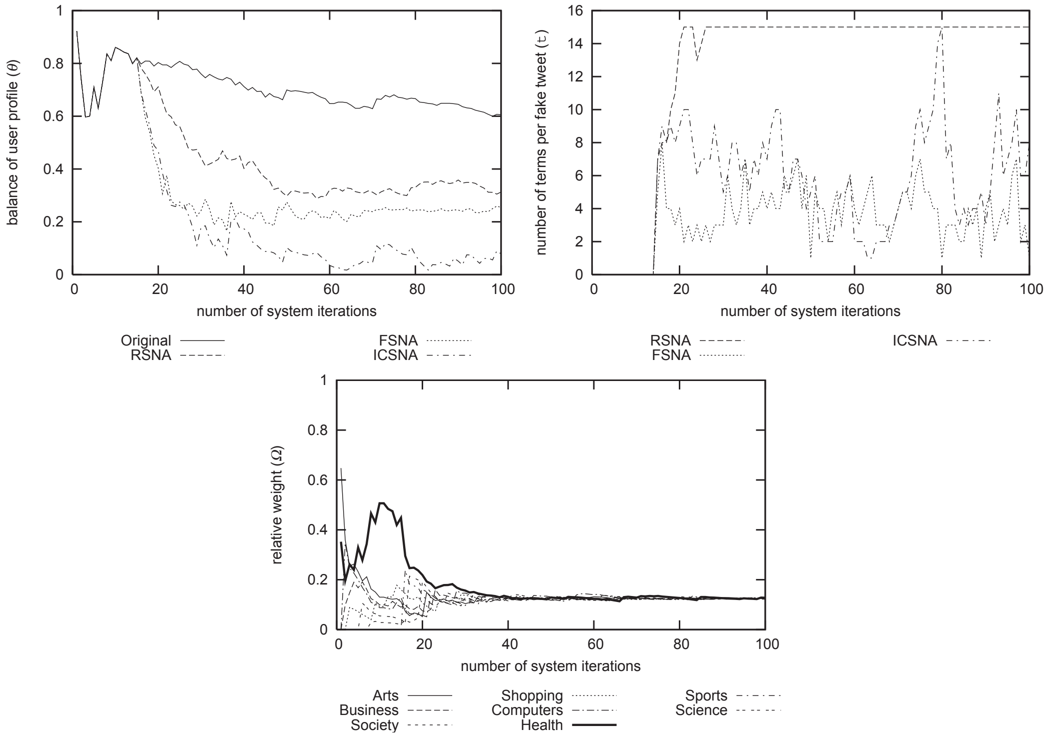

Figure 3: Results for @goodhealth

the number of terms introduced in each fake tweet allows to ascertain the level of noise introduced by a noise generator and, hence, it may be important to evaluate its performance.

Finally, another interesting point to be evaluated is the evolution of the weights of each category in a user profile when using our proposed noise generator module. In order to reflect that, we quantify the *relative weight* of each category i (Ω_i) in the user profile Π' after each system iteration. This is computed as follows:

$$\Omega_i = \frac{v'_i}{\sum_{\forall < c_j, v'_j > \in \Pi'} v'_j}$$

Due to the lack of related works proposing profile distortion methods for Twitter (and microblogging, in general) to which compare ours, we have implemented two more basic strategies for semantic noise addition:

- *Random semantic noise addition (RSNA).* At each iteration, the fake

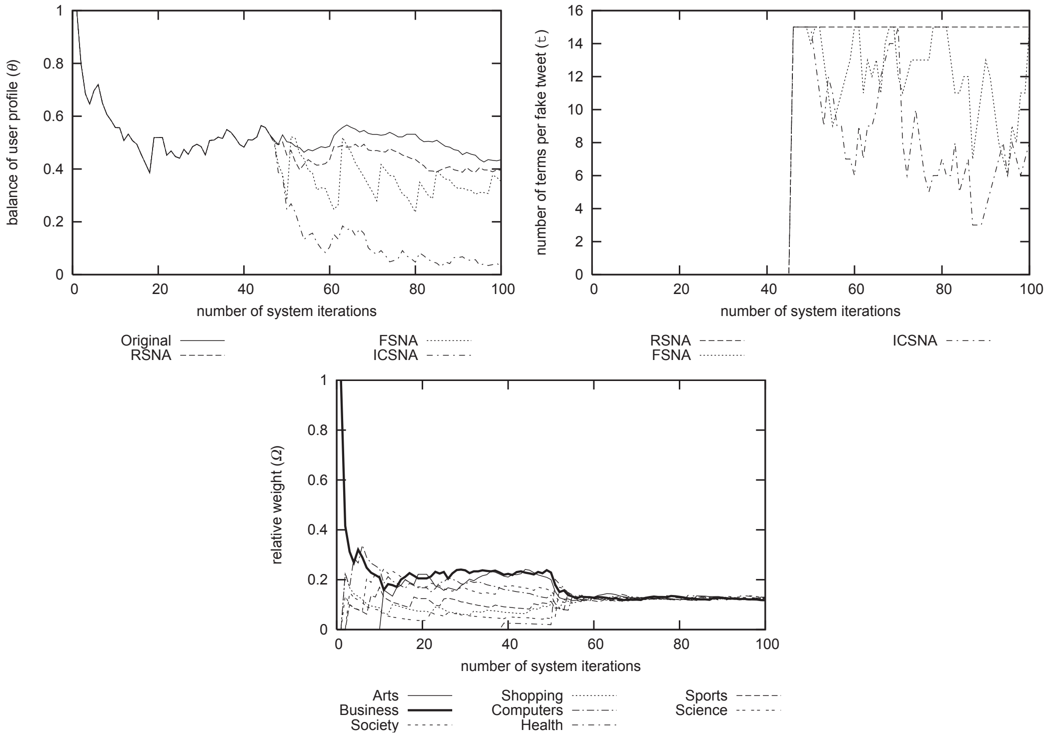

Figure 4: Results for @zappos

tweet is constructed by adding a number of t terms of random categories (where t is computed in the same adaptive manner as detailed in Section 3.3). In this case, the semantics associated to user tweets and fake ones are not considered. This method follows the same principles as numerical random noise addition and states the baseline for profile distortion methods based on *semantic noise addition*.

- *Frequency-based semantic noise addition (FSNA).* This second strategy, instead of basing the profile analysis and the fake tweet construction on the IC of terms, it solely relies on the absolute amount of term occurrences per category. As a result, fake tweets are constructed to balance the number of occurrences (*i.e.,* all categories should present the same number of occurrences), without considering the amount of information provided by each term to each category or by the categories themselves. The number of terms t added for each fake tweet is also based on the number of occurrences needed to balance the most

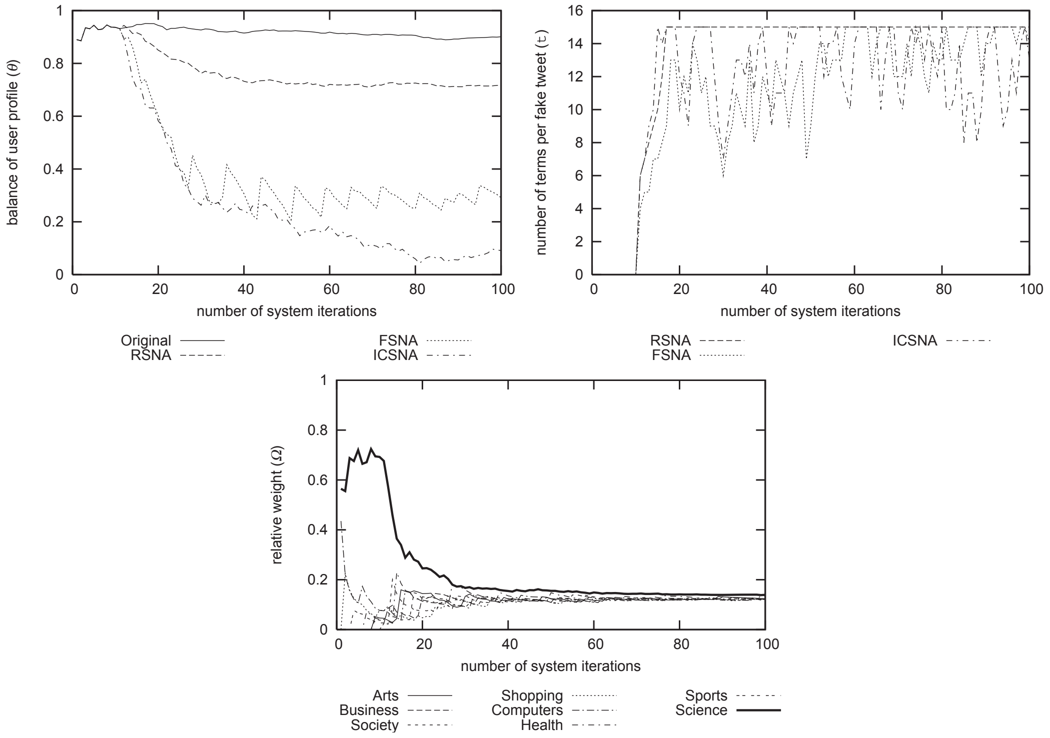

Figure 5: Results for @CERN

prevalent one.

These methods have been compared with our *IC-based semantic noise addition (ICSNA)* method for the nine considered users according to the profile balancing metric (see Section 3.2.1) and the amount of noise added by each strategy (*i.e.,* number of terms t per fake tweet).

Additionally, and only related to our ICSNA mechanism, a graph for each user that presents the evolution of the relative weight of each category after each system iteration is also included. The main category linked to each user (*e.g., Arts* is the main category linked to @MuseumModernArt) is highlighted to show its evolution in a clearer way.

Results are reflected in Figures 2-10. Provided graphs show the evolution of each analyzed aspect (*i.e.,* balance of each user profile, number of terms per fake tweet and relative weight of each category) for each system iteration that, as described in section 3.3, are controlled by the tweet publication rate of the user. Note that the first iterations only consider original tweets since

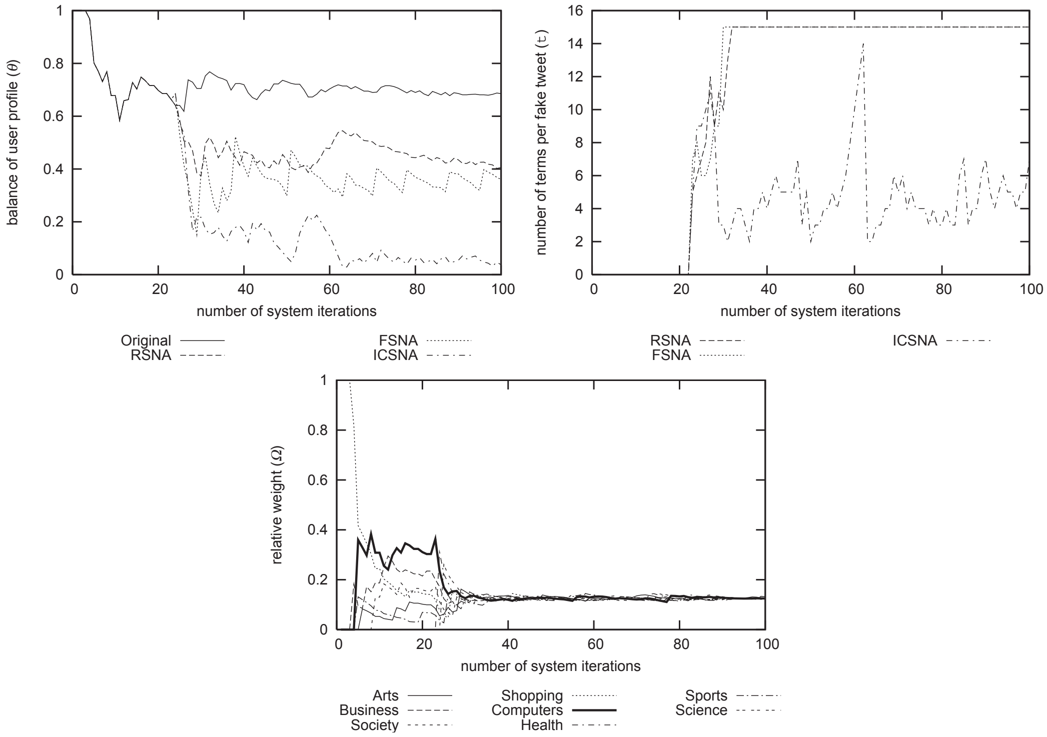

Figure 6: Results for @thurrott

the user profile is not characterized yet, whereas the latter ones consider both the original tweets and the fake ones introduced by the *noise generator*.

Several conclusions can be drawn from the analysis of the graphs. First, for all users, the proposed ICSNA method provides a better profile balance than the others, with θ values that are kept in most cases below 0.1 (in average, 54 published tweets are required to achieve this). Compared to the frequency-based method (FSNA) we observe a notorious difference, which is more noticeable for some users (*e.g., @London2012*) than for others (*e.g., @MuseumModernArt*). This is motivated by the fact that the most prevalent category for a user (*e.g., Sport* for *@London2012*) is more concrete than for another (*e.g., Art* for *@MuseumModernArt*). Due to the lack of normalization of category contribution for the FSNA method, the relative differences against the ICSNA method are increased. This fact, together with the more accurate quantization of terms' contributions to the profile, explain the better balancing and softer shape of the ICSNA graph.

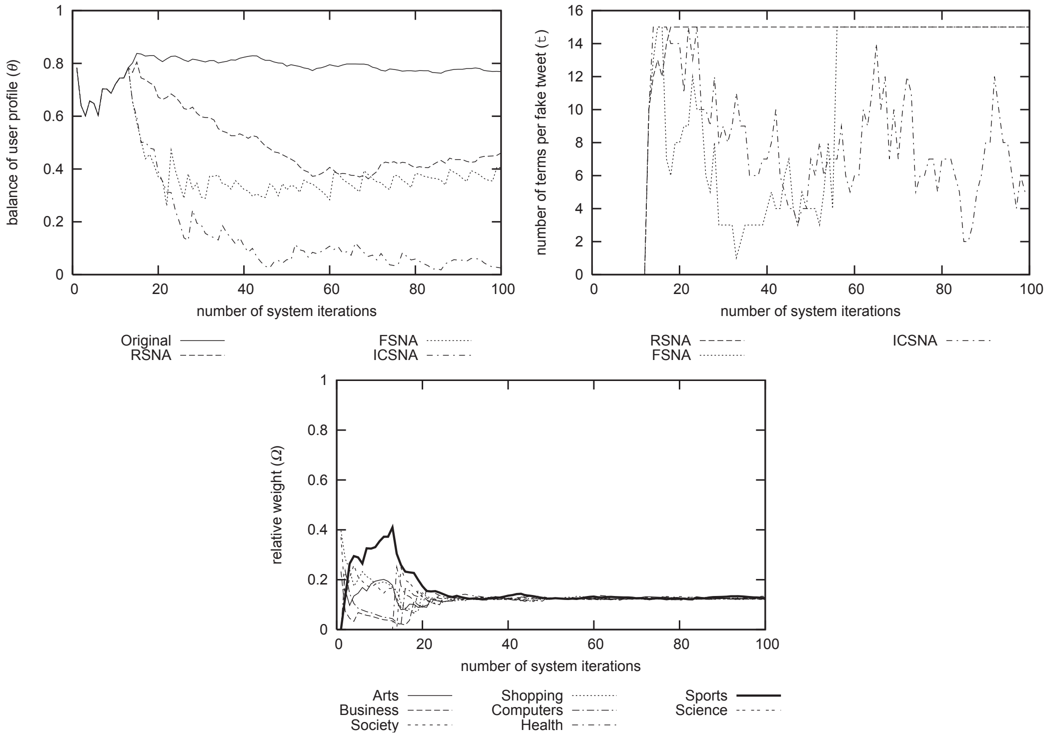

Figure 7: Results for @London2012

The comparison against the random approach (RSNA) is also interesting. For some users with more heterogeneous profiles (*e.g., @Exec, @Thurrot*) we observe that a random addition of terms is able to even rival the FSNA. This is because, with heterogeneous profiles, categories are more uniformly distributed and, hence, by adding uniform noise, they slowly tend to balance. Other users with more focused profiles (*e.g., @MuseumModernArt, @CERN*), in which a category clearly dominates, show greater differences between semantically-grounded (*i.e.,* FSNA and ICSNA) and non-semantic methods (*i.e., RSNA*). In these cases, the addition of uniform noise produces a less noticeable difference in the profile balancing due to the fact that added information is compensated by the significant prevalence of the dominant category. On the contrary, the use of semantically-correlated noise results in quicker and better balancing of user profiles due to these focus on adding the least relevant categories while avoiding the dominant one.

To more accurately analyze profile balancing figures, these should be

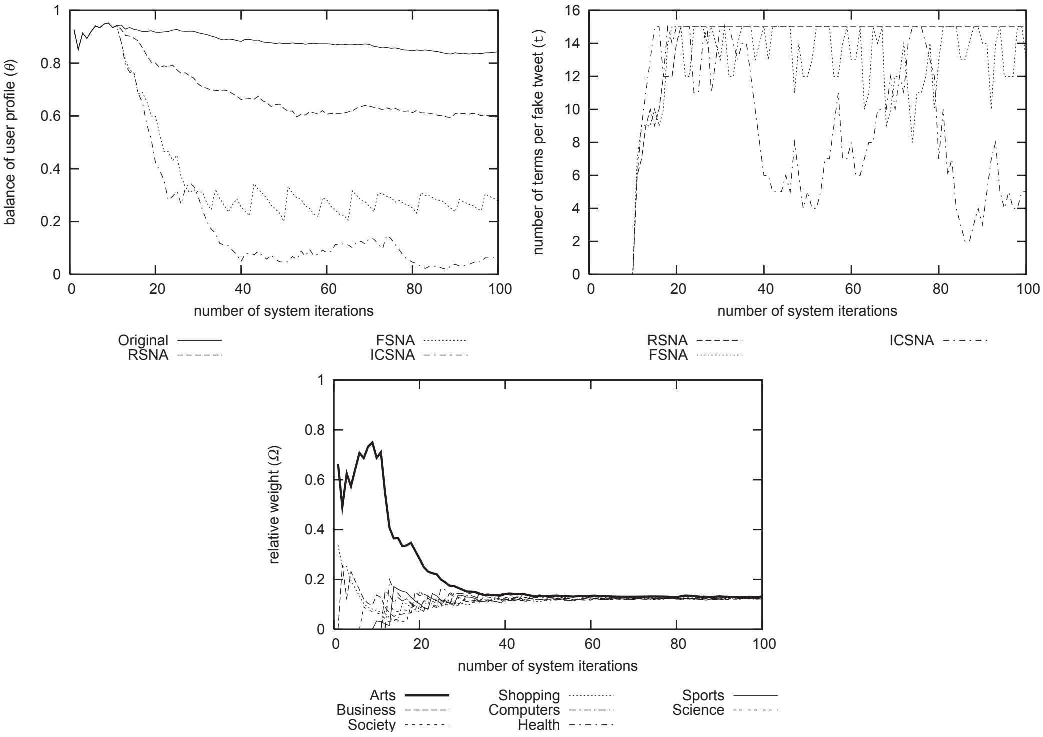

Figure 8: Results for @celebritygossip

taken in conjunction with the amount of tweet terms added per fake tweet. On one hand, these graphs show the exact moment in which the system decides to start adding noise, a moment that constitutes the starting point for the profile balancing. On the other hand, we observe that the random method (RSNA) always results in adding the maximum number of terms per fake tweet (*i.e.,* 15) because its assessed profile never achieves the pursued balance. Knowledge-based methods (FSNA and ICSNA), on the contrary, adapt the number of terms to the needs of each new tweet, so that only the required amount of noise is added at each moment. By looking at the graph shapes, we observe how the spikes in the profile balance (that indicate an increase in the dominance of a particular topic) are tamed by adding more terms at the next iteration (see *@goodhealth*, for example), while the amount noise is reduced when the profile tends to minimize. The amount of noise is also consequent with the user profile, showing larger values for users with very focused profiles (*e.g. @CERN* and *@MuseumModernArt*) that are more

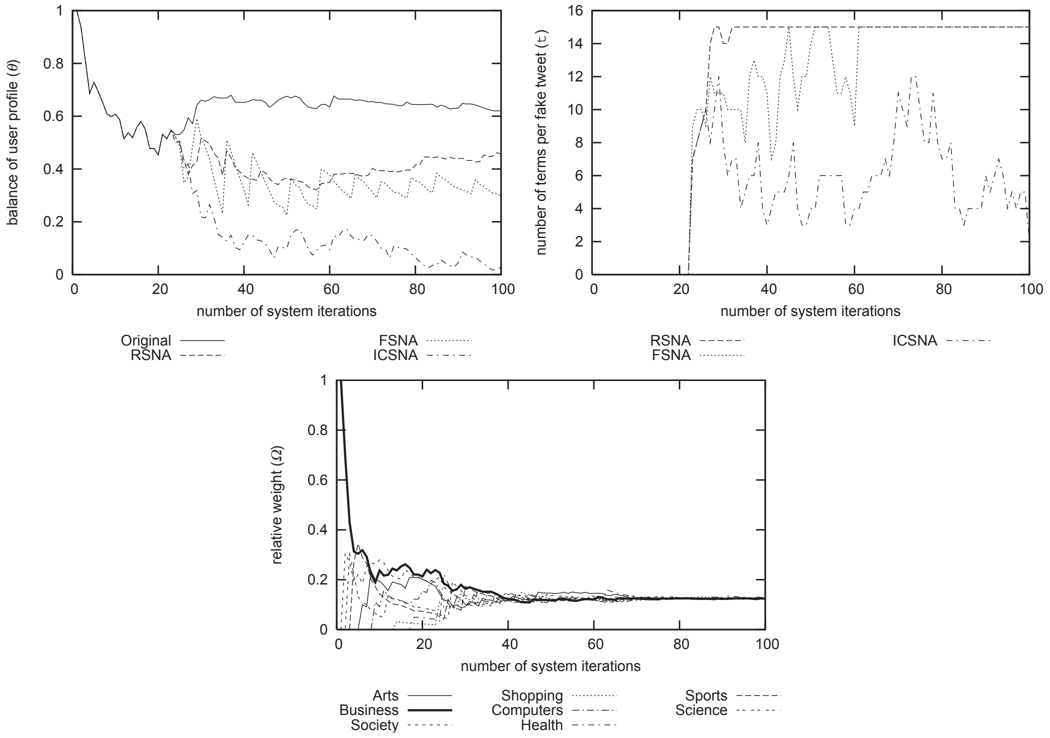

Figure 9: Results for @exectweets

difficult to balance than more heterogeneous users (*e.g., @thurrot*) for which their profiles are less exposed. Finally, we observe that the FSNA method tends to add more noise than the ICSNA, a circumstance that is also synchronized with a spikier graph. This is motivated by the coarser profile characterization, based only on the number of occurrences compared to the finer profile quantification of the ICSNA method. From the IC perspective this results in a less accurate profile balancing being, in some cases (*e.g., @Exec, @thurrot*), unable to achieve a balance even though the maximum amount of noise is added.

Regarding the relative weight of each category for each user after each system iteration, in all the cases, it can be seen how in the first tweets the main category has the most relevant weight and then, when the noise generator module starts to introduce fake tweets, it decreases until becoming hidden among the relative weights of the other categories.

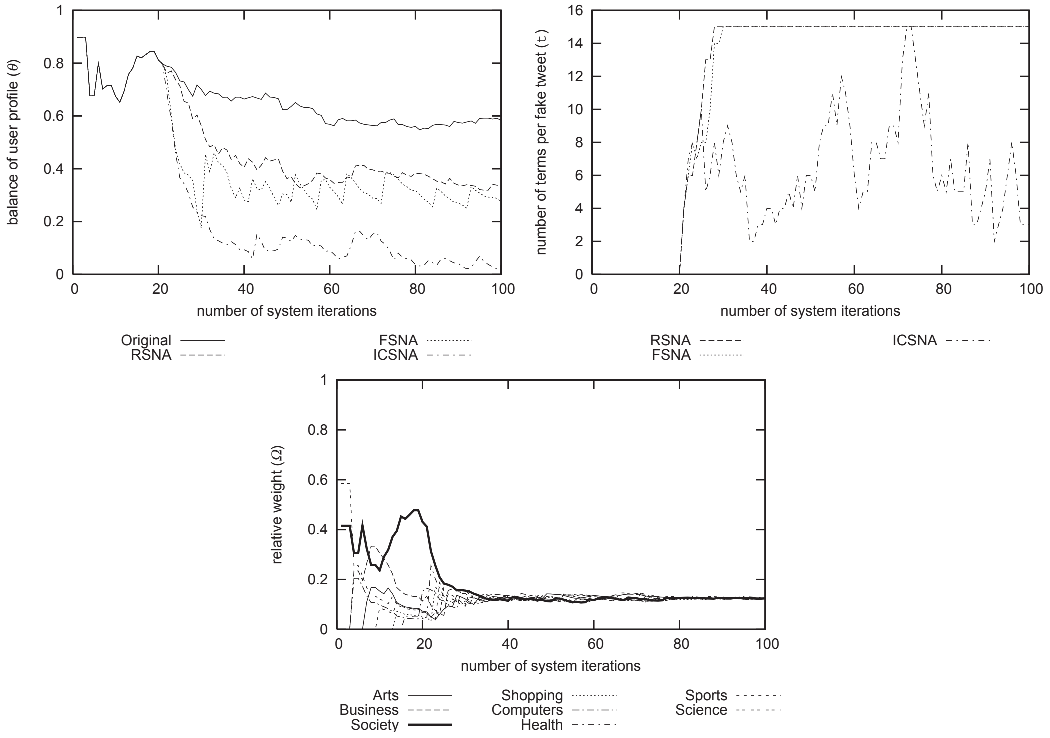

Figure 10: Results for @cnnbrk

5. Concluding remarks

In this paper, we have addressed the problem of preventing profiling of users of text-based Web 2.0 applications in front of automatic systems that gather and analyze their publications.

Our solution is meant to minimize the impact on both users and their readers. In this way, one of our premises is that adding new information to distort user profiles is more admissible than modifying or restricting user contents as other methods suggest. As stated previously, users contribute to Web 2.0 applications to seek attention from others and they might not be willing to use a privacy method that modifies or restricts their legitimate publications. Therefore, a successful approach should be based on introducing fake information that effectively hides the user profiles while enabling readers to easily discern between that fake data and the original one. In order to achieve this, we exploit the fact that human readers interpret messages according to their semantics, whereas profiling systems are mainly based on analyzing information distribution.

As a result, we have proposed a new knowledge-based scheme that generates fake messages together with legitimate ones. Fake messages contain a concatenation of terms correlated with respect to the original data so that user profiles can be balanced and, hence, hidden. As discussed in the related work section, already existing profilers which are used to compile user profiles characterize users according to the information distribution computed from term occurrences in posted messages. Therefore, they will equally count fake and original terms, obtaining a more homogeneous (*i.e.,* less distinguished) user profile. Regarding the impact on human readers, in order to allow them to discern between legitimate messages and fake ones, the latter lack a semantically coherent discourse.

Our proposal has been evaluated using Twitter, a very well-known Web 2.0 microblogging platform, which is especially challenging due to its heavy limitations in post lengths and its high dynamicity. Evaluation results achieved by our proposal prove that it effectively balances user profiles in front of automatic profiling based on term distribution. Among its advantages, we stress that it is adaptive and capable of profiling and obfuscating users with a quite limited number of publications. Besides, it provides a fast response time to legitimate modifications of the user profile (users are not static and their interests may vary through time).

As for future work, it would be interesting to analyze the limitations of the proposed scheme when facing ad-hoc profiling methods specially designed to detect fake tweets. More specifically, knowing how the proposed method behaves, ad-hoc systems that recognize and omit fake messages could be developed using trained classifiers or defining specific detection rules. The analysis should discuss the cost related to their development (*e.g.,* design, training, etc) and it also should include different approaches to thwart them. Possible countermeasures could be based on modifying the way tweets are generated (*e.g.,*, with grammatically, but not semantically, coherent constructions) and published (*e.g.,* simulating the publication rate of users across time and randomly distributing them across original messages) or combining different knowledge bases or even random terms.

In addition, we are also planning to test the proposed system with other text-based environments without length limitations (*e.g.,* classic blogs), so that the fake message construction may have more flexibility allowing, for example, the delay and concatenation of several ones. Finally, it could be interesting to evaluate its performance, its convenience and degree of acceptance in the long term when used daily by regular users.

Disclaimer and acknowledgments

Authors are solely responsible for the views expressed in this paper, which do not necessarily reflect the position of UNESCO nor commit that organization. This work was partly supported by the Spanish Ministry of Science and Innovation (through projects eAEGIS TSI2007-65406-C03-01, CO-PRIVACY TIN2011-27076-C03-01, ARES-CONSOLIDER INGENIO 2010 CSD2007-00004 and Audit Transparency Voting Process IPT-430000-2010-31), by the Spanish Ministry of Industry, Commerce and Tourism (through projects eVerification2 TSI-020100-2011-39 and SeCloud TSI-020302-2010-153) and by the Government of Catalonia (under grant 2009 SGR 1135).

References

Abel, F., Gao, Q., Houben, G.J., Tao, K., 2011. Semantic enrichment of twitter posts for user profile construction on the social web, in: Proc. of the 8th extended semantic web conference on The semanic web: research and applications – ESWC'11, pp. 375–389.

Abril, D., Navarro-Arribas, G., Torra, V., 2011. On the declassification of confidential documents, in: Proc. of Modeling Decisions for Artificial Intelligence – MDAI'11, pp. 235–246.

Achrekar, H., Gandhe, A., Lazarus, R., Yu, S.H., Liu, B., 2011. Predicting flu trends using twitter data, in: Proc. of the IEEE Conference on Computer Communications Workshops – InfoCom, IEEE Computer Society. pp. 702–707.

Bernstein, M., Suh, B., Hong, L., Chen, J., Kairam, S., Chi, E., 2010. Eddi: Interactive topic-based browsing of social status streams, in: Proc. of the 23nd annual ACM symposium on User interface software and technology, pp. 303–312.

Bilton, B., 2010. Price of facebook privacy? start clicking, in: The New York Times.

Brand, R., 2002. Microdata protection through noise addition. Inference Control in Statistical Databases , 97–116.

Brown, E., 2012. Twitter exposes historical data in partnership with datasift. ZDNET .

Cilibrasi, R., Vitányi, P., 2006. The google similarity distance. IEEE Trans. Knowl. Data. Eng. 19, 370–383.

Consumer Reports National Research Center, 2010. Annual state of the net survey 2010. Consumer Reports 75.

Crimes, S., 2012. Twitter sells old tweets to marketers - should users be worried? The Inquirer .

DBpedia, 2012. http://dbpedia.org (last accessed: 19/01/2012).

Ebner, M., Mühlburger, H., Schaffert, S., Schiefner, M., Reinhardt, W., Wheeler, S., 2010. Getting granular on twitter: Tweets from a conference and their limited usefulness for non-participants. Key Competencies in the Knowl. Society 324, 102–113.

Fellbaum, C., 1998. WordNet: An Electronic Lexical Database. MIT Press.

Freni, D., Ruiz Vicente, C., Mascetti, S., Bettini, C., Jensen, C.S., 2010. Preserving location and absence privacy in geo-social networks, in: Proc. of the 19th ACM international conference on information and knowledge management, ACM press.

Garcia Esparza, S., O'Mahoney, M.P., Smyth, B., 2012. Mining the real-time web: A novel approach to product recommendation. Knowl.-Based Syst. 29, 3–11.

Guarino, N., 1998. Formal ontology in information systems, in: Proc. of the 1st International Conference on Formal Ontology in Information Systems – FOIS'98, pp. 3–15.

Guha, S., Tang, K., Francis, P., 2008. NOYB: Privacy in online social networks, in: Proc. of the first workshop on Online social networks.

Hay, M., Miklau, G., Jensen, D., Towsley, D., Weis, P., 2008. Resisting structural identification in anonymized social networks, in: Proc. of the 2008 Conference on Very Large Databases – VLDB'08.

Hotonline.net, 2012. Twitter involved in a huge scandal: Selling users private data for marketing strategies. Hotonline.net .

Islam, M., Brankovic, L., 2011. Privacy preserving data mining: A noise addition framework using a novel clustering technique. Knowl.-Based Syst. 24, 1214–1223.

Li, D., Lv, Q., Xie, X., Shang, L., Xia, H., Lu, T., Gu, N., 2012. Interest-based real-time content recommendation in online social communities. Knowl.-Based Syst. 28, 1–12.

Luo, W., Xie, Q., Hengartner, U., 2009. Facecloak: an architecture for user privacy on social networking sites, in: Proc. of the 2009 International Conference on Computational Science and Engineering, pp. 26–33.

Marin, L., Isern, D., Moreno, A., Valls, A., 2011. On-line dynamic adaptation of fuzzy preferences. Inf. Sciences in press.

McMillan, G., 2011. Twitter reveals active user number, how many actually say something, in: Time - Techland.

Michelson, M., Macskassy, S., 2010. Discovering users' topics of interest on twitter: a first look, in: Proc. of the fourth workshop on Analytics for noisy unstructured text data.

Nocera, N., Ursino, D., 2011. An approach to providing a user of a "social folksonomy" with recommendations of similar users and potentially interesting resources. Knowl.-Based Syst. 24, 1277–1296.

Peerindex, 2012. http://www.peerindex.com (last accessed: 19/01/2012).

Resnik, P., 1995. Using information content to evalutate semantic similarity in a taxonomy, in: Proc. of the 14th International Joint Conference on Artificial Intelligence – IJCAI'95, pp. 448–453.

Ross, S., 1976. A First Course in Probability. Macmillan.

Rui, H., Whinston, A., 2011. Verification and validation issues in electronic voting. Inf. Syst. and E-Bussiness Manag. , 1–16.

Rui, H., Whinston, A., Winkler, E., 2009. Follow the tweets, in: The Wall-street Journal.

Sakaki, T., Okazaki, M., Matsuo, Y., 2010. Earthquake shakes twitter users: Real-time event detection by social sensors, in: Proc. of the ACM 19th Int. Conf. on World Wide Web, ACM press. pp. 851–860.

Salamó, M., López-Sánchez, M., 2011. Adaptive case-based reasoning using retention and forgetting strategies. Knowl.-Based Syst. 24, 230–247.

Salton, G., Buckley, C., 1988. Term-weighting approaches in automatic text retrieval. Inf. Proc. and Manag. 24, 513–523.

Sánchez, D., 2010. A methodology to learn ontological attributes from the web. Data & Knowl. Eng. 63, 573–597.

Sánchez, D., Batet, M., 2011. Semantic similarity estimation in the biomedical domain: An ontology-based information-theoretic perspective. J. Biomed. Inform. 44, 749–759.

Sánchez, D., Batet, M., Isern, D., 2011a. Ontology-based information content computation. Knowl.-Based Syst. 24, 297–303.

Sánchez, D., Batet, M., Valls, A., Gibert, K., 2010. Ontology-driven web-based semantic similarity. J. of Intell. Inf. Syst. 35, 383–413.

Sánchez, D., Isern, D., Millan, M., 2011b. Content annotation for the semantic web: an automatic web-based approach. Knowl. and Inf. Syst. 27, 393–418.

Sánchez, D., Moreno, A., 2008. Learning non-taxonomic relationships from web documents for domain ontology construction. Data & Knowl. Eng. 63, 600–623.

Scoble, R., 2008. Facebook disabled my account, in: Scobleizer.

Turney, P., 2001. Mining the web for synonyms: Pmi-ir versus lsa on toefl, in: Proc. of the 12th European Conference on Machine Learning – ECML'01, pp. 491–502.

TweetPsych, 2012. http://tweetpsych.com (last accessed: 19/01/2012).

Van Eecke, P., Truyens, M., 2010. Privacy and social networks. Computer Law & Security Rev. 26, 535–546.

Wefollow, 2012. http://wefollow.com (last accessed: 16/11/2011).

Wilson, D., 2011. Users are worried about social network security and privacy. The Inquirer .

Zhang, C., Sun, J., Zhu, X., Fang, Y., 2010. Privacy and security for online social networks: Challenges and opportunities. IEEE Netw. 24, 13–18.

Zoltan, K., Johann, S., 2011. Semantic analysis of microposts for efficient people to people interactions, in: Proc. of the Roedunet International Conference – RoEduNet'11, pp. 1–4.