Proceedings of the

2nd International Workshop on Semantic Adaptive Social Web (SASWeb 2011)

co-located with the

19th User Modeling, Adaptation and Personalization Conference (UMAP 2011)

July 15, 2011, Girona, Spain

http://semantic-adaptive-social-web.uniud.it/events/2011/sasweb/

ISSN 1613-0073

Organizing Committee

- Federica Cena University of Turin, Italy.
- Antonina Dattolo University of Udine, Italy.
- Ernesto William De Luca Technische Universität Berlin, Germany.
- Pasquale Lops University of Bari Aldo Moro, Italy.
- Till Plumbaum Technische Universität Berlin, Germany.
- Julita Vassileva University of Saskatchewan, Canada.

Program Committee

- Liliana Ardissono Università degli Studi di Torino, Italy
- Lora Aroyo University of Amsterdam, Holland
- Shlomo Berkovsky CSIRO, TasICT Centre, Australia
- Peter Brusilovsky University of Pittsburgh, USA
- David Bueno Vallejo Universidad de Málaga, Spain
- Ivan Cantador Universidad Autónoma de Madrid, Spain
- Francesca Carmagnola University of Turin, Italy
- Pablo Castells Universidad Autónoma de Madrid, Spain
- Cristina Gena University of Turin
- Marco de Gemmis University of Bari Aldo Moro, Italy
- Darina Dicheva Winston Salem University , USA
- Vania Dimitrova University of Leeds, UK
- Tom Heath Talis Information Ltd, United Kingdom
- Eelco Herder L3S Research Center, Germany
- Andreas Hotho University of Würzburg, Germany
- Gilles Hubert IRIT, Toulouse, France
- Geert-Jan Houben TU Delft, Holland
- Dietmar Jannach Dortmund University of Technology
- Tsvi Kuflik University of Haifa, Israel
- Vincenzo Loia University of Salerno, Italy
- Luca Mazzola ITC, University of Lugano, Switzerland
- Alessandro Micarelli University “Roma Tre”, Roma, Italy
- Cecile Paris CSIRO ICT Centre, Australia
- Francesco Ricci Free University of Bozen-Bolzano, Italy
- Giovanni Semeraro University of Bari Aldo Moro, Italy
- Sergey Sosnovsky German Research Center for AI, Saarbrucken, Germany
- Armando Stellato University of Tor Vergata, Rome
- Ilaria Torre University of Turin
- Markus Zanker University Klagenfurt, Austria
- Torsten Zesch Technical University of Darmstadt, Germany

Preface

Social Web (also called Web 2.0) is growing daily, together with the number of users and applications. In this way, users generate a significant part of Web content and traffic: they create, connect, comment, tag, rate, remix, upload/download, new or existing resources in an architecture of participation, where user contribution and interaction add value. Users are also involved in a broad range of social activities like creating friendship relationships, recommending and sharing resources, suggesting friends, creating groups and communities, commenting friends activities and profiles and so on.

At the same time, *Semantic Web* (also called Intelligent Web), whose main goal is to describe Web resources in a way that allows machines to understand and process them, has started to go out from academies and begins to be exploited in many web sites, incorporating high-quality user contributed content and semantic annotations using Internet-based services as an enabling platform.
The recent advances in the Semantic Web area, and specifically the widespread use of *weak semantic techniques* (the so-called *'lowercase' semantic web*), such as the use of microformats (e.g. eRDF, RDFa) to attach semantics to content, also provide new standardized ways to process and share information. This approach allows information intended for end-users (such as contact information, geographic coordinates, calendar events) to also be automatically processed by machines, and this obviates other more complicated methods of processing, such as natural language processing or screen scraping.

In this workshop we are interested to analyze the benefits adaptation and personalization have to offer in the Web of the future, the so called *Social Semantic Web* or *Web 3.0*, that puts together Semantic Web and Social Web.

The workshop aims at discussing the state-of-the-art, open problems, challenges and innovative research approaches in adaptation and personalization for the Social Semantic Web. It provides a forum for proposing innovative and open models, applications and new data sharing scenarios, as well as novel technologies and methodologies for creating and managing these applications. Examples of stimulating application fields are social bookmarking environments, publication sharing systems, intelligent cultural guides, collaborative working, social networking sites, digital libraries, e-learning and recommender systems.

We would like to thank all the authors for their submissions, and our Program Committee and additional reviewers for their precious work.

June 2011

Federica Cena
Antonina Dattolo
Ernesto William De Luca
Pasquale Lops
Till Plumbaum
Julita Vassileva

SASWeb 2011 Workshop Chairs

Table of Contents

Visualizing and Managing Folksonomies
Antonina Dattolo, Emanuela Pitassi

Selective Propagation of Social Data in Decentralized Online Social Network
Udeep Tandukar, Julita Vassileva

Towards a Followee Recommender System for Information Seeking Users in Twitter
Marcelo G. Armentano, Daniela Godoy, Analía Amandi

Adaptive Faceted Search on Twitter
Ilknur Celik, Fabian Abel, Patrick Siehndel

cTag: Semantic Contextualisation of Social Tags
Ignacio Fernández-Tobías, Iván Cantador, Alejandro Bellogín

Social Semantic Web Fosters Idea Brainstorming
Matteo Gaeta, Vincenzo Loia, Giuseppina Rita Mangione, Francesco Orciuoli, Pierluigi Ritrovato

Recommending #-Tags in Twitter
Eva Zangerle, Wolfgang Gassler, Gunther Specht

Visualizing and Managing Folksonomies

Antonina Dattolo and Emanuela Pitassi

University of Udine, Via delle Scienze 206, I-33100 Udine, Italy
{antonina.dattolo, emanuela.pitassi}@uniud.it

Abstract. Social tagging represents an innovative and powerful mechanism introduced by social Web: it shifts the task of classifying resources from a reduced set of knowledge engineers to the wide set of Web users. Tags generate folksonomies; in the current popular social tagging systems (such as delicious or Bibsonomy), they are difficult to manage, modify, and visualize in dynamic and personalized ways.
The aim of this paper is to describe Folkview, an innovative way to conceive a folksonomy in terms of a multi-agent system. Folkview is able to support specific modular tools for personalizing customized and dynamic visualization features allowing users to simply update, manage and modify a folksonomy.

Key Words: Folksonomy, Formal model, Multi-agent system, Personalized views, Authoring

1 Introduction

Social tagging systems are characterized by the active participation and interaction of users, which upload, share and freely annotate with labels, known as *tags*, a huge amount of resources, explicitly inducing on them personal classifications. Although these systems are widely used and personal annotations represent a democratic, powerful and easy way of classifying resources, they suffer from different issues:

- The lack and the exigence of general methodologies for extracting semantic information (this topic is widely discussed in literature, see the survey [1]);
- the lack and the exigence of personalized and dynamic workspaces in which users can *visualize personalized views* of the folksonomy or *apply personal changes*.

The creation of personalized views, which may display a limited, well defined and personalized sub-portion of an entire hyperspace is something that has already been considered in different settings. To implement this strategy, most of traditional Web browsers become to offer personalized views, so-called *start pages* such as, e.g. NetVibes[1], My Yahoo[2] and iGoogle[3]. Some extensions to

[1] http://www.netvibes.com/it
[2] http://my.yahoo.com/
[3] http://www.google.it/ig

these examples are adaptive bookmarking systems such as PowerBookmarks [2], Siteseer [3] and WebTagger [4].

These applications highlight that a crucial task for the developers of nowadays Web application is how to model and create specific tools for providing personalized views to the users. All the types of social tagging systems should deal with these compelling and open challenges, expanding their capabilities and enhancing possible multiple visualizations, in order to achieve (a) a more effective comprehension of the semantic relations of a folksonomy, (b) a more useful navigation through the involved elements, and (c) the manipulation of the existing relations among tags and resources according to the user needs.

The folksonomies (and the personomies) are generally visualized as a tag cloud: in spite of this, the work [5] states that this kind of visualization is not sufficient as the sole means of navigation. Let we consider for instance the Figure 1, where is shown a portion of workspace offered by delicious, the popular social bookmarking application.

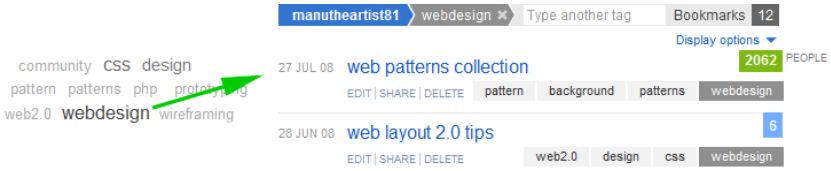


Figure 1. A sample view taken from delicious

The user navigates her tag cloud (shown on the left); when she selects a specific tag ("Webdesign", in our case), the number of bookmarks related to the chosen tag and the list of resources annotated with it are shown. The navigation may continue by clicking on each resource, tag or user, but

- the tag cloud is not adaptive;
- personalized views cannot be created;
- it is not possible to simply modify the personomy, or the personal view of the folksonomy (for example, renaming a tag for a set of resource or merging two or more tags on a unique label).

These limitations are partially ascribable to the static nature attributed in literature to a folksonomy; in fact, it has been defined in terms of finite sets of *users*, *resources* and *tags* [6] and represented as a hyper graph or as a tripartite graph [7, 8]. These definitions do not consider the dynamic aspects, like the personalization and the authoring, as intrinsic features of a folksonomy, although they are. In fact, the role and the importance of a folksonomy are not in the trivial, passive storage and visualization of data, but in the semantics contained in it, in the identification of user features, habits, needs, and in the possibility of inferring recommendations.

The main aim of this work is to propose a novel, distributed, modular system called Folkview, whereby a folksonomy is conceived dynamically through the use of multiple agents. These agents will be capable of

- managing the structural and semantic properties;
- cooperating for obtaining common objectives;
- offering personalized and dynamic views.

The paper is organized as follows: in next Section 2 we discuss related work, in Section 3 we present the formal, multi agent-based model at the basis of Folkview, while, in Section 4, we discuss its dynamic features with respect to authoring and personalized views. Finally, final considerations end the paper.

2 Related Work

Early definitions of folksonomy [9–11] are related to the user activity of annotating resources with metadata for her own individual aims, and/or for sharing them in a community. In these definitions, only three kind of entities (users, resources and tags) and the relations among them, called *tas* (tag assignments), are considered, instead of any dynamic aspect of visualization and manipulation. An extended definition of the previous ones is given in [12] where the authors propose the social application GroupMe!, defining an additional element, the group, which can be both a resource or a group. Even if some interesting relations are highlighted in this application, like the relation between tags assigned to different resources of the same group, users are not allowed either to directly manipulate her personomy or to navigate through different and more effective visualizations.
As observed in the introduction paragraph, a folksonomy is usually represented by a tripartite graph or network, but this leads to another issue related to the complexity of the nature of the graph itself. Various researches have dealt with this problem, projecting a folksonomy on simplified structures. For example, in [13], the tri-partite network is first projected on a bipartite network, then on a unipartite one, thanks to the correlations between two nodes of the same kind. In a recent work [14] the authors, starting from the edge-colored multigraph of users, tags, and resources, propose some simplified definitions that maintain some of its properties. Thanks to this mechanism, the information extraction process becomes easier and simplifies the application of a modular and extensible methodology applied for discovering synonyms, homonyms and hierarchical relationships amongst sets of tags. However, these researches are oriented to provide a different and intuitive way to visualize a folksonomy, but do not discuss about possible simple modifications of them. For example, at the best of our knowledge, there are not dynamic authoring tools that allow the user to globally change the tag labeled in a certain way within her personomy. The same social tagging applications, such as Bibsonomy, delicious or Flickr, suffer from similar limitations.

A few research projects have addressed some of them: in [15] the authors use a customized cluster maps for visualizing both the overview and the detail of semantic relationships intrinsic in the folksonomy; in [16] the authors use information visualization techniques to discover implicit relationships between users, tags and bookmarks and offer end-users different ways to discover content and information that would not have been found through explicit searches.

Another project is TagGraph[4], a folksonomy navigator which visualizes the relationships between Flickr tags. User may enter a Flickr username or a tag, and the graph sets out drawing itself automatically; after this early step, she may navigate through related tags or among related images, but could not manipulate her personomy.

The mentioned projects are by all means interesting attempts of interactive visualizations of folksonomies; nevertheless they do not provide neither personalized views nor effective dynamic changes according to the user needs or preferences.

3 Folkview: the formal model

Traditionally, given the sets U, T and R respectively of users, tags and resources, a folksonomy is defined as the set of tag assignments (tags, for short) $(u_i, r_j, t_k) \in U \times T \times R$, where $i = 1, \ldots, |U|; j = 1, \ldots, |T|; k = 1, \ldots, |R|$, each of them indicating that user u_i has tagged the resource r_j with t_k. User profiles, functions, metrics or semantic relations among users, tags, resources and tas are not intrinsic properties of the folksonomy, but may be (or not) applied by the system which hosts the folksonomy. We indicate this traditional concept of folksonomy as static folksonomy F. In order to define a F, we identify three classes of sets:

- $T_{u_i,r_j} \in T$ is the set of tags used by u_i on r_j;
- $R_{u_i,t_k} \in R$ is the set of resources tagged by u_i with t_k;
- U_{t_k,r_j} is the set of users that tagged r_j with t_k.

Each set represents a structural component of the folksonomy, and we call it *structural* ; the tags are grouped associating to them a semantic label for identifying their meaning in that dimension. A graphical example of 6 sets of tags is given in Figure 2, on the left.

The first three linear paths contain the resources tagged by user u_1, using respectively t_1, t_2 and t_3. So, the labels associated with them are respectively u_1, t_1, u_1, t_2 and u_1, t_3.

Definition 1. *A structural dimension is a labeled path*

$$D_{u_i,r_j} = (V, E, \lambda)$$

where $V = T_{u_i,r_j}$ *is the set of vertices,* E *is the set of edges,* $\lambda(e) = (u_i, r_j)\ \forall e \in E$ *is an edge labeling, and* $degree(t_k) = 0, 1, 2\ \forall t_k \in T_{u_i,r_j}$. *In particular,* $degree(t_k) = 0, 1, 2$ *only if* $|T_{u_i,r_j}| = 1$.

[4] http://taggraph.com/

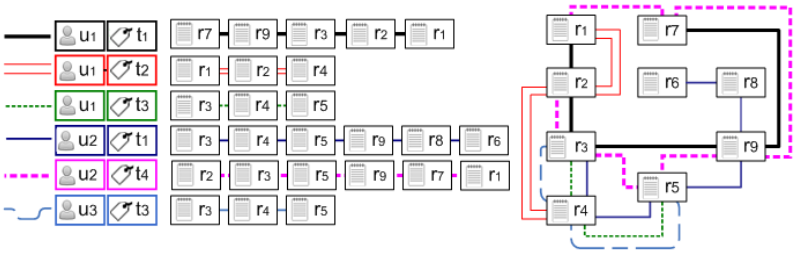


Figure 2. 6 structural dimensions (left) and the correspondingfolksonomy (right)

Analogously we define D_{u_i,t_k} (resp. D_{t_k,r_j}) as the labeled path constituted by the set of resources R_{u_i,t_k}, labeled with t_k by the user u_i (resp. by the set of users U_{t_k,r_j}, that assigned the tag t_k to the resource r_j).

Definition 2. *A static folksonomy F is a labeled multigraph given by the union of three families of structural dimensions.*

$$F = \bigcup_{i,k} D_{u_i,r_j} \cup \bigcup_{i,j} D_{u_i,t_k} \cup \bigcup_{j,k} D_{t_k,r_j}$$

where $u_i \in U$, $r_j \in R$ and $t_k \in T$.

An example of F is shown in Figure 2 (right); it is based on the six dimensions visualized on the left.

The previous definition is restrictive for a folksonomy: several works [1] emphasize the role of a folksonomy for

- supporting tag suggestions, or recommendations;
- inferring knowledge about the user profile, her habits, preferences, and skills;
- for identifying similar users, resources or tags.

In summary, folksonomies add semantics on the data. For this reason, we propose a new concept of folksonomy, conceived as a dynamic entity, organized as an universe of inherently autonomous computational sub-entities, which interact with each other by sending messages and reacting to external stimuli by executing some predefined procedural skills. Various authors have proposed different definitions of agents. In our setting an agent is formally defined as follows.

Definition 3. *An agent $A = (Ts, En, Re, Ac)$ is a quadruple where*

- *Ts represents its* **topological structure***;*
- *$En = \{\eta_1, \eta_2, \ldots\}$ defines its local* **environment***;*
- *$Re = \{\rho_1, \rho_2, \ldots\}$ is the finite set of incoming* **requests***;*
- *$Ac = \{\alpha_1, \alpha_2, \ldots\}$ is the discrete, finite set of possible* **actions***.*

Ts and *En* represent the passive part of the agent, while *Re* and *Ac* its active part.

Finally, we can introduce the definition of the dimension $\mathcal{D}_{u_i,r_j}$ based on the structural dimension D_{u_i,r_j}.

Definition 4. *A dimension* $\mathcal{D} = (Ts, En, Re, Ac)$ *is an agent where*

- $Ts = D_{u_i,r_j}$;
- $En = \{u_i, r_j, t_1, \ldots, t_n\}$;
- $Re = \{\emptyset\}$, *initially;*
- $Ac = \{add\text{-}tag, delete\text{-}tag, modify\text{-}tag, \ldots\}$

Analogously, we can define new classes of agent dimensions, not only for structural dimensions. New dimensions can be created directly from the user, or computed by the system applying specific metrics, or generated applying ontological models; each dimension can contain other dimensions; each dimension associates a semantics to the set of grouped entities.

Definition 5. *A folksonomy* F *is a multi-agent system formally described as a labeled multigraph of agent entities, organized in semantic contexts, called dimensions.*

$$\mathcal{F} = \bigcup_{i=1}^{n} \mathcal{D}^i$$

All in a folksonomy is a computational agent, equipped with a set of *local variables*, that define its internal state, and a modular and extensible set of *procedural skills*. So, for example, each user is represented in a folksonomy by an user agent: it knows the resources tagged by the user, and the used tags; but it also contains and manages the user profile, and it is able to calculate specific local metrics for her, such as the average number of tags applied on a single resource, the average time spent on a resource, the tagging date, etc. They can further communicate with the other agents present in the personomy and in the folksonomy, such as the tag agents, or the resource agents, or the same dimension agents.

Some semantic connections may be inferred by $\mathcal{F}$, applying opportune metrics, for example a similarity function for identifying neighbors tags, users or resources, or ontological relations.

4 Personalized views and authoring

It is simple recognize in the labeled multigraph contained in the definition of $\mathcal{F}$ the *zz-structures* [17]: they are non-hierarchical, minimalist, scalable structure for storing, linking and manipulating different kind of data. From these structures, we inherit many strengths, such as their intrinsic capability to preserve contextual interconnections among different information, thanks to their particular properties.

The peculiarity of such structures derives from the relation among their component elements: data is stored into *cells*, that may contain very different type of contents, which are connected with links of the same color into linear sequences called *dimensions*. A single series of connected cell among one dimension is called *rank*, while the starting and the ending cells of a rank are called *headcell* and *tailcell*. There is also a restriction according to which for any dimension, each cell can connect almost two other cells following the direction of the dimension.

As discussed in literature [18], zz-structures are used with success in many applications, implemented for different platforms, and due to their flexibility and adaptivity, they have been successful used in several fields, such as bioinformatic, electronic music, e-learning [19], virtual museum tours [20, 21] and so on.

In [22] the authors compare zz-structures with mSpaces and Polyarchies, generating a taxonomy from the graph theory point of view, whereas the work [23], defining a formal model for zz-structure conceived as multigraph graph, proposes different visualizations and a set of navigational information (e.g. such as the distance between the visited cells). Zz-structures can be visualized in different customizable visualizations called *views*, such as H-view, I-view, star-view, m-extended star view, and also *view spaces*, as canvases, projection spaces, presentational fields and viewing tanks [17].

In Figure 3 (left) we show a *H-view*, on two dimensions, $\mathcal{D}_{u_2,t_4}$ and $\mathcal{D}_{u_2,t_1}$, extracted from the folksonomy shown in Figure 2.

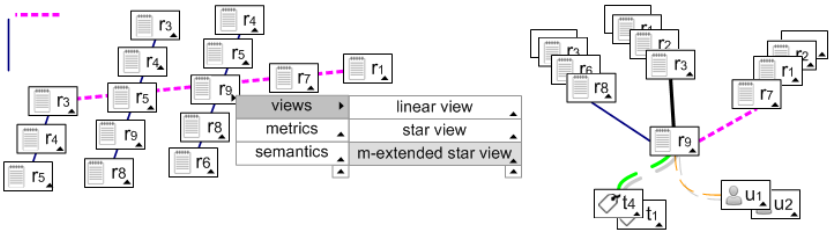


Figure 3. A *H-view* (left) with menu on $r9$; a *5-extended star view on* $r9$ (right)

We note the presence of a black triangle symbol, in two positions, corresponding to selected/not selected: these triangles are associated to scripts related to the session agent of the current visualization, and represent the mean to interact with the cell-agent. When selected, the session agent asks to the chosen resource (r_9, in our example) the set of actions Ac that can be activated on it. In order to satisfy this request, r_9 sends a multicast message to all the dimensions in which it is included, and a run-time created contextual menu, organized in three meta-categories (views, metrics and semantics) is shown. The first category is concerning the different kind of possible views while the other two categories of functions offered by the menu, are related to the computation of an extensible set of *metrics*, and to the application of opportune *semantic relations and ontologies* in order to generate, for example, specific recommendations on content, tag and user.

In our example, the user selects the menu item *views* and then, from its sub-menu, the menu item *m-extended star view* (where $m = 3$). The related 5-extended star view is displayed in Figure 3 (right): we can note that the cell r_9 is connected to the following:

– three labeled edges related to the dimensions $\mathcal{D}_{u_1,t_1}$, $\mathcal{D}_{u_2,t_1}$ $\mathcal{D}_{u_2,t_4}$;

– one labeled edge related to the dimension $\mathcal{D}_{u_i,r_j}$, i.e. the tags (t_4 and t_1) which other users applied on the same resource r_9;
– one labeled edge concerning the dimension $\mathcal{D}_{t_k,r_j}$, i.e. the users (u_1 and u_2) which annotated the resource r_9 with the same tag.

Comparing the two visualizations, it is clear that the *3-extended star view* provides a deep insight of all the dimensions connected to a given focus cell.

Other features, not displayed in Figure 3, regard the possibility to dynamically change, at local or global level, the features of each agent, simply clicking directly on the visualized item and applying modifications. To this extent we can highlight that due to the agent-based technology the folksonomy grows and changes according to the user contributes, and can be shared with the other users.

5 Conclusion and future work

In this paper we have proposed an innovative way to conceive a folksonomy in terms of a multi-agent system, first defining a formal model and then showing Folkview. Such system can be used to simply display personalized user views, to create personalized and adaptive paths for users and to modify the associations between tags and resources.

Up to now we have built a partial, but modular and extensible, prototype, based on a public dataset taken from delicious, and that implements the structural aspects of the considered folksonomy, adding main existing metrics functions, and using both server-side and client technology.

As future work we want to extend the prototype to all the main functionality we discussed, focusing our attention on a semantic personalization. In particular, we plan to make user tests to assess the impact and the effectiveness of the proposed tool, comparing particular user-tasks on our proposal system than what already exists as social tagging system. Furthermore, although we started with a specific dataset, we intend to extend our tool in order to extract data from a large number of social tagging systems.

References

1. Dattolo, A., Ferrara, F., Tasso, C.: The role of tags for recommendation: a survey. In: 3rd IEEE Intern. Conf. on Human System Interaction. (2010) pp. 548–555
2. Li, W.S., Vu, Q., Agrawal, D., Hara, Y., Takano, H.: Powerbookmarks: A system for personalizable web information organization, sharing, and management. Computer Networks **31**(11-16) (1999) 1375–1389
3. Rucker, J., Polanco, M.J.: Siteseer: personalized navigation for the web. Commun. ACM **40** (March 1997) 73–76
4. Keller, R.M., Wolfe, S.R., Chen, J.R., Rabinowitz, J.L., Mathe, N.: A bookmarking service for organizing and sharing urls. Comput. Netw. ISDN Syst. **29** (1997) 1103–1114

5. Sinclair, J., Cardew-Hall, M.: The folksonomy tag cloud: when is it useful? Journal of Information Science **34**(1) (2007) pp. 15–29
6. Hotho, A., Jäschke, R., Schmitz, C., Stumme, G.: Bibsonomy: A social bookmark and publication sharing system. In: Conceptual Structures Tool Interoperability Workshop - 14th Intern. Conf. on Conceptual Structures, Aalborg, Denmark (2006)
7. Cattuto, C., Benz, D., Hotho, A., Stum, G.: Semantic grounding of tag relatedness in social bookmarking systems. In: 7th Intern. Conf. on The Semantic Web. ISWC '08, Berlin, Heidelberg, Springer-Verlag (2008) pp. 615–631
8. Chojnacki, S., Klopotek, M.: Random graph generative model for folksonomy network structure approximation. Procedia Computer Science **1**(1) (2010) pp. 1683 – 1688 ICCS 2010.
9. VanderWal, T.: Folksonomy. http://www.vanderwal.net/folksonomy.html
10. Mathes, A.: Folksonomies - cooperative classification and communication through shared metadata. Computer Mediated Communication - LIS590CMC (2004)
11. Mika, P.: Ontologies are us: A unified model of social networks and semantics. In: Intern. Semantic Web Conference. LNCS, vol. 3729, Springer (2005) pp. 522–536
12. Abel, F., Henze, N., Krause, D.: A novel approach to Social Tagging: GroupMe! In: 4th Int. Conf. on Web Information Systems and Technologies. (2008) 42–49
13. Lambiotte, R., Ausloos, M.: Collaborative tagging as a tripartite network. In: Computational Science. LNCS, vol. 3993, Springer Berlin (2006) pp. 1114–1117
14. Dattolo, A., Eynard, D., Mazzola, L.: An integrated approach to discover tag semantics. In: 26th Annual ACM Symposium on Applied Computing, Web Technologies Track. Taichung, Taiwan. (2011)
15. Hassan-Montero, Y., Herrero-Solana, V.: Visualizious: Visualizing social indexing semantics. http://www.nosolousabilidad.com/hassan/visualizious/ (2007)
16. Klerkx, J., Duval, E.: Visualising social bookmarks. Jodi **10**(2) (2009)
17. Nelson, T.H.: A cosmology for a different computer universe: Data model, mechanisms, virtual machine and visualization infrastructure. Jodi **5**(1) (2004)
18. Dattolo, A., Luccio, F.L.: A state of art survey on zz-structures. In: 1st Workshop on New Forms of Xanalogical Storage and Function. CEUR:508 (2009) pp. 1–6
19. Dattolo, A., Luccio, F.L.: Formalizing a model to represent and visualize concept spaces in e-learning environments. In: 4th Webist International Conference - WEBIST08. (May 2004) 339–346
20. Dattolo, A., Luccio, F.: Visualizing personalized views in virtual museum tours. In: International Conference on Human System Interaction- HSI08. (May 2008) 109–114
21. Dattolo, A., Luccio, F.: A formal model for supporting the adaptive access to virtual museums. In Hippe, Z., Kulikowski, J., eds.: Human-Computer Systems Interaction. Volume 60 of Advances in Soft Computing. Springer Berlin - Heidelberg (2009) 481–492
22. McGuffin, M., Schraefel, M.: A comparison of hyperstructures: Zzstructures, mspaces, and polyarchies. In: 15th ACM Conference on Hypertext and Hypermedia - HT'04. (August 2004) 153–163
23. Dattolo, A., Luccio, F.L.: A formal description of zz-structures. In: 1st Workshop on New Forms of Xanalogical Storage and Function. CEUR:508 (2009) pp. 7–11

Selective Propagation of Social Data in Decentralized Online Social Network

Udeep Tandukar and Julita Vassileva

Department of Computer Science, University of Saskatchewan, Saskatoon, Canada
udeep.tandukar@usask.ca, jiv@cs.usask.ca

Abstract. In Online Social Networks (OSNs) users are overwhelmed with huge amount of social data, most of which are irrelevant to their interest. Due to the fact that most current OSNs are centralized, people are forced to share their data with the site, in order to be able to share it with their friends, and thus they lose control over it. Decentralized OSNs provide an alternative which allows users to maintain control over their data. This paper discusses an approach for propagation of social data in a decentralized OSN so as to reduce irrelevant data among users. The approach uses interaction between users to construct relationship model of interest. This relationship model acts as a filter later while propagating social data of the same interest group. This paper also presents a plan of a simulation to analyze our approach.

Keywords: Online Social Network, Decentralization, Peer-to-peer system, Information propagation, Relationship modeling

1 Introduction

Online Social Networks (OSNs) have become a common ground where people are generating and consuming huge amount of information. This information varies from personal thoughts (like status updates) to global news (such as wars, world cup, etc.). There have been growth in the number of providers of such OSNs and user's data are scattered over these different providers. In OSNs there is a huge flow of information of which only a fraction is relevant to the users. Since decentralized OSNs also inherit most of the issues from OSN, they have to deal with the question of how to provide relevant information to users and filter out the irrelevant information.

The currently popular OSNs are centralized which means they store all the information that people are generating or consuming. This information is mostly private and people voluntarily share it with the site, in order to be able to share it with their friends. However, in this way, they have less control over their own data and their data is scattered over the internet in different OSN providers which usually do not support data interoperability (apart from trivial user profile information).

This issue of control and privacy in centralized OSNs has recently motivated research into decentralized OSNs where the data of users are kept in their own clients (nodes). There have been several attempts to build decentralized OSNs and these

projects are still going on [3], [7], [15]. Among several approaches to build decentralized OSN, a peer-to-peer (P2P) infrastructure has also been proposed. P2P has been popular among file-sharing applications, but not in OSNs. The inherent nature of how people connect with each other in a social network makes peer-to-peer architectures suitable for building decentralized OSNs.

This paper proposes an approach to deal with both problems mentioned above – information overload and privacy – using a P2P infrastructure and relationship models according to interest groups among users to filter out irrelevant information from flowing out of the source. These relationship models are updated depending on feedback resulting from the interaction between users and what they do with the information they receive. The relationship models are used later to route the information that a user sends to her friends appropriately according to its semantic meaning.

The rest of the paper is structured as follows. A review of related works is presented in Section 2, followed by definition of our research problem in Section 3. Section 4 discusses our approach on filtering out irrelevant social data using relationship models. The proposed plan of simulation and evaluation of the discussed approach is presented in Section 5. Finally, Section 6 concludes the paper.

2 Background and Motivation

The section begins with an overview of online social networks and decentralized online social networks as an alternative to centralized ones. Then there is discussion about using peer-to-peer architecture for decentralized online social networks. Then we mention about information dissemination in social networks, and user modeling which is related to relationship modeling in our work. And the problem statement of our research is covered at the end of this section.

2.1 Online Social Networks

An Online Social Network (OSN) is defined as web platform in which a person can create a profile, connect to other people, view and traverse network of connections within the system, share resources and information within the system, and use social applications with which people within the system can interact and collaborate with each other [5], [9]. With the growth of internet usage, OSNs first came into existence in the form of SixDegree.com (in 1997) which had basic OSN features [5]. In the following years many more OSN service providers (like Friendster, MySpace, Last.FM, Hi5, Twitter, Facebook, etc.) came into existence, some of which have grown to have millions of active users. All these OSN followed client-server architecture in which the service provider is centralized. This architecture supports high accessibility since users can access the service from any web-browser wherever and whenever they desire. But due to this centralized nature, these OSN have inherent issues like single point of failure, central administration that can control activities of users, privacy issues due to central data storage, and requirement of larger servers and

bandwidth to accommodate the growing number of users. With these issues in centralized OSN we saw discontinuation of some popular OSN services like SixDegree.com and Friendster [5], and the users lost all their social contacts and data when they lost access to those services. Some centralized OSN services like Twitter, due to growth in their popularity were having many performance scalability issues and still face some frequently slow response or even unresponsiveness [8]. In addition to these technical issues, there are also social issues arising with the rapidly growing popularity of social networking. People became more conscious about the information that they share in their social networks. Services, like Facebook which have millions of users, frequently have to deal with privacy issues. For example, Facebook's Beacon online ad system was tracking activities of the users in third party websites even when users were logged off from Facebook and had declined to broadcast their activities [14]. The system caused an outrage among Facebook users, and Facebook quickly discontinued Beacon. However, present centralized OSN still have full control over user data once shared, the user losses control over it and cannot remove it or export it into another OSN. Although there are various OSN available in web, there is no easy interoperability across them. Users of OSN (e.g. MySpace) cannot interact with users of another OSN (e.g. Facebook). This has lead OSNs being viewed as "information silos" [23]. As alternative to this centralized OSN, we can consider decentralized OSN in which users have control over their data.

2.2 Decentralizing Online Social Networks

Decentralized online social networks have distributed computing structure with trusted network of servers or peer-to-peer network. In [23], the authors suggest that decentralized OSN will give back to users the control of their data with respect to privacy, data ownership and information dissemination.

Users hosting their own social data. According to Yeung et.al [23], in decentralized OSN the user is not required to be a part of social networking services like Facebook, Twitter, etc. to maintain his/her online social presence. The user can host a FOAF [6] (Friend-Of-A-Friend) file, an activity log, photos/videos, and social client in a trusted server. They will have full control over whom and what to share out of his/her social data. The authors describe how the functionality offered by popular social applications, such as "Personal Wall", "Photos", and "News Feed" can be implemented in decentralized OSN. In the proposed system, the user shares and communicates social data with other users by using WebDAV [22] or SPARQL Update [18] protocols. As a prototype they have developed "Tabulator" [3] which is a generic data browser and editor of linked RDF (Resource Description Framework) data [4]. These types of decentralized OSN encourage users to store their social data on the web in standard format such as RDF and it should be accessible through URI (Universal Resource Indicator). Therefore, the user does not have to rely on only one social application, and can use any social application that support these open technologies.

Using P2P infrastructure. Decentralized OSN can also be implemented with the use of Peer-to-Peer (P2P) network. A P2P network is a distributed network in which nodes are connected with each other to participate in processing, memory, and bandwidth intensive tasks. These networks scale better than centralized server architectures without the need of costly centralized resources. P2P networks have been popular mostly as file sharing networks (such as KaZaA, BitTorrent, etc.) and sometimes as collaborative sharing networks (such as Skype), but have not been used as a medium for online social networking. The inherent nature of peer-to-peer connection between users in a social network makes OSN suitable for peer-to-peer architecture [9]. In file sharing P2P systems like Gnutella, most of the users are free riders [1]. In contrast, P2P applications like Skype, where users tend to stay connected to the network to receive calls from their friends, shows the potential that P2P holds as implementation infrastructure for OSN.

To accommodate the familiar functionality of centralized OSN (like status updates, photo uploads, commenting, rating) in decentralized OSN, there are various challenges. Since the social data is stored at the peers, the availability of the social data depends on the online behaviour of peers. The storing of data on the peers allows encryption of these data which can ensure privacy while transmitting data from peer to another. The propagation of social data or updates among users in the OSN should be managed so that there is less duplication and no latency. These and other challenges have been discussed in detail in [9].

Some P2P systems have exploited properties of social networks (like trust, collaboration) in other to improve the performance of P2P networks. System like Tribler [15], has adopted social networking on BitTorrent based P2P file-sharing network in order to recommend, search, and download contents. The authors have used “Buddycast Algorithm” that exchanges preferences of peers in the implicitly defined social network to generate recommendation list and search contents. By using a collaborative download protocol called “2Fast”, in which users collaborate to contribute their bandwidth, download performance also improved. Some also used the graph topology of real social networks in order to form a P2P overlay topology and used it to improve lookups and scalability [2].

PeerSoN [7] is an effort to build decentralized OSN over theP2P architecture. It has implemented encryption of user data to protect privacy and ensure direct exchange of data between devices for delay-tolerance and opportunistic networking. It uses OpenDHT [17], a Distributed Hash Table (DHT) service, for look-up service to find other peers in the P2P network and also to store data like, IP address, file information, and notifications for peers. In DHT [16], {key, value} pairs are stored in a distributed nodes and node can retrieve the value associated with any key efficiently. The prototype of PeerSoN provides functionality like social links (becoming friends), storage to maintain profile and post by their friends, asynchronous messaging, and live chatting.

2.3 Information dissemination

Nowadays, OSN produce huge amount of information and propagation of this information to its destination has to be well coordinated so as to reduce information overload, duplication, latency and to ensure quality.

In a file-sharing P2P network, the location of a resource is a very important piece of information. Generally, users send queries about resources of interest and system returns lists of their locations (i.e. peers that store these resources). In contrast to this process of searching, [13] discuss "selective information push" where user posts her profile to "super-peers" and receives notifications about resources that match her interests as these resources become available. Since the user is both consumer and producer of the resources shared in the network, she can also post advertisements of her resources and the super-peer will push notifications about these resources to relevant peers (peers with matching interests). This mechanism depends on the preferences of the user querying super-peers and if user has a more general interest then she might get lots of notifications. Here, the super-peer can be taken as a recommender that is pushing information according to the peer's interest.

The Push-poll recommender algorithm [20] propagates information through implicit social network, formed by peers with similar interests, using "word of mouth" mechanism. This system also takes in account feedback from the recipient to determine the future influence of sender on recipient. KeepUp recommender system [21] is based on the Push-poll algorithm. It allows user to interactively adjust the amount of influence that her neighbours have on the recommendations she receives. This gives power to user to decide indirectly what and how much information is propagated to her.

GoDisco [10] focuses on dissemination of social data according to the context of the information. The nodes gossip about their interests and strength of these interests with their neighbours in a regular interval. They also keep track of the behaviour of their neighbours (like activeness, forwarding behaviour). This knowledge of each other is used in the dissemination phase where the messages are assumed to have some semantic value that can be mapped to the interests of the nodes. Our work will consider the degree of relationship between these nodes and influence of this on dissemination. We will be creating relationship model of neighbouring nodes to determine degree of relationship.

2.4 User Modeling

Each user has her own characteristics, e.g. interests, preferences, etc., and we can utilize it to provide her with relevant social data in social network. These characteristics comprise the user model for the system. In our work, we will be building user models of the neighbours of a user within particular domain of interest in order to evaluate which information to forward to them or not.

Since the user will be using her OSN data on different devices and possibly across different applications, interoperability of the user models along with other social data is very important. That is why it is desirable for the representation of user data in the

user model to follow some ontology so that it could be understood and interpreted outside of the context of the application in which the model was created. Using RDF-based user model like UserML [12] can enable distribution of the user model among different devices. UserML divides user model dimension in three parts: auxiliary, predicate and range. If we want to express the interest of a user in UserML then auxiliary will be "hasInterest", predicate will be "reading" to indicate her interest and range can be "low-medium-high".

To stimulate cooperation while sharing resources in P2P system, Sun et.al [19] has applied user modeling and modeling of relationships between users. With the help of user models of interest, they were able to route information to other users with similar interests. Using relationship modeling between users, they were able to determine typical time patterns of neighbour's behaviours to ensure better quality service. The authors created an overlay topology over the P2P network, where a relationship between users is created when a user successfully downloads a file from another user and the strength of the relationship grows with the number of successful interactions between these users.

With the increasing interest of users in social networking on web, there has been significant growth in research related to OSN. Users are becoming more and more sensitive to their data and decentralized OSN holds a key for these users to use web with full control over their data. As discussed earlier for decentralized OSN users can either choose secure server to host their data or use P2P infrastructure. As a new domain for P2P infrastructure, OSN holds lots of challenges to the researchers. As OSN is based on sharing of information, well-coordinated propagation of information is very important to handle information overload, latency, and repetitions. In our work, we will be focusing on using models of interest of neighbouring users for proper propagation of information.

2.5 Problem Statement

Online social network (OSN) has provided a medium for people to communicate and share information (social data). People share their thoughts, photos, videos, links to web pages, etc. in this network. The network in OSN constitute of members that are interconnected with each other through some relationship like friendship, common preferences, etc. When shared to the network, the shared social data propagate to each and every member of the network whether it is relevant to them or not. From the viewpoint of a sender, she is sharing only one social data at a moment. But from the viewpoint of a receiver, there can be more than one sender. According to statistics from Facebook [11], on average users have 130 friends. If all of the friends share a social data then a user will get 130 different social data. In this way, the user's stream is flooded with huge amount of social data, most of which are irrelevant to the user's interest.

Most of the available OSNs are based on client-server architecture in which user's social data are kept centralized. As discussed earlier this centralized nature has some issues and as an alternative we can have OSN in decentralized architecture, where users have control over their own social data. Even in decentralized OSN, due to the

social nature of the network we will have to deal with propagation of social data to reduce irrelevancy, redundancy, and latency. The research domain in this paper is related to propagation of social data from a user to her neighbours so that they only get data relevant to them.

3 Approach

An approach of selective propagation of social data (i.e. information shared by users in social networks, such as status updates, shared links) by modeling interest of neighbouring users in a social network is proposed next, that ensures that social data reaches only the relevant users for whom it would be interesting.

3.1 Social P2P Network of Users

The system is a decentralized online social network implemented over P2P network. For simplicity, we will be dealing with a social network which can be represented by social graph. Social graph is a graph in which each user is a node and relationships between users is edges. Let G be a social graph represent by $\{N, E\}$ where N represents set of nodes (users) and E represents set of edges (relationships) between nodes. We can say $n_a \in N$ have some relationship with $n_b \in N$ *iff* there exists $\{n_a, n_b\}$ *or* $\{n_b, n_a\} \in E$.

To route relevant social data to users, each user or node in the graph will model the interests of other users with whom she has relationships. From the point of view of a given user, the model of interests of other users is considered as relationship model since it signifies how many positive interactions have happened between the users in the context of particular area of interest. Positive interaction between two users in a given area of interest means that one user has sent social data related to the area of interest to the second user, and the second user has given positive feedback after receiving the social data. As a result of positive interaction, the strength of the relationship between the two users in the area of interest increases. The relationship model is used by the peer to adaptively disseminate social data related to a given interest area *I*, by sending it to peers with whom the user has sufficiently strong relationship in area *I*.

3.2 Relationship modeling

In an online world, relationships between users strengthen as the interaction between them increases. In the proposed approach, not only interaction between users in general but within certain subject or interest is taken into account, so that the system can model the strength of relationship in an area of specified interest between interacting users. To determine the area of interest of the social data, users have to either tag their updates with the interest areas or the system has to extract semantics from the data.

Interaction between users within certain context is captured by tracking the feedback of the shared content. Feedback from friends (users connected in social graph) can be of different types. For the proposed system, feedback is categorized as follows.

Table 1: Categorization of feedback

Type	Action	Value
Type 1	Comment / Share	0.9
Type 2	Rate / Like	0.7
Type 3	View / Open	0.5
Type 4	Ignored / Not open	0.3

The response value varies from 0.3 to 0.9 according to the type of action the users takes. These feedbacks from the receiver of the social data depend on the level of interest and relationship model with the sender for that context. The relationship model depends on the previous interaction between two users, and priority of a new social data is determined according to previous history of interaction in the system.

The relationship model consists of a list of areas of interest and the corresponding strength of relationship between two users in each of these areas. Strength of relationship between user A and user B for an interest area I should increase with stronger feedback and decrease with weaker feedback, therefore it is calculated using following equation:

$$S_A^B(I) = \alpha * S_A^B(I)_P + (1 - \alpha) * F\,. \qquad (1)$$

Here, $S_A^B(I)$ is the new strength of relationship, $S_A^B(I)_P$ is the previous strength of relationship for an interest area X. The parameter $\alpha \in [0, 1]$ is a linear function of the number of social data produced by the user in particular interest area X. Initially α is 0.9 so that the latter half of the equation has very low effect on the new strength. The feedback from the recipient is denoted by F, and its value varies from 0.3 to 0.9 as specified in Table 1. The increase and decrease of the strength of relationship calculated according to equation (1) is at very minimal rate, so as to maintain the relationship between the users as long as possible.

For the propagation of social data belonging to interest I, the strength of relationship between users should be more than a threshold value. Initially, this strength of relationships among all users is set as 1 and it will increase and decrease according to the interactions between users.

This approach of propagating social data takes into account the feedback of friends and uses this feedback to calculate strength of relationship, which is used in the future as a filter while sending the data of similar topic. It is taken in consideration that if information is relevant to a user, she will at least open that message and the feedback value is 0.5 which is more than critical value. If the information is irrelevant to a user, she will ignore it and hence the feedback value is 0.3 which is less than critical value and reduces the strength of relationship. This relationship models are all stored in the user's device since our system follows decentralized architecture. The process of filtration during propagation is done at the sender; therefore, some computation power

is consumed at the sender side but network traffic is reduced and the friend's node does not have to do much of the filtration process.

As the strength of relationship for a particular *interest I* in *user A* for *user B* fades away, *user B* will not get any social data related to *interest I* from *user A*. For *user B* to get social data related to *interest I* from *user A*, she has to make relationship model in *user A* between her and *user A* stronger. *User B* can send a social data to *user A* related to *interest I*, this will show that *user B* is becoming interested in *I* and *user A* will increase the strength of relationship as high as possible so that social data from her can reach *user B*. To give more control to users over their relationships, it is also possible to allow users to directly adjust the relationship strength with other users via an appropriate GUI, similar to the interactive influence adjustment deployed in the KeepUP Recommender System [21].

In this way, relationships between users will grow or fade away in context to certain interest groups. This will allow better communication between users since they do not have to deal with irrelevant social data.

4 Evaluation Plan

In order to evaluate the discussed approach of using relationship modeling to filter out irrelevant social data in a social network, a simulation of the system will be developed using synthetically generated social graph and real-world social graph from StudiVZ and Facebook. A random social graph with small world properties will be generated using JUNG[1] (Java Universal Network/Graph) Framework as a synthetic dataset. Afterwards, two different real datasets will be used to generate the network and message streams – one from StudiVZ[2] and one from Facebook[3]. Both have around 1 million users or nodes.

4.1 Distribution of interest

Possible areas of interest can be defined for users in hierarchical way by introducing general categories and sub-categories, so initially, to avoid widely separated interests in population the system will only have one level of general categories of interest, such as "sports", "news & events", "politics", "personal status updates", "photos", "videos", "curiosities & jokes". Interests are distributed exponentially over users in the social network with most common interests (in the currently most popular music, movies, etc.) taking large portion of population and less common interest (e.g. local sport) popular among small portion. The mechanisms to generate such skewed distributions are known: growth – people gain new interests with time, and preferential attachment – areas of interest that are already popular attract newcomers with a higher likelihood. The simulation will use these rules to

[1]http://jung.sourceforge.net/index.html
[2]http://studivz.irgendwo.org/
[3]http://odysseas.calit2.uci.edu/doku.php/public:online_social_networks#available_datasets

generate a realistic distribution of interests for a fixed set of interest semantic categories.

The system depends on the growth of relationship strength between users for particular interest. The feedback of each shared data is important to calculate this strength. This feedback depends on the interest level of the receiver of the social data. The system will simulate around 25 interest categories and these will be distributed to all of the nodes in the graph. The nodes will have different interest levels, $I \in [0, 1]$, which signifies how much the user is interested on each category. When the users receives an update of an interest (semantic) category, the higher the interest level, the likelihood of feedback from Type 1 to 3 as illustrated in Table 1 increases by the users. The users who have lower interest levels in the category of the update will give Type 4 feedback. The distribution of the interest levels initially will be random. Since there will be some nodes which will have more connection that other nodes, there is probability of these nodes being interested in more interest groups. With the interaction on particular area, the value of interest level will also grow. These considerations will be taken into account while designing the proposed simulation.

4.2 Propagation of social data

The propagation of social data depends upon the relationship model of each simulated user. Initially the system will consist of equally distributed relationship model (equal value of relationship strength) so that propagation of social data at the initial stage of the system reaches all friends of the user. With the feedback from friends, these relationship models will either strengthen or weaken according to equation (1). The likelihood of type of feedback as illustrated in Table 1 depends on the interest level of the friends as discussed in earlier section. For simplicity, each social data which will propagate will also carry semantic meaning along with it. The semantic meaning will consist of types of interest group the social data belongs to. To simulate the phenomenon of users injecting new content in the system, each node in the system is fed with a number of social data within random interval of time. The behaviour of each node whether to forward (share) an incoming social data to its neighbour (friend) depends on the interest level, strength of the relationship with respect to the interest category of the social data represented in the relationship model. All these changes of relationship model, forwarding of data, filtering of irrelevant data and interest level will be recorded and used in future analysis.

4.3 Analysis

For each node, the following data will be recorded in the simulation:

- Relationship model established between each node with its friends.
- Interest level of each node.
- The number of social data a node shared with its friends.
- The number of social data filtered by relationship model in each node.
- The number of social data that is forwarded by a node.

- The total number of nodes that forwarded social data received from its friends and interest level at the moment of forwarding.

With the simulation of our proposed system, we hope to have the following analysis and insights.

- New social data generated by a user will get filtered and the propagation will be limited by the relationship model. The rate at which the nodes evolve to filter out irrelevant social data will be analyzed.
- The level of interest in a user has direct impact on the type of feedback to a shared data. This feedback is used to calculate the relationship strength. The level of interest will change as the number of interaction increases or decreases. The correlation between interest level and relationship strength will be analyzed.
- A node can forward an incoming social data. This behaviour largely depends on the interest level. The system will record this forwarding behaviour of each node to analyze range of propagation of a social data. Range of propagation means how far a social data is forwarded from its source.
- Spreading or sharing of social data will largely depend on the interest level and relationship model of each node in the system. Sharing of a social data stops where node has low interest level or weak relationship model with her friends.

When the system reaches its maturity, it will have nodes interacting with each other without concerns about getting irrelevant social data.

5 Conclusion

In this paper, we have discussed an approach of using feedback from interaction between users as a relationship building mechanism to filter out irrelevant social data in decentralized online social networks. Decentralized online social networks give users the control of their social data with respect to privacy, data ownership and information dissemination. A simulation will be implemented to analyze the discussed approach. The simulation will also deal with the numbers of hops a social data transverse so as to analyze its spread. The simulation will have nodes which are always online and cooperative. But in the real system, there are always issues with availability, free riders, latency, and other factors. The area of decentralized online social networks holds exciting research questions, associated with storage of social data, privacy issues of social data, searching and indexing of friends in the network, and many more. We plan to implement a real decentralized OSN that follows P2P architecture after the successful analysis of the simulation and evaluation.

References

1. Adar, E. and Huberman, B.A.: Free Riding on Gnutella. First Monday 5 (10), 2-13 (2000)
2. Anwar, Z., Yurcik, W., Pandey, V., Shankar, A., Gupta, I., and Campbell, R.H.: Leveraging Social-Network Infrastructure to Improve Peer-to-Peer Overlay Performance: Results from Orkut. ACM Computing Research Repository (2005)

3. Berners-Lee, T., Hollenbach, J., Lu, K., Presbrey, J., and Schraefel, M.: Tabulator redux: Browsing and writing linked data. In Proceedings of the 1st Workshop on Linked Data on the Web (2008)
4. Berners-Lee, T.: Linked Data, http://www.w3.org/DesignIssues/LinkedData.html (2006)
5. boyd, d and Ellison, N.B.: Social Network Sites: Definition, History, and Scholarship. Journal of Computer-Mediated Communication 13 (1), 210-230 (2008)
6. Brickley, D. and Miller, L.: FOAF Vocabulary Specification 0.98, http://xmlns.com/foaf/spec/ (2010)
7. Buchegger, S., Schiöberg, D., Vu, L.H., and Datta, A.: PeerSoN: P2P social networking: early experiences and insights. In Proceedings of the Second ACM EuroSys Workshop on Social Network Systems, ACM, 46–52 (2009)
8. Cozzatti, J.P.: A Perfect Storm of Whales, http://engineering.twitter.com/2010/06/perfect-stormof-whales.html (2010)
9. Datta, A., Buchegger, S., Vu, L.H., Strufe, T., and Rzadca, K.: Decentralized Online Social Networks. In B. Furht, ed., Handbook of Social Network Technologies and Applications, Springer, 349–378 (2010)
10. Datta, A. and Sharma, R.: GoDisco: Selective Gossip Based Dissemination of Information in Social Community Based Overlays. Distributed Computing and Networking, 227–238 (2011)
11. Facebook: Statistics, http://www.facebook.com/press/info.php?statistics (2010)
12. Heckmann, D. and Krueger, A.: A user modeling markup language (UserML) for ubiquitous computing. Lecture Notes in Artificial Intelligence 2702, 393-397 (2003)
13. Koubarakis, M., Tryfonopoulos, C., Idreos, S., and Drougas, Y.: Selective Information Dissemination in P2P Networks: Problems and Solutions. ACM SIGMOD Record 32 (3), 71-76 (2003)
14. Perez, J.C.: Facebook's Beacon More Intrusive Than Previously Thought, http://www.pcworld.com/article/140182/facebooks_beacon_more_intrusive_than_previously_thought.html (2007)
15. Pouwelse, J.A., Garbacki, P., Wang, J., et al.: TRIBLER: a social-based peer-to-peer system. Concurrency and Computation: Practice and Experience 20 (2), 127–138 (2008)
16. Ratnasamy, S., Francis, P., Handley, M., Karp, R., and Shenker, S. A scalable content-addressable network. In Proceedings of conference on Applications, technologies, architectures, and protocols for computer communications, ACM, 161–172 (2001)
17. Rhea, S., Godfrey, B., Karp, B., et al.: OpenDHT: A public DHT service and its uses. ACM SIGCOMM Computer Communication Review 35 (4), 73–84 (2005)
18. Seaborne, A., Manjunath, G., Bizer, C., et al.: SPARQL Update, http://www.w3.org/Submission/SPARQL-Update/ (2008)
19. Sun, L., Upadrashta, Y., and Vassileva, J.: Ensuring quality of service in p2p file sharing through user and relationship modelling. In Proceedings of User Modelling UM03 Workshop on User and Group Models for Web-Based Adaptive Collaborative Environments, Johnstown, 57-66 (2003)
20. Webster, A. and Vassileva, J.: Push-poll recommender system: Supporting word of mouth. In Proceedings of User Modelling, UM2007, Corfu, Greece, Springer, 278–287 (2007)
21. Webster, A. and Vassileva, J.: The KeepUP recommender system. Proceedings of the 2007 ACM conference on Recommender systems, ACM, 173–176 (2007)
22. Whitehead Jr, E.J. and Goland, Y.Y.: WebDAV: A network protocol for remote collaborative authoring on the Web. In Proceedings of the sixth conference on European Conference on Computer Supported Cooperative Work, 291–310 (1999)
23. Yeung, C.A., Liccardi, I., Lu, K., Seneviratne, O., and Berners-Lee, T.: Decentralization: The future of online social networking. In W3C Workshop on the Future of Social Networking Position Papers (2009)

Towards a Followee Recommender System for Information Seeking Users in Twitter

Marcelo G. Armentano, Daniela Godoy and Analía Amandi

ISISTAN Research Institute, Fac. Cs. Exactas, UNCPBA
Campus Universitario, Paraje Arroyo Seco, Tandil, 7000, Argentina
CONICET, Consejo Nacional de Investigaciones Científicas y Técnicas, Argentina
{marmenta, dgodoy, amandi}@exa.unicen.edu.ar

Abstract. Micro-blogging activity taking place in sites such as Twitter gains everyday more importance as a source of real-time information and news spreading medium. Finding relevant information sources among the increasing number of Twitter members is essential for users needing to cope with real-time information. In this paper we study Twitter aiming at generating a set of recommendations to a target user consisting in people who publish tweets that might be interesting to him/her. We evaluate and compare two recommendation approaches: the first selects a set of candidate recommendations using only the network topology and the second exploits the user-generated content available in their tweets. We report the results of a set of controlled experiments with real users carried out to evaluate and compare the performance of both algorithms.

Keywords: Recommender Systems; Micro-blogging Activity; Online Social Networks

1 Introduction

Twitter is a social networking site that has been selected for many users as a means of disseminating (and reading) news and information. There are three main factors for choosing Twitter for this goal. First, unlike many other online social networks such as Facebook, Hi5, Orkut, LinkedIn or MySpace, connections in Twitter are unidirectional. This means that a user decides to "follow" other users with no need of this relation to be accepted or reciprocated. Second, the 140-characters length restriction applied to the messages that users can post in Twitter (which are called tweets) enable users to receive their followees updates in almost any mobile device or to quickly read a bunch of them directly on the Internet or within a desktop application. Finally, any user can easily "retweet" another user's post. In this way the information will be spread out from the author followers to other users' followers. Kwak et al. [11] identified that 77.9% of Twitter's connections are unidirectional and only 22.1% of the relations are reciprocate. Moreover, 67.6% of users are not followed by any of their followees, indicating that these users probably use Twitter as a source of information rather than to keep in touch with friends. Finally, Kwak et al. found that retweets collectively determine the importance of the original tweet expressing a form of collective intelligence. All these

facts, in addition to the great explosion in the number of registered users in Twitter[1], make us believe that information-seeking users would benefit from a recommender system able to suggest information sources that they might be interest in following.

In this work we study Twitter from a user modeling perspective. Our goal is to provide recommendations to information seekers about users that publish tweets that might be of their interest. In order to be valuable, the recommended followees should be in the category of information broadcasters, since these users will probably generate content that the target user may be interested in reading.

Unlike traditional recommendation systems, we do not have any explicit information available about the user's interests in the form of ratings on items he/she likes or dislikes. For profiling a Twitter user the structure of the followers/followees network and the tweets published in this network is the only information available. Both are considered in this paper as a means to recommend people either belonging to the user's neighborhood or sharing content-related interests.

In this article we present two recommendation algorithms using two different techniques: a collaborative filtering technique [16] and a content based technique [14]. The first algorithm is based only on the topology of Twitter network. It first explores the connections starting at the target user (the user to whom we wish to recommend new followees) in order to select a set of candidate recommendations and finally it ranks those candidates according to a scoring function. The scoring function we design involves three factors that take into account the most influential properties of the Twitter network, according to previous studies. The second algorithm first creates a vector of terms describing the interests of the target user based on the tweets published by his/her followees. This vector is then used to discover new users that might not belong to the target user neighborhood in spite of being similar to him/her. Since these users are not taken from the connections starting at the target user, in principle they would not be discovered by the topology-based algorithm.

Unlike other works that focus on ranking users according to their influence in the entire network [18,19], the algorithm we propose explores the follower/following relationships of the user up to a certain level, so that more personalized factors are considered in the selection of candidates for recommendation, such as the number of mentions of these candidates. Furthermore, the approach proposed in this work was evaluated with a controlled experiment with real users. From the experiments performed we found that although the average precision tend to be similar for both algorithms, the content-based approach is better at positioning relevant recommendations at the top of the ranking.

The rest of this work is organized as follows. Section 2 describes some aspects about Twitter and discuss how related work is related to our research. Next, in Section 3 we describe our approach to the problem of followee recommendation in Twitter. In Section 4 we present the experiments we performed to validate our proposal. Finally, in Section 5 we discuss the results we obtained and present our conclusions and future work.

[1] In 2010 Twitter grew by more than 100 million registered accounts. http://yearinreview.twitter.com/whosnew/. Accessed on April 2011

2 Background and Related Work

Twitter is a social network with micro blogging service that enables users to send and receive messages with a length shorter than 140 characters that are called "tweets" or status updates. Relationships in Twitter are unidirectional: a Twitter user *U* interested in the tweets published by another user registers himself as a "follower". Although user *U* has no need to follow their followers back, it is possible for him/her to obtain the list of users following him/her.

As stated above, the Twitter network is populated with tweets. Tweets can have any (textual) content; however there exist users that only publish tweets about a particular subject, such as sports, movies, music or a about a particular rock band. These users can be considered as information sources or broadcasters. In contrast, many people uses Twitter to get information on particular subject, as a form of RSS reader, registering themselves as followers of their favorite artists, celebrities, bloggers, or TV programs. For this last type of users finding high quality and reliable information sources in the constantly increasing Twitter community becomes a challenging issue.

Several recent research efforts have been dedicated to understand micro-blogging as a novel form of communication and news spreading medium. Java et al. [8] and Krishnamurthy et al. [10] presented a characterization of Twitter users grouping them into three categories. The first correspond to *information sources* or *broadcasters*, which are users that are characterized by having a much larger number of followers than they themselves are following. The second category groups *information seekers*, users who rarely post a tweet authored by themselves but that regularly follows other users. Finally, users categorized as *friends* or *acquaintances* are users that tend to use Twitter as a typical on-line social network and are characterized by reciprocity in their relationships.

The influence of users in Twitter has also been subject of several studies. In [11] it was shown that ranking users by the number of followers and by their PageRank give similar results. However, ranking users by the number of re-tweets indicates a gap between influence inferred from the number of followers and that from the popularity of users' tweets. Coincidentally, a comparison between in-degree, re-tweets and mentions as influence indicators [1] concluded that the first is more related to user popularity. Analyzing spawning re-tweets and mentions, it was found that most influential users hold significant influence over a variety of topics but this influence is gained only through a concentrated effort (such as limiting tweets to a single topic). TwitterRank [18], an extension of PageRank algorithm, tries to find influential twitterers by taking into account the topical similarity between users as well as the link structure. Garcia et al. [5] propose a method to weight popularity and activity of links for ranking users. User recommendation, however, can not be based exclusively on general influence rankings since people get connected for multiple reasons.

While the mentioned studies focus on analyzing micro-blogging usage, other works try to capitalize the massive amount of user-generated content as a novel source of preference and profiling information for recommendation. Chen et al. [3] proposed an approach to recommend interesting URLs coming from information streams such as tweets based on two topic interest models of the target user and a social voting mechanism so that the most popular URLs within the group are recommended. *Buzzer* [15]

indexes tweets and recent news appearing in user specified feeds, which are considered as examples of user preferences, to be matched against tweets from the public timeline or from the user Twitter friends for story ranking and recommendation. Esparza et al. [4] address the problem of using real-time opinions of movie fans expressed through the Twitter-like short textual reviews for recommendation. This work assumes that tweets carry on preference-like information that can be used in content-based and collaborative filtering recommendation.

In contrast to the previous works that address the problem of suggesting potentially relevant content from micro-blogging services, we concentrate in recommending interesting people to follow. In this direction, Sun et al. [17] proposes a diffusion-based micro-blogging recommendation framework which identifies a small number of users playing the role of news reporters and recommends them to information seekers during emergency events. Closest to our work are the algorithms for recommending followees in Twitter evaluated and compared using a subset of users in [7]. Multiple profiling strategies were considered according to how users are represented in a content-based approach, a collaborative filtering approach and two hybrid approaches. User profiles are indexed and recommendations generated using a search engine, receiving a ranked-list of relevant Twitter users based on a target user profile or a specific set of query terms. Our work differs from this approach in that we do not require indexing profiles from Twitter users, instead topology-based and content-based algorithms explored the follower/followee network in order to find candidate users to recommend. Furthermore, we consider in the evaluation of our approach the target user assessment about the his/her interest in the provided recommendations.

Finally, the problem of helping users to find and to connect with people on-line to take advantage of their friend relationships has been also studied in the context of social networks. For example, SONAR [6] recommends related people in the context of enterprises by aggregating information about relationships as reflected in different sources within a organization, such as organizational chart relationships, co-authorship of papers, patents, projects and others. [12] presented different methods for link prediction based on node neighborhoods and on the ensemble of all paths. These methods were evaluated using co-authorship networks. Authors found that there is indeed useful information contained in the network topology alone. Chen et al. [2] compared relationship-based and content-based algorithms in making people recommendations, finding that the first ones are better at finding known contacts whereas the second ones are stronger at discovering new friends. Weighted minimum-message ratio (WMR) [13] is a graph-based algorithm which generates a personalized list of friends in a social network built according to the observed interaction among members. Unlike these algorithms that gathered social networks in enclosed domains from structured data (such as interactions, co-authorship relations, etc.), we proposed two algorithms to take advantage of the massive, unstructured, dynamic and inherently noisy user-generated content from Twitter.

3 Followees Recommendations on Twitter

We have designed two different algorithms for followee recommendation on Twitter. The first algorithm is only based on the topology of the followers/followees network and suggests users that are neighboring the target user up to some distance. The second algorithm is content-based and aims at suggesting users that may not be in the neighborhood of the target user, but whose tweets may be interesting to him/her.

3.1 Topology-based recommender

The general idea behind this algorithm is to suggest users that are in the neighborhood of the target user and that can be potential followees. A user's neighborhood is determined from the follower/followee relations in the social network. We apply the following heuristic to obtain the list of candidate users for recommendation:

1. Starting with the target user u_T, obtain the list of users he/she follows, let's call this list S, i.e. $S(u_T) = \bigcup_{\forall f \in followees(u_T)} f$.
2. For each element in S get its followers, let's call the union of all these lists L, i.e. $L(u_T) = \bigcup_{\forall s \in S} followers(s)$.
3. For each element in L obtain its followees, let's call the union of all these lists T, i.e.$T(u_T) = \bigcup_{\forall l \in L} followees(l)$.
4. Exclude from T those users that the target user is already following. Let's call the resulting list of candidates R, $R = T - S$.

Each element in R is a possible user to recommend to the target user. Notice that each element can appear more than once in R, depending on the number of times that each user appears in the the followees or followers lists obtained at steps 2 and 3 above.

The rationale behind this heuristic procedure is that the target user is an information seeker that has already identified some interesting users acting as information sources, which are his/her followees. Other people that also follows some of the users in this group (i.e. is subscribe to some of the same information sources) have interests in common with the target user and might have discover other relevant information sources in the same topics, which are in turn their followees.

Finally, we give each unique user $u_c \in R$ a score given by the Equation 1:

$$score(u_c) = \frac{occurrences(u_c, R)}{|R|} \times \frac{\mid followers(u_c) \mid}{\mid followees(u_c) \mid} \times \frac{\mid mentions(u_c) \mid}{M} \qquad (1)$$

The first term corresponds to the number of occurrences of the user in the final list of $|R|$ candidates for recommendations. The number of occurrences of a user u_c in this final list is an indicator of the amount of (indirect) neighbors that also have u_c as a (direct) connection itself.

The second term is the relation between the number of followers a user has with respect to the number of users that he/she follows. Since we seek for information sources

to be recommended, we assume that this kind of users will have many followers and that they will follow few people. In [11] it has been shown that the rankings of users that can be obtained by number of followers and by PageRank are very similar. We opted to use this factor as an estimator of the "importance" of a given user because the number of followers is a metric by far more easy to obtain that the PageRank score in a network with an order of almost 2 billion social relations. Cha et al. [1] also support the fact that the number of followers, along with both retweets and mentions, are factors that represents a user's influence on Twitter. They found that while the number of followers is an indicator of a user's popularity, retweets and mentions represent other important factors such as engaging the audience with valuable content.

For the reason expressed above, we finally add a factor that considers the number of times that a user has been mentioned in the social network in recent posts. According to Kwak et al. [11] ranking Twitter users by the number of retweets shows the rise of micro-blogging as an alternative communication media. In other words, retweets are considered the feature that has made Twitter a new medium of information dissemination. Hence, we consider mentions of a user instead of retweets because mentions are a broader concept that includes retweets. The most recent mentions to a user can be easily obtained through Twitter's Query API, up to a maximum of M mentions. Currently M is set to 100.

3.2 Content-based recommender

Information seekers are characterized for posting few tweets themselves, but follow people that generate content more actively. It is assumed that users actively select their followees expecting that their tweets will be interesting to them. Then, in order to develop a content-based followees recommender algorithm, we assumed that the interests of the target user can be described by the content of the tweets published by the users he/she follows. Let $tweets(u_f) = \{t_1, t_2, \cdots, t_k\}$ be the set of tweets published by user u_f, $profile_{base}(u_f)$ the term vector built from $tweets(u_f)$, and $followees(u_T) = \{f_1, f_2, \cdots, f_l\}$ the followees of user u_T. Then the profile of a user u_T is defined as the union of term vectors of his/her followees:

$$profile(u_T) = \bigcup_{\forall u_f \in followees(u_T)} profile_{base}(u_f)$$

In order to search for candidate recommendations, this algorithm does not take candidate users from the topology of the social network. Instead, it aims at discovering new users that may not be connected to the target user by a short path in the graph but appear in an information stream provided by Twitter which is known as *public timeline*. This stream contains the collection of the most recently published tweets, and is fed by all accounts that are not configured to be private. The public timeline can be considered as the current flow of information in Twitter, and is a good source to obtain active users in the social network.

The content-based algorithm we designed works as follows:

1. Obtain the authors of the most recent publications that appear in Twitter's public timeline, $U = \{u_1, u_2, \ldots, u_m\}$.

2. For each user $u_C \in U$, build $profile_{base}(u_C)$. That is, we build the term vector corresponding to each u_C .
3. For each user $u_C \in U$, compute

$$sim(u_C, u_T) = max_{\forall i: f_i \in followees(u_T)} sim_{cos}\,[profile_{base}(f_i), profile_{base}(u_C)]$$

   Where sim_{cos} is simply the cosine similarity between the two vectors.
   If $sim(u_C, u_T) > \gamma$, add u_C to the list of recommendations ordered by similarity.
4. Repeat steps 1 to 4 until the desired number of recommendations is obtained.

In order to build the term vectors associated to users, we first detect the language of the tweets[2] and then we apply the corresponding stop-word and stemming filters. We use a term frequency weighting scheme in the term vectors.

We use a similarity threshold of $\gamma = 0.1$ to consider a user relevant for recommendation. This threshold was set very low so that the desired number of recommendations could be obtained in a reasonable time. However, it can be adjusted according to the recommender application. For example, if recommendations can be calculated off-line the threshold can be set to a higher value, likely improving the precision of recommendations, at expense of some additional calculation time.

4 Experimental evaluation

4.1 Experiment setup

In order to evaluate the proposed algorithms, we have carried out a preliminary experiment using a group of 26 users. These users, 20 males and 6 females, were in the last years of their course of studies and were students of a "Recommender Systems" course dictated at our university as an elective course during 2010. The students selected for the experiment were volunteers familiarized with Twitter. We asked these users to create a new Twitter account[3] and to follow at least 20 Twitter users who publish information or news about a set of particular subjects of their interest. The general interests expressed by users ranged among diverse subjects. Some users only concentrated on one particular subject while others distributed their followees among several topics. Then we provided these users with a desktop tool that allowed them to login to Twitter and ask for recommendations using both methods (topology-based and content-based). The tool offered the logged user 20 recommended users and we asked them to explicitly evaluate whether the recommendations were relevant or not according to the same topical criteria they have chosen to select their followees as information sources.

For each recommendation in the resulting ranking the application showed the user name, description, profile picture and a link to the home page of the corresponding account. This link could be used to read the tweets published by the recommended user in the case that the information provided by the application was not enough to determine the student's interest in the recommendation. The question we asked students

[2] We are currently working with English and Spanish.

[3] We asked students to create a new Twitter account so that they did not need to reveal their real account and the people they follow

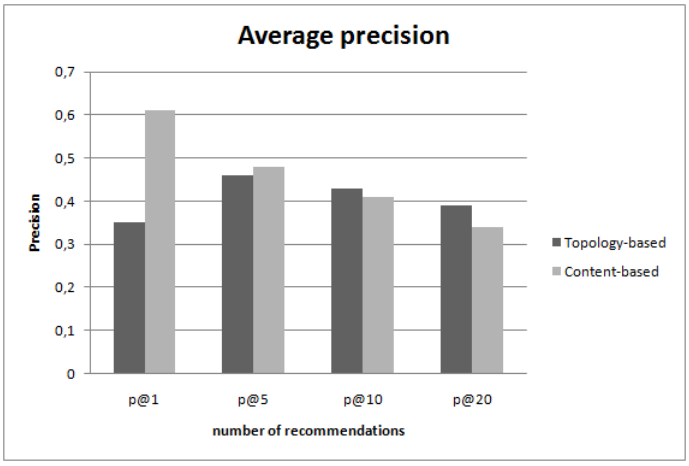


Fig. 1. Average precision for both recommendation algorithms

to ask themselves to determine whether a recommendation was relevant or not was "Would you have followed this recommended user in the first place (when selecting which users to follow in the first part of the experiment), if you had know this account?" For example, if a given student was interest in technology and he/she had not discovered the account @TechCrunch during his/her first selection of followees, that would be an interest recommendation because @TechCrunch tweets about news on technology.

4.2 Results

The quality of lists of top-N followee recommendations generated by each algorithm was first evaluated in terms of their overall precision. *Precision* can be defined as the number of relevant recommendations over the number of recommendations presented to the user and it can be also computed at different positions in the ranking. For example, P@5 ("*precision at five*") is defined as the percentage of relevant recommendations among the first five, averaged over all runs. Figure 1 shows the precision obtained for both algorithms at four different positions of the ranking: P@1, P@5, P@10 and P@20.

In general, both algorithms perform similarly at different positions of the ranking, with the exception of P@1 for which the content-based approach clearly outperforms the topology-based algorithm (61% of relevant recommendations for the content-based algorithm against 35% of relevant recommendations for the topology based-algorithm). For P@5 we obtained 48% of relevant recommendations for the content-based algorithm and 46% of relevant recommendations for the topology based algorithm. At this point we should point out that although we report precision up to 20 recommendations, recommender systems generally present to users shorter recommendations lists aiming at helping them to focus on the most relevant results. In these small lists the content-based algorithm reached good levels of precision, recommending mostly interesting users.

For recommendations lists longer than 5, performance decreases and we can observe that the topology based algorithm tends to give better results than the content-based algorithm. However the difference in performance of both algorithms is always lower than 5%. Due to the reduced number of users who participate in the experiment, we performed the Student's t-test of significance on the results obtained. The Student's t-test looks at the average difference between the performance scores of two algorithms, normalized by the standard deviation of the score difference. For this test we obtain that only the difference in precision at the first position of the ranking (P@1) is statistically significant.

Although precision measure gives a general idea of the overall performance of the presented algorithms, it is also very important to consider the position of relevant recommendations in the ranking presented to the user. Since it is known that users focus their attention on items at the top of a list of recommendations [9], if relevant recommendations appear at the top of the ranking using one algorithm and at the bottom of the ranking using the other, the first algorithm will be perceived by users as performing better even though their general precision might be similar.

Discounted cumulative gain (DCG) is a measure of effectiveness used to evaluate ranked lists of recommendations. DCG measures the usefulness, or gain, of a document based on its position in the result list using a graded relevance scale of documents in a list of recommendations,. The gain is accumulated from the top of the result list to the bottom with the gain of each result discounted at lower ranks. The premise of DCG is that highly relevant documents appearing lower in a list should be penalized as the graded relevance value is reduced logarithmically proportional to the position of the result. The DCG accumulated at a particular rank position k is defined as $DCG@k = rel_1 + \sum_{i=2}^{k} \frac{rel_i}{log_2 i}$. DCG is often normalized using an *ideal DCG* that is computed by sorting documents of a result list by relevance. Figure 2 shows the normalized DCG obtained for both algorithms at four different positions of the ranking: nDCG@1, nDCG@5, nDCG@10 and nDCG@20.

Success at rank k (S@k) is another metric commonly used to evaluate ranked lists of recommendations. The success at rank k is defined as the probability of finding a good recommendation among the top k recommended users. In other words, S@k is the percentage of runs in which there was at least one relevant user among the first k recommended users. Figure 3 shows the results we obtained for this metric for values of k ranging from 1 to 6.

S@1 is equivalent to P@1 by definition. Then, we can see that the content-based algorithm always positions relevant users earlier in the ranking than the topology-based algorithm. Indeed, all users in the experiment found a relevant recommendation before position 4 in the ranking using the content-based algorithm. For the topology-based algorithm, most users found a relevant recommendation before position 5 except for one user that found the first relevant recommendation at rank 6.

5 Discussion and Conclusions

In this article we presented two simple but effective algorithms for recommending users in the Twitter social network. The first algorithm models a given user from his/her con-

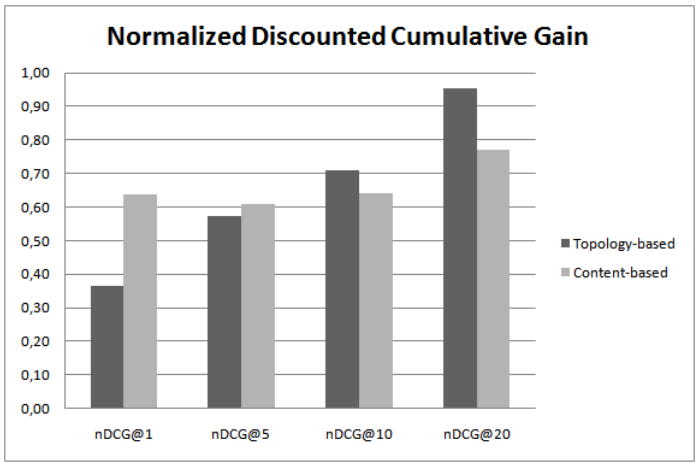


Fig. 2. Normalized discounted cumulative gain for both recommendation algorithms

nections in the social graph whereas the second algorithm models users using the content of the tweets published by his/her followees. We evaluated the proposed algorithms with real users and found that they work fairly similar in finding users that might result interesting for the target user to start following.

From the experiments presented we can conclude that although the average precision tend to be similar for both algorithms, if we consider the position on the recommendations in the ranking the content-based approach is better at giving good recommendations. We believe that results obtained with the content-based algorithm can be improved by setting a higher threshold for the similarity measure used for filtering the term vectors representing users. However, this will increase the response time of the algorithm since users are taken randomly from Twitter's public timeline.

Among the advantages of the topology-based algorithm, on the other hand, we can mention that recommendations can be found quickly based on a simple analysis of the network structure, without considering the content of the tweets posted by the candidate user.

Although the results reported seems promising, we are planning to repeat the experiment this year in order to involve more users in the experiment and obtain more statistical significance about the two proposed algorithms.

A natural extension in which we are currently working on is a hybrid algorithm that combines the best of both algorithms presented in this paper. This hybrid algorithm filters the candidate recommendations found with the topology-based method with a content-based analysis of the tweets posted by the candidate users. We are also very optimistic about the potential improvements that could be obtained with this hybrid approach.

As a possible limitation of our approach, we can mention that, we assumed that the target user is an *information-seeker* user, according to the categorization proposed by [8]. However, users may play different (and multiples) roles of information source,

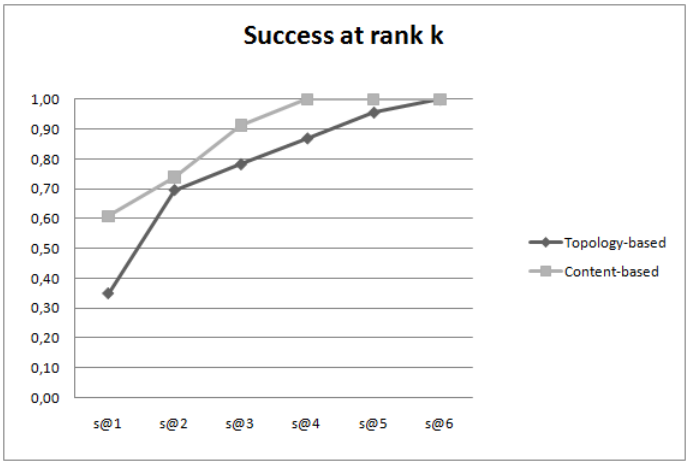


Fig. 3. Success at rank k for both recommendation algorithms

information seeker or friends in different communities. This is a challenging factor to consider that we leave for future investigation.

The experiments presented make us feel optimistic about the potential of a followee recommender system for Twitter using the methods described in this article or a combination of them. This work is the first step towards exploring the great potentials of this new platform to build recommendation systems.

References

1. Cha, M., Haddadi, H., Benevenuto, F., Gummadi, K.: Measuring user influence in Twitter: The million follower fallacy. In: Proc. of the 4th Int. Conf. on Weblogs and Social Media (ICWSM'10). Washington DC, USA (2010)
2. Chen, J., Geyer, W., Dugan, C., Muller, M., Guy, I.: Make new friends, but keep the old: recommending people on social networking sites. In: Proc. of the 27th Int. Conf. on Human Factors in Computing Systems. pp. 201–210 (2009)
3. Chen, J., Nairn, R., Nelson, L., Bernstein, M., Chi, E.: Short and tweet: experiments on recommending content from information streams. In: Proc. of the 28th Int. Conf. on Human Factors in Computing Systems (CHI'10). pp. 1185–1194 (2010)
4. Esparza, S.G., O'Mahony, M.P., Smyth, B.: On the real-time web as a source of recommendation knowledge. In: Proc. of the 4th ACM Conf. on Recommender Systems (RecSys'10). pp. 305–308 (2010)
5. Garcia, R., Amatriain, X.: Weighted content based methods for recommending connections in online social networks. In: Workshop on Recommender Systems and the Social Web. pp. 68–71. Barcelona, Spain (2010)
6. Guy, I., Ronen, I., Wilcox, E.: Do you know?: recommending people to invite into your social network. In: Proc. of the 13th Int. Conf. on Intelligent User Interfaces (IUI'09). pp. 77–86 (2009)
7. Hannon, J., Bennett, M., Smyth, B.: Recommending Twitter users to follow using content and collaborative filtering approaches. In: Proc. of the 4th ACM Conf. on Recommender Systems (RecSys'10). pp. 199–206 (2010)

8. Java, A., Song, X., Finin, T., Tseng, B.: Why we twitter: understanding microblogging usage and communities. In: Proc. of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis. pp. 56–65 (2007)
9. Joachims, T., Granka, L., Pan, B., Hembrooke, H., Gay, G.: Accurately interpreting clickthrough data as implicit feedback. In: Proc. of the 28th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR'05). pp. 154–161 (2005)
10. Krishnamurthy, B., Gill, P., Arlitt, M.: A few chirps about Twitter. In: Proc. of the 1st Workshop on Online Social Networks (WOSP'08). pp. 19–24 (2008)
11. Kwak, H., Lee, C., Park, H., Moon, S.: What is Twitter, a social network or a news media? In: Proc. of the 19th Int. Conf. on World Wide Web (WWW'10). pp. 591–600 (2010)
12. Liben-Nowell, D., Kleinberg, J.: The link prediction problem for social networks. In: Proc. of the 12th Int. Conf. on Information and knowledge management. pp. 556–559. CIKM '03, ACM, New York, NY, USA (2003)
13. Lo, S., Lin, C.: WMR–A graph-based algorithm for friend recommendation. In: Proc. of the 2006 IEEE/WIC/ACM Int. Conf. on Web Intelligence (WI'06). pp. 121–128. Washington, DC, USA (2006)
14. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The adaptive Web, pp. 325–341. Springer-Verlag (2007)
15. Phelan, O., McCarthy, K., Smyth, B.: Using Twitter to recommend real-time topical news. In: Proc. of the 3rd ACM Conf. on Recommender Systems (RecSys'09). pp. 385–388 (2009)
16. Schafer, J., Frankowski, D., Herlocker, J., Sen, S.: Collaborative Filtering Recommender Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web, LNCS, vol. 4321, chap. 9, pp. 291–324. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
17. Sun, A.R., Cheng, J., Zeng, D.D.: A novel recommendation framework for micro-blogging based on information diffusion. In: Proc. of the 19th Workshop on Information Technologies and Systems (2009)
18. Weng, J., Lim, E.P., Jiang, J., He, Q.: TwitterRank: finding topic-sensitive influential twitterers. In: Proc. of the 3rd ACM Int. Conf. on Web Search and Data Mining (WSDM'10). pp. 261–270. New York, NY, USA (2010)
19. Yamaguchi, Y., Takahashi, T., Amagasa, T., Kitagawa, H.: TURank: Twitter user ranking based on user-tweet graph analysis. In: Web Information Systems Engineering. LNCS, vol. 6488, pp. 240–253. Hong Kong, China (2010)

Adaptive Faceted Search on Twitter

Ilknur Celik[1], Fabian Abel[1], Patrick Siehndel[2]

[1] Web Information Systems, Delft University of Technology
{celik,abel}@tudelft.nl
[2] L3S Research Center, Leibniz University Hannover, Germany
siehndel@l3s.de

Abstract. In the last few years, Twitter has become a powerful tool for publishing and discussing information. Yet, content exploration in Twitter requires substantial efforts and users often have to scan information streams by hand. In this paper, we approach this problem by means of faceted search. We propose strategies for inferring facets and facet values on Twitter by enriching the semantics of individual Twitter messages and present different methods, including personalized and context-adaptive methods, for making faceted search on Twitter more effective.

Key words: faceted search, twitter, semantic enrichment, adaptation

1 Introduction

Twitter is a Social Web phenomenon that is attracting interest from people all around the world for a variety of different purposes [1], such as consuming and propagating news [2], crisis management [3] or communication with other people [4]. Over the last few years, Twitter has shown an exponential growth and became the most popular microblogging site with several hundreds of millions of users and more than 50 million Twitter messages (tweets) per day[3]. Highly active users regularly receive thousands of tweets every day [5]. This information overload may cause users to get lost in the information network, become demotivated and frustrated. Finding your way around Twitter is indeed not very straightforward due to the lack of a user-friendly browsing option that goes beyond the existing chronologically-ordered clutter option [5, 6].

Recently, researchers started to study strategies for recommending URLs [7], news articles [8] or entire conversations on Twitter [9]. However, search on Twitter has not been studied extensively yet which motivates, for example, the TREC 2011 track on Microblogs that defines first search tasks on Twitter[4]. In line with the TREC research objectives, we investigate ways to enhance search and content exploration in the microblogosphere by means of faceted search.

Traditional faceted search interfaces allow users to search for items by specifying queries regarding different dimensions and properties of the items (facets) [10]. For example, online stores such as eBay[5] or Amazon[6] enable end-users to narrow

[3] http://techcrunch.com/2010/06/08/twitter-190-million-users/
[4] http://sites.google.com/site/trecmicroblogtrack/
[5] http://ebay.com/
[6] http://amazon.com/

down their search for products by specifying constraints regarding facets such as the price, the category or the producer of a product. In contrast, information on Twitter is rather unstructured. Tweets are short text messages that do not explicitly feature facets. How can facets be extracted from tweets and how can we design appropriate faceted search strategies on Twitter? In this paper, we answer these questions and introduce an adaptive faceted search framework for Twitter. Our main contributions can be summarized as follows.

Semantic Enrichment We present methods for enriching the semantics of tweets by extracting facets (entities and topics) from tweets and related external Web resources.

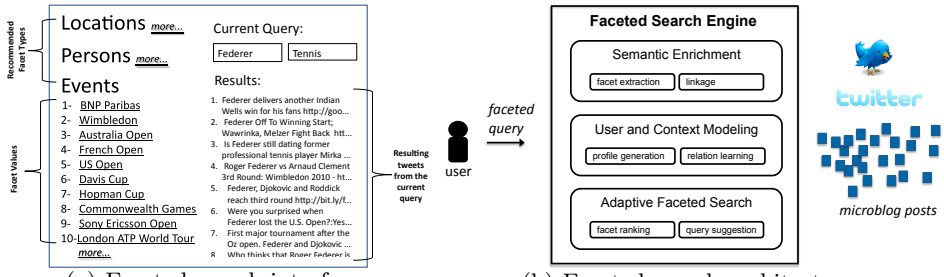
User and Context Modeling Given the semantically enriched tweets, we propose user and context modeling strategies that identify (current) interests of a given Twitter user and allow for contextualizing the demands of this user.

Adaptive Faceted Search We introduce faceted search strategies for content exploration on Twitter and propose methods that adapt to the interests and context of a user.

2 Faceted Search on Twitter

On Twitter, facets describe properties of a Twitter message. For example, persons that are mentioned in a tweet or events a tweet refers to. Oren et al. [10] formulate the problem of faceted search in RDF terminology. Given an RDF statement $(subject, predicate, object)$, the faceted search engine interprets (i) the subject as the actual resource that should be returned by the engine, (ii) the predicate as the facet type and (iii) the object as the facet value (restriction value). A faceted query (facet-value pair) that is sent to a faceted search engine thus consists of a predicate and an object. We follow this problem formulation proposed by Oren et al. [10] and interpret tweets as the actual resources the faceted search engine should return. If a tweet (subject) mentions an entity then the type of the entity is considered as facet type (predicate) and the actual identifier of the entity is considered as facet value (object). For example, given a tweet t that refers to the tennis player "Federer", the corresponding URI of the entity ($URI_{federer}$) and the URI of the entity type (URI_{person}) are used to describe the tweet by means of an RDF statement: $(t, URI_{person}, URI_{federer})$.

Figure 1(a) illustrates how we envision the corresponding faceted search interface that allows users to formulate faceted queries. Given a list of facet values which are grouped around facet types such as locations, persons and events, users can select facet-value pairs such as $(URI_{event}, URI_{wimbeldon})$ to refine their current query (see top right in Fig. 1(a): $(URI_{person}, URI_{federer})$, $(URI_{sportsgame}, URI_{tennis})$). A faceted query thus may consist of several facet-value pairs. Only those tweets that match all facet-value constraints will be returned to the user. The ranking of the tweets that match a faceted query is a research problem of its own and could be solved by exploiting the popularity of tweets – e.g. measured via the number of re-tweets or via the popularity of the user who published the tweet (cf. [11]). The core challenge of the faceted search interface is to support the facet-value selection as good as possible. Hence, the



(a) Faceted search interface (b) Faceted search architecture

Fig. 1. Adaptive faceted search on Twitter: (a) example interface and (b) architecture of the faceted search engine.

facet-value pairs that are presented in the faceted search interface (see left in Figure 1(a)) have to be ranked so that users can quickly narrow down the search result lists until they find the tweets they are interested in. Therefore, the *facet ranking problem* can be defined as follows.

Definition 1 (Facet Ranking Problem). *Given the current query F_{query}, which is a set of facet-value pairs $(predicate, object) \in F_{query}$, the hit list H of resources that match the current query, a set of candidate facet-value pairs $(predicate, object) \in F$ and a user u, who is searching for a resource t via the faceted search interface, the core challenge of the faceted search engine is to rank the facet-value pairs F. Those pairs should appear at the top of the ranking that restrict the hit list H so that u can retrieve t with the least possible effort.*

The effort, which u has to invest to narrow down the search result list H, can be measured by click and scroll operations (e.g. the number of facet-value pair selections). Our goal is to provide a faceted search interface that minimizes the effort a user has to invest in retrieving the tweets in which the user is interested in. Query suggestions and ranking facet-value pairs according to the current demands of a user are therefore essential and will be discussed in the next section.

2.1 Architecture for Adaptive Faceted Search on Twitter

Figure 1(b) illustrates the architecture of the engine that we propose for faceted search on Twitter. The main components of the engine are the following.

Semantic Enrichment The semantic enrichment layer aims to extract facets from tweets and generate RDF statements that describe the facet-value pairs which are associated with a Twitter message. In particular, each tweet is processed to identify entities (facet values) that are mentioned in the message. We therefore make use of the OpenCalais API[7], which allows for the extraction of 39 different types of entities (facet types) including persons, organizations, countries, cities and events. As Twitter messages are limited to 140 characters, the extraction of entities from tweets is a non-trivial problem. Thus, we introduced a set of strategies that link tweets with external Web resources (news

[7] `http://www.opencalais.com/`

articles) and propagate the semantics extracted from these resources to the related tweets, which is explained in detail in [8]. For example, given a tweet "This is great http://bit.ly/2fRds1t", we extract entities from the referenced resource (http://bit.ly/2fRds1t) and attach the extracted entities to the tweet.

User and Context Modeling In order to adapt the facet ranking to the people who are using the faceted search engine, we propose user modeling and context modeling strategies. The user modeling strategies model the interests of the users in certain facet values (entities and topics). We therefore exploit the tweets that have been published (including re-tweets) by a user. In future work, we also plan to consider click-through data from the faceted search engine. Context modeling covers mining of new knowledge from the Twitter data. We therefore propose relation learning strategies that exploit co-occurrence of entities in Twitter messages to infer typed relationships between entities [12].

Adaptive Faceted Search Based on the semantically enriched tweets, the learnt relationships between entities extracted from tweets and the user profiles generated by the user modeling layer, the adaptive faceted search layer solves the actual facet ranking problem. It provides methods that adapt the facet-value pair ranking to the given context and user. Furthermore, it provides query suggestions by exploiting the relations learnt from the Twitter messages. Given the current facet query, which is a list of facet-value pairs where each value refers to an entity, we can exploit relationships between entities in order to identify entities that are related to those entities that occur in the current facet query. We leave the analysis of such query suggestions for future work. Instead, we focus on the facet ranking problem and propose different strategies for ranking facet-value pairs in the next subsection.

2.2 Adaptive Faceted Search and Facet Ranking Strategies

Non-Personalized Facet Ranking A lightweight approach is to rank the facet-value pairs $(p, e) \in F$ based on their occurrence frequency in the current hit list H, the set of tweets that match the current query (cf. Definition 1):

$$rank_{frequency}((p, e), H) = |H_{(p,e)}| \quad (1)$$

$|H_{(p,e)}|$ is the number of (remaining) tweets that contain the facet-value pair (p, e) that can be applied to further filter the given hit list H. By ranking those facets that appear in most of the tweets, $rank_{frequency}$ minimizes the risk of filtering out relevant tweets but might increase the effort a user has to invest to narrow down search results.

Context-adaptive Facet Ranking The context-adaptive strategy exploits relationships between entities (facet values) to produce the facet ranking. A relationship is therefore defined as follows:

Definition 2 (Relationship). *Given two entities e_1 and e_2, a* relationship *between these entities is described via a tuple $rel(e_1, e_2, type, t_{start}, t_{end}, w)$, where* type *labels the relationship, t_{start} and t_{end} specify the temporal validity of the relationship and $w \in [0..1]$ is a weighting score that allows for specifying the strength of the relationship.*

The higher the weighting score w the stronger the relationship between e_1 and e_2. We use co-occurrence frequency as weighting scheme. Hence, given the enriched tweets, we count the number of tweets both entities (e_1 and e_2) are associated with. The context-adaptive facet ranking strategy ranks the facet-value pairs $(p, e) \in F$ according to $w(e_i, e)$, where e_i is a facet value that is already part of the given query: $(p_i, e_i) \in F_{query}$ (cf. Definition 1):

$$rank_{relation}((p, e), F_{query}) = \sum_i w(e_i, e) | (p, e_i) \in F_{query} \quad (2)$$

Hence, the context-sensitive strategy can only be applied in situations where the user has already made one selection, so that $|F_{query}| > 0$.

Personalized Facet Ranking The personalized facet ranking strategy adapts the facet ranking to a given user profile that is generated by the user modeling layer depicted in Figure 1(b). User profiles conform to the following model and specify a user's interest into a specific facet value (entity).

Definition 3 (User Profile). *The profile of a user $u \in U$ is a set of weighted entities where with respect to the given user u for an entity $e \in E$ its weight $w(u, e)$ is computed by a certain function w.*

$$P(u) = \{(e, w(u, e)) | e \in E, u \in U\}$$

Here, E and U denote the set of entities and users respectively.

Given the set of facet-value pairs $(p, e) \in F$ (see Definition 1), the personalized facet ranking strategy utilizes the weight $w(u, e)$ in $P(u)$ to rank the facet-value pairs:

$$rank_{personalized}((p, e), P(u)) = \begin{cases} w(u, e) \text{ if } w(u, e) \in P(u) \\ 0 \qquad \text{otherwise} \end{cases} \quad (3)$$

By combining the above three strategies it is possible to generate further facet ranking methods. A combination of two strategies can be realized by building the weighted average computed for a given facet-value pair (p, e) (e.g. $rank_{combined} = \alpha \cdot rank_{\alpha}((p, e)) + (1 - \alpha) \cdot rank_{\beta}((p, e))$, where $\alpha \in [0..1]$).

3 Preliminary Analysis and Future Work

In this paper, we explored strategies for faceted search on Twitter. We presented a framework that enriches the semantics of Twitter messages in order to generate facets that describe the content of tweets (e.g. persons, locations, organizations). To do so, we extracted entities both from tweets and linked external Web resources. Our analysis based on a large Twitter dataset[8] showed that the exploitation of links for enriching tweets is necessary to better support faceted search on Twitter, due to the obvious increase in the number of facet values when tweets are enriched with entities of related news articles. We proposed an adaptive faceted search engine with different strategies for ranking facets including methods that adapt to the actual context and user. The context-adaptive method exploits relationships between facet values (entities) that we learn from

[8] We make our dataset publicly available at `http://wis.ewi.tudelft.nl/umap2011/`

tweets and linked news articles [12]. Our strategy discovers relationships between persons/groups (including organizations) and events (including political, sports and entertainment) with high precision of 0.92 and 0.87 regarding P@10 and P@20. Our analysis indicates that these relationships can be used to suggest and rank facets that are related to the current context (the current faceted query) with high precision.

Furthermore, the personalized facet ranking requires a user profile $P(u)$ in order to adapt the facet ranking to the preferences of the user. In [8], we showed that our user modeling strategies, which are based on semantic enrichment of tweets, outperform other strategies such as hashtag-based approaches significantly for recommending news articles. In future work, we will analyze the applicability of those user modeling strategies for our adaptive faceted search engine. In our evaluation, we will measure the effect of our facet ranking strategies (i) in the context of an automatic experiment as proposed by Koren et al. [13] and (ii) in practice by enabling real users to experiment with our adaptive faceted search interface for Twitter.

References

1. Java, A., Song, X., Finin, T., Tseng, B.: Why we twitter: understanding microblogging usage and communities. In: WebKDD/SNA-KDD, ACM (2007) 56–65
2. Kwak, H., Lee, C., Park, H., Moon, S.: What is Twitter, a social network or a news media? In: WWW, ACM (2010) 591–600
3. Hughes, A.L., Palen, L.: Twitter Adoption and Use in Mass Convergence and Emergency Events. In: ISCRAM, iscram.org (2009)
4. Zhao, D., Rosson, M.B.: How and why people Twitter: the role that micro-blogging plays in informal communication at work. In: GROUP, ACM (2009) 243–252
5. Bernstein, M., Kairam, S., Suh, B., Hong, L., Chi, E.H.: A torrent of tweets: managing information overload in online social streams. In: CHI Workshop on Microblogging (2010)
6. Owens, J.W., Lenz, K., Speagle, S.: Trick or Tweet: How Usable is Twitter for First-Time Users? Usability News **11** (2009)
7. Chen, J., Nairn, R., Nelson, L., Bernstein, M., Chi, E.: Short and tweet: experiments on recommending content from information streams. In: CHI, ACM (2010) 1185–1194
8. Abel, F., Gao, Q., Houben, G.J., Tao, K.: Analyzing User Modeling on Twitter for Personalized News Recommendations. In: UMAP, Springer (2011)
9. Chen, J., Nairn, R., Chi, E.H.: Speak Little and Well: Recommending Conversations in Online Social Streams. In: CHI, ACM (2011)
10. Oren, E., Delbru, R., Decker, S.: Extending faceted navigation for RDF data. In: ISWC, Springer (2006) 559–572
11. Weng, J., Lim, E.P., Jiang, J., He, Q.: TwitterRank: finding topic-sensitive influential Twitterers. In: WSDM, ACM (2010) 261–270
12. Celik, I., Abel, F., Houben, G.-J.: Learning Semantic Relationships between Entities in Twitter. In: ICWE, Springer (2011)
13. Koren, J., Zhang, Y., Liu, X.: Personalized interactive faceted search. In: WWW, ACM (2008) 477–486

cTag: Semantic Contextualisation of Social Tags

Ignacio Fernández-Tobías, Iván Cantador, Alejandro Bellogín

Departamento de Ingeniería Informática, Universidad Autónoma de Madrid
28049 Madrid, Spain
ign.fernandez01@estudiante.uam.es, {ivan.cantador, alejandro.bellogin}@uam.es

Abstract. In this paper, we present an algorithmic framework to identify the semantic meanings and contexts of social tags within a particular folksonomy, and exploit them for building contextualised tag-based user and item profiles. We also present its implementation in a system called cTag, with which we preliminary analyse semantic meanings and contexts of tags belonging to Delicious and MovieLens folksonomies.

Keywords: social tagging, folksonomy, ambiguity, semantic contextualisation, clustering, user modelling.

1 Introduction

Social tagging has become a popular practice as a lightweight mean to classify and exchange information. Users create or upload content (resources, items), annotate it with freely chosen words (tags), and share these annotations with others. In a social tagging system, the whole set of tags constitutes an unstructured collaborative knowledge classification scheme that is commonly known as *folksonomy*. This implicit classification serves various purposes, such as for item organisation, promotions, and sharing with friends or with the public. Studies have shown, however, that tags are generally chosen by users to reflect their interests [8]. These findings lend support to the idea of using tags to derive precise user preferences, and bring with new research opportunities on personalised search and recommendation [11,12,13].

Despite the above advantages, social tags are free text, and thus suffer from various vocabulary problems. Ambiguity (polysemy) of the tags arises as users apply the same tag in different domains (e.g. `bridge`, the architectonical structure vs. the card game). At the opposite end, the lack of synonym control can lead to different tags being used for the same concept, precluding collocation (e.g. `biscuit` and `cookie`). Synonym relations can also be found in the form of acronyms (e.g. `nyc` for `new york city`), and morphological deviations (e.g. `blog`, `blogs`, `blogging`). Moreover, there are tags that have single meanings, but are used in different semantic contexts that should be distinguished (e.g. `web` may be used to annotate items about distinct topics such as Web development, Web browsers, and Web 2.0).

Aiming to address such problems, we present herein a system called cTag, which consists of an algorithmic framework that allows identifying semantic meanings and contexts of social tags within a particular folksonomy, and exploits them to build contextualised tag-based user and item profiles.

2 Semantic Contexts of Social Tags

Current folksonomy-based content retrieval systems have a common limitation: they do not deal with semantic ambiguities of tags. For instance, given a tag such as `sf`, existing content retrieval strategies do not discern between the two main meanings of that tag: *San Francisco* (the Californian city) and *Science Fiction* (the literary genre).

Semantic ambiguity of social tags, on the other hand, is being investigated in the literature. There are approaches that attempt to identify the actual meaning of a tag by linking it with structured knowledge bases [1,6]. These approaches, however, rely on the availability of external knowledge bases, and so far are preliminary, and have not been applied to personalised search and recommendation.

Other works are based on the concept of tag co-occurrence, that is, on extracting tag semantic meanings and contexts within a particular folksonomy by clustering the tags according to their co-occurrences in item annotation profiles [2,7,14]. For example, for the tag `sf`, often co-occurring tags such as `sanfrancisco`, `california` and `bayarea` may be used to define the context "San Francisco, the Californian city", while co-occurring tags like `sciencefiction`, `scifi` and `fiction` may be used to define the context "Science Fiction, the literary genre."

In this paper, we follow a clustering strategy as well, but in contrast to previous approaches, ours provides the following benefits:

- Instead of using simple tag co-occurrences, we propose to use more sophisticated tag similarities, which were presented by Markines et al. in [9], and are derived from established information theoretic and statistical measures.
- Instead of using standard hierarchical or partitional clustering strategies, which require defining a stop criterion for the clustering processes, we propose to apply the graph clustering technique presented by Newman and Girvan [10], which automatically establishes an optimal number of clusters. Moreover, to obtain the contexts of a particular tag, we propose not to cluster the whole folksonomy tag set, but a subset of it.

In the following, we briefly describe the above tag similarities and clustering technique. In Section 3, we shall explain how obtained tag similarities and clusters are exploited to contextualise tag-based profiles.

2.1 Tag Similarities

A folksonomy $\mathcal{F}$ can be defined as a tuple $\mathcal{F} = \{\mathcal{T}, \mathcal{U}, \mathcal{I}, \mathcal{A}\}$, where $\mathcal{T} = \{t_1, \dots, t_L\}$ is the set of tags that comprise the vocabulary expressed by the folksonomy, $\mathcal{U} = \{u_1, \dots, u_M\}$ and $\mathcal{I} = \{i_1, \dots, i_N\}$ are respectively the sets of users and items that annotate and are annotated with the tags of $\mathcal{T}$, and $\mathcal{A} = \{(u_m, t_l, i_n)\} \in \mathcal{U} \times \mathcal{T} \times \mathcal{I}$ is the set of assignments (annotations) of each tag t_l to an item i_n by a user u_m.

To compute semantic similarities between tags, we follow a two step process. First, we transform the tripartite space of a folksonomy, represented by the triples $\{(u_m, t_l, i_n)\} \in \mathcal{A}$, into a set of tag-item relations $\{(t_l, i_n, w_{l,n})\} \in \mathcal{T} \times \mathcal{I} \times \mathbb{R}$ (or tag-user relations $\{(t_l, u_m, w_{l,m})\} \in \mathcal{T} \times \mathcal{U} \times \mathbb{R}$), where $w_{l,n}$ (or $w_{l,m}$) is a real number that expresses the relevance (importance, strength) of tag t_l when describing item profile i_n (or user profile u_m). In [9], Markines et al. call this transformation as tag assignment "aggregation", and present and evaluate a number of different aggregation methods.

We focus on two of these methods, *projection* and *distributional* aggregation, which are described with a simple example in Figure 1. Projection aggregation is based on the Boolean use of a tag for annotating a particular item, while distributional aggregation is based on the popularity (within the community of users) of the tag for annotating such item. Second, in the obtained bipartite tag-item (or tag-user) space, we compute similarities between tags based on co-occurrences of the tags in item (or user) profiles. In [9], the authors compile a number of similarity metrics derived from established information theoretic and statistical measures. cTag computes some of these metrics, whose definitions are given in Table 1.

Tag assignments [user, tag, item]

Alice	conference	recommender	research
www.umap2011.org	1	1	
www.delicious.com		1	
ir.ii.uam.es		1	1

Bob	conference	recommender	research
www.umap2011.org	1	1	1
www.delicious.com		1	
ir.ii.uam.es			

⇩

Tag assignment aggregation [tag, item]

Projection	conference	recommender	research
www.umap2011.org	1	1	1
www.delicious.com		1	
ir.ii.uam.es		1	1

Distributional	conference	recommender	research
www.umap2011.org	2	2	1
www.delicious.com		2	
ir.ii.uam.es		1	1

Figure 1. An example of projection and distributional tag assignment aggregations. 2 users, Alice and Bob, annotate 3 Web pages with 3 tags: `conference`, `recommender` and `research`.

Table 1. Tested tag similarity metrics. $I_1, I_2 \subseteq I$ are the sets of items annotated with $t_1, t_2 \in \mathcal{T}$.

Similarity	***Projection aggregation***	***Distributional aggregation***
Matching	$sim(t_1,t_2) = \lvert I_1 \cap I_2 \rvert$	$sim(t_1,t_2) = -\sum_{t \in I_1 \cap I_2} \log\ \mathrm{p}(t)$
Overlap	$sim(t_1,t_2) = \frac{\lvert I_1 \cap I_2 \rvert}{\min(I_1, I_2)}$	$sim(t_1,t_2) = \frac{\sum_{t \in I_1 \cap I_2} \log\ \mathrm{p}(t)}{\max(\sum_{t \in I_1} \log \mathrm{p}(t), \sum_{t \in I_2} \log\ \mathrm{p}(t))}$
Jaccard	$sim(t_1,t_2) = \frac{\lvert I_1 \cap I_2 \rvert}{\lvert I_1 \cup I_2 \rvert}$	$sim(t_1,t_2) = \frac{\sum_{t \in I_1 \cap I_2} \log\ \mathrm{p}(t)}{\sum_{t \in I_1 \cup I_2} \log\ \mathrm{p}(t)}$
Dice	$sim(t_1,t_2) = \frac{2\lvert I_1 \cap I_2 \rvert}{\lvert I_1 \rvert + \lvert I_2 \rvert}$	$sim(t_1,t_2) = \frac{2\sum_{t \in I_1 \cap I_2} \log\ \mathrm{p}(t)}{\sum_{t \in I_1} \log \mathrm{p}(t) + \sum_{t \in I_2} \log\ \mathrm{p}(t)}$
Cosine	$sim(t_1,t_2) = \frac{I_1}{\sqrt{\lvert I_1 \rvert}} \cdot \frac{I_2}{\sqrt{\lvert I_2 \rvert}} = \frac{\lvert I_1 \cap I_2 \rvert}{\sqrt{\lvert I_1 \rvert \cdot \lvert I_2 \rvert}}$	$sim(t_1,t_2) = \frac{I_1}{\lVert I_1 \rVert} \cdot \frac{I_2}{\lVert I_2 \rVert}$

2.2 Tag Clustering

We create a graph G, in which nodes represent the social tags of a folksonomy, and edges have weights that correspond to semantic similarities between tags. By using the similarity metrics presented in Section 2.1, G captures global co-occurrences of tags within item annotations, which in general, are related to *synonym* and *polysemy* relations between tags.

Once G is built, we apply the graph clustering technique presented by Newman and Girvan in [10], which automatically establishes an optimal number of clusters. However, we do not cluster G, but subgraphs of it. Specifically, for each tag $t_l \in \mathcal{T}$, we select its T_1 most similar tags and then, for each of these new tags, we select its T_2 most similar tags[1] to allow better distinguishing semantic meanings and contexts of t_l within the set of T_1 most similar tags. With all the obtained tags (at most $1 + T_1T_2$), we create a new graph G_l, whose edges are extracted from the global graph G.

Tables 2 and 3 show examples of semantic meanings and contexts retrieved by our approach for Delicious[2] and MovieLens[3] tags. Delicious is an online system where users bookmark and tag Web pages. Since bookmarks can be related with any topic, a wide range of domains are covered by Delicious tags, and semantic meanings are easily distinguished in many cases. It can be seen, for instance, that most of the Web pages tagged with `sf` are about *San Francisco* and *Science Fiction*. Moreover, for a particular meaning, several contexts can be found. Web pages about San Francisco may belong to *restaurants* or announce *events* in that city.

Table 2. Examples of semantic contexts identified for different Delicious tags.

tag	context centroid	context popularity	context tags
sf	fiction	0.498	fiction, scifi, sciencefiction, schi-fi, stores, fantasy, literature
	sanfrancisco	0.325	sanfrancisco, california, bayarea, losangeles, la
	restaurants	0.082	restaurants, restaurant, dining, food, eating
	events	0.016	events, event, conferences, conference, calendar
web	webdesign	0.434	webdesign, webdev, web_design, web-design, css, html
	web2.0	0.116	web2.0, socialnetworks, social, socialmedia
	javascript	0.077	javascript, js, ajax, jquery
	browser	0.038	browser, browsers, webbrowser, ie, firefox
holiday	christmas	0.336	christmas, xmas
	travel	0.274	travel, trip, vacation, tourism, turismo, planner
	airlines	0.104	airlines, airline, flights, flight, cheap
	rental	0.019	rental, apartment, housing, realestate

MovieLens, on the other hand, is a recommender system where users rate and tag movies. We may expect that the number of contexts for a particular tag in MovieLens folksonomy is much lower than in Delicious' since the scope of the former (movies belonging to a limited number of genres) is smaller than the latter (Web pages related to any domain and topic). Moreover, we may also expect that distinct meanings and contexts of a particular tag are hardly differentiated in MovieLens since the number of tags and tag assignments per user and item is lower than in Delicious. Examples in Table 3, however, show that is not necessarily the case: there are `animation` movies produced by different studios (e.g. Disney and Pixar), movies interpreted by `Will Smith`, the American actor, with different genres (e.g. comedy, action, and science fiction), and movies with characters that can be described based on different facets, e.g. `James Bond`, as a spy, as a killer, or as a hero.

[1] In preliminary experiments, we have tested $T_1 = 20, 25, 30$ and $T_2 = 3, 5$

[2] Delicious - Social bookmarking, http://www.delicious.com

[3] MovieLens - Movie recommendations, http://www.movielens.org

Table 3. Examples of semantic contexts identified for different MovieLens tags.

tag	context centroid	context popularity	context tags
animation	animals	0.354	animals, children, fun, kids, talking animals
	pixar	0.147	cartoon, inventive, pixar, toys come to life, vivid characters
	disney	0.127	classic, disney, disney studios, family, fantasy
	anime	0.032	anime, hayao miyazaki, japanese, studio ghibli, zibri studio
will smith	fantasy	0.226	fantasy, seen more than once, adventure, action, exciting
	funny	0.032	funny, comedy, jim carrey, claymation, very funny
	conspiracy	0.020	conspiracy, michael moore, twist ending, politics
	comic	0.016	comic, adapted from comic, superhero, based on a comic
james bond	murder	0.427	murder, bond, 007, assassin, killer as protagonist, serial killer
	action	0.079	action, scifi, adventure, superhero
	espionage	0.074	espionage, matt damon, robert ludlum, tom cruise, spies
	england	0.041	england, british, uk, based on a book

3 Semantically Contextualised Tag-based Profiles

We define the profile of user u_m as a vector $\mathbf{u}_m = (u_{m,1}, \ldots, u_{m,L})$, where $u_{m,l}$ is a weight (real number) that measures the "informativeness" of tag t_l to characterise contents annotated by u_m. Similarly, we define the profile of item i_n as a vector $\mathbf{i}_n = (i_{n,1}, \ldots, i_{n,L})$, where $i_{n,l}$ is a weight that measures the relevance of tag t_l to describe i_n. There exist different schemes to weight the components of tag-based user and item profiles. Some of them are based on the information available in individual profiles, while others draw information from the whole folksonomy. We have implemented several forms of weighting strategies based on the well-known TF, TF-IDF, and BM25 information retrieval models [3].

In each of the built profile, a tag t_l is transformed into a semantically contextualised tag t_l^m (or t_l^n), which is formed by the union of t_l and the semantic context $c_{l,m}$ (or $c_{l,n}$) of t_l within the corresponding user profile u_m (or item profile i_n). For instance, tag `sf` in a user profile with tags like `city`, `california` and `bayarea` may be transformed into a new tag `sf|sanfrancisco`, since in that profile, "sf" clearly refers to San Francisco, the Californian city. With this new tag, matchings with item profiles containing contextualised tags such as `sf|fiction`, `sf|restaurants` or `sf|events` would be discarded by a personalised search or recommendation algorithm because they may annotate items related to Science Fiction, or more specific topics of San Francisco like restaurants and events in the city.

More formally, the context (centroid) $c_{l,m}$ (or $c_{l,n}$) of tag t_l within the user profile u_m (or item profile i_n), and the corresponding contextualised tag t_l^m (or t_l^n) are defined as follows:

$$\forall (u_m, t_l, i_n) \in \mathcal{A}, \quad \begin{array}{l} c_{l,m} = \mathrm{c}(t_l, u_m) = \arg\max_{c_l} cos(\mathbf{c}_l, \mathbf{u}_m) \Rightarrow t_l^m = t_l \cup c_{l,m} \\ c_{l,n} = \mathrm{c}(t_l, i_n) = \arg\max_{c_l} cos(\mathbf{c}_l, \mathbf{i}_n) \Rightarrow t_l^n = t_l \cup c_{l,n} \end{array}$$

where $\mathbf{c}_l = (c_{l,1}, \ldots, c_{l,L})$ is the weighted list of tags that define each of the contexts c_l of tag t_l within the folksonomy (see Tables 2 and 3).

Tables 4 and 5 show several examples of contextualised tag-based Delicious and MovieLens profiles generated by our approach. Each table shows four item profiles in which two of them contain a certain tag, but used in two different contexts: `sf` as *San Francisco* and *Science Fiction*, `web` in the contexts of *Web development* and *Web 2.0*, *Disney* or *Anime* `animation` movies, `will smith` featuring *fantasy* or *funny* movies.

Table 4. Four semantically contextualised tag-based item profiles of Delicious dataset. Each original *tag* is transformed into a *tag*|*context* pair.

bayarea|sf	california|sf	city|sustainability	conservation|green	eco|green
environment|recycle	government|activism	green|environment	home|green	local|sanfrancisco
recycle|environment	recycling|environment	sanfrancisco|sf	**sf|sanfrancisco**	solar|environment
sustainability|recycling	sustainable|green	trash|green	urban|sustainability	volunteer|environmental
culture|philosophy	essay|interesting	fiction|sf	future|scifi	futurism|philosophy
god|science	interesting|science	literature|scifi	mind|philosophy	read|philosophy
religion|philosophy	research|science	sci-fi|sf	sciencefiction|sf	scifi|writing
sf|fiction	storytelling|fiction	toread|philosophy	universe|philosophy	writing|fiction
ajax|javascript	css|javascript	design|web	embed|webdesign	framework|javascript
gallery|jquery	html|javascript	icons|web	javascript|ajax	jquery|webdev
js|javascript	library|javascript	plugin|webdev	programming|javascript	site|webdev
toolkit|webdev	tutorials|webdev	**web|javascript**	web2.0|web	webdev|javascript
articles|web	blogs|web2.0	idea|community	internet|tools	library|opensource
network|tools	podcasts|education	rdf|web	reading|education	school|educational
semantic|semanticweb	semanticweb|web	semweb|semanticweb	software|utilities	technology|web2.0
tim|web	trends|technology	**web|web2.0**	web2.0|social	wiki|web2.0

Table 5. Four semantically contextualised tag-based item profiles of MovieLens dataset. Each original *tag* is transformed into a *tag*|*context* pair.

3d|animated	**animation|disney**	pixar animation|animation	comedy|animation	fun|adventure
disney|family	kids|toys come to life	animated|pixar animation	funny|animation	bright|toys come to life
computer|animation	disney animation|pixar	favorite|toys come to life	fantasy|animation	family|disney
toys|toys come to life	pixar|toys come to life	toys come to life|animated	classic|comedy	funny|animation
fantasy|zibri studio	dragon|anime movie	mythical creatures|anime	secret door|anime	japan|zibri studio
animation|anime	miyazaki|zibri studio	hayao miyazaki|myazaki	zibri strudio|anime	myazaki|zibri studio
fun|adventure	adventure|zibri studio	environment|mythical creatures	animated|animation	strange|foreign
foreign|japan	great anime film|anime	anime movie|mythical creatures	fanciful|zibri studio	anime|zibri studio
oscar winner|scifi	aliens|scifi	**will smith|fantasy**	frantic|scifi	end of the world|scifi
adventure|scifi	want|scifi	seen more than once|scifi	sf|scifi	action|fantasy
alien invasion|action	scifi|fantasy	seen at the cinema|scifi	war|action	disaster|scifi
dvd|space	watchfully|action	patriotic|scifi	invasion|scifi	et|scifi
comedy|funny	humor|comedy	end of the world|scifi	stupid|comedy	aliens|stupid
funny|comedy	amazing|fantasy	formulaic|will smith	action|fantasy	very funny|funny
predictable|scifi	fight|funny	seen more than once|comedy	futurism|scifi	cool|comedy
will smith|funny	cool but freaky|funny	violently stupid|comedy	dvd|space	space|alien invasion

4 cTag

cTag[4] is a system with the implementation of the algorithmic framework for tag and profile contextualisation presented in Sections 2 and 3, and allows using and testing it through a Web application and a Web service. Figure 2 shows a screenshot of cTag Web application. The user selects a dataset –Delicious or MovieLens– and a tag similarity, queries for a social tag available in the dataset, and obtains the semantic contexts associated to that tag. The user can also set a profile (manually or automatically via Delicious API) to contextualise. The retrieved contexts (clusters) are shown in the form of weighted lists of tag clouds, and a coloured clustered graph.

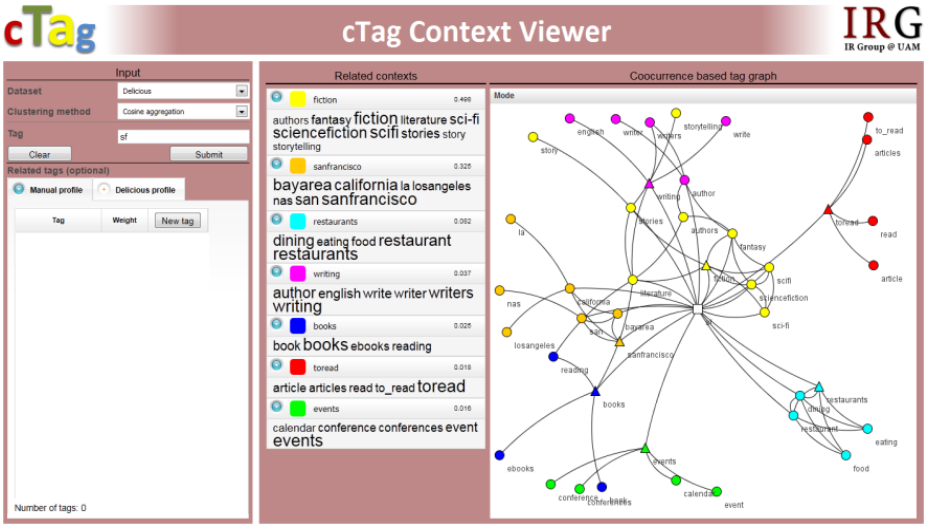


Figure 2. Screenshot of cTag Web application.

Figure 3 shows the XML response from cTag Web service for the input tag `sf` and profile {(`books`, 0.7), (`sci-fi`, 0.3)}, by using the cosine aggregation method with T_1=20 and T_2=5, on Delicious dataset. It can be seen that two semantic contexts are retrieved: *books* and *fiction*. Both of them are related to *Science Fiction* genre, but the former takes a higher weight since it focuses on books and readings, which is the main topic of the input profile.

[4] cTag Web application and Web service, http://ir.ii.uam.es/reshet/results.html

```
<tag_contextualization_results method="cosine_aggregation_20_5" dataset="delicious">
   <tag value="sf">
      <profile>
         <profile_tag weight="0.7">books</profile_tag>
         <profile_tag weight="0.3">sci-fi</profile_tag>
      </profile>
      <contexts>
         <context name="books" similarity="0.107571">
            <context_tag weight="0.35857">books</context_tag>
            <context_tag weight="0.229219">book</context_tag>
            <context_tag weight="0.207827">ebooks</context_tag>
            <context_tag weight="0.204383">reading</context_tag>
         </context>
         <context name="fiction" similarity="0.0806848">
            <context_tag weight="0.145413">fiction</context_tag>
            <context_tag weight="0.144174">scifi</context_tag>
            <context_tag weight="0.12935">sciencefiction</context_tag>
            <context_tag weight="0.115264">sci-fi</context_tag>
            <context_tag weight="0.099144">stories</context_tag>
            <context_tag weight="0.0890222">fantasy</context_tag>
            <context_tag weight="0.0834318">literature</context_tag>
            <context_tag weight="0.0683994">authors</context_tag>
            <context_tag weight="0.0661398">story</context_tag>
            <context_tag weight="0.0596612">storytelling</context_tag>
         </context>
      </contexts>
   </tag>
</tag_contextualization_results>
```

Figure 3. Example XML response from cTag Web service.

As shown in Table 6, in addition to the differences in the number and nature of their domains, cTag datasets[5] obtained from Delicious and MovieLens systems present distinct characteristics that may affect the contextualisation process (Table 7), and its further application to folksonomy-based personalisation and recommendation strategies. Although the number of users is quite similar (~2K) for both datasets, the number of tagged items (and tag assignments) is much different; the purpose of Delicious is bookmarking and tagging Web pages, and MovieLens's is rating movies. Moreover, in Delicious dataset, a significant amount of tags was not contextualised because they are expressions that are not commonly shared by the community.

Table 6. Description of cTag datasets.

	Delicious	**MovieLens**
#users	1867	2113
#items	69226	5909
#tags	53388	5291
Avg. #tags/user	123.697 (99.870)	10.093 (52.193)
Avg. #tags/item	7.085 (3.397)	6.353 (8.141)
#TAS	437593	47958
Avg. #TAS/user	234.383 (192.395)	22.697 (169.948)
Avg. #TAS/item	6.321 (6.356)	8.116 (12.638)
#contextualised tags	14295	5291

[5] cTag datasets, published at HetRec'11 workshop: http://ir.ii.uam.es/hetrec2011

Table 7. Description of obtained clusters for each dataset and tag similarity.

		Delicious		MovieLens	
		Avg. #clusters/tag	**Avg. cluster size**	**Avg. #clusters/tag**	**Avg. cluster size**
Projection aggregation	**Matching**	**4.870** (1.517)	8.698 (3.897)	**6.165** (1.743)	7.875 (4.433)
	Overlap	**9.687** (3.022)	7.310 (3.270)	**10.154** (2.721)	7.305 (3.547)
	Jaccard	8.397 (2.848)	6.630 (2.674)	8.616 (2.902)	6.768 (3.501)
	Dice	8.407 (2.846)	6.622 (2.678)	8.633 (2.909)	6.754 (3.497)
	Cosine	8.579 (2.878)	6.538 (2.678)	8.719 (2.967)	6.689 (3.477)
Distributional aggregation	**Matching**	**4.875** (1.502)	8.687 (3.885)	**6.036** (1.745)	7.995 (4.382)
	Overlap	**9.767** (3.031)	7.244 (3.213)	**10.443** (2.796)	7.019 (3.402)
	Jaccard	8.403 (2.844)	6.640 (2.686)	8.868 (2.823)	6.808 (3.328)
	Dice	8.413 (2.845)	6.631 (2.682)	8.887 (2.832)	6.793 (3.326)
	Cosine	9.019 (2.858)	6.511 (2.576)	8.874 (3.135)	6.182 (3.169)

5 Conclusions and Future Work

In this paper, we have presented cTag, a system which consists of an algorithmic framework to identify the semantic meanings and contexts of social tags within a particular folksonomy, and exploit them for building contextualised tag-based user and item profiles. The main benefit of cTag approach is that it utilises a clustering technique that exploits sophisticated co-occurrence based similarities between tags, is very efficient since it is not executed on the whole tag set of the folksonomy, and provides an automatic stop criterion to establish the optimal number of clusters.

As shown in previous works [1,7,11,13], semantic disambiguation and contextualisation of social tags can be used to improve folksonomy-based personalised search and recommendation strategies. Recently, in [3], we have preliminary evaluated cTag with a number of state of the art recommenders [4] on a Delicious dataset, and have obtained 13% to 24% precision/recall improvements by only contextualising 5.3% of the tags available in that dataset. In the study, we have also conducted a manual evaluation of our tag contextualisation approach. By considering as ground-truth data a set of 1,080 manual context assignments provided by 30 human evaluators for 78 distinct tags within several profiles, our approach have achieved 63.8%, 81.1% and 88.4% accuracies selecting respectively the first, second and third top contexts for each particular tag.

The effect that semantic contextualisation of tags in folksonomies describing a single domain (movies in MovieLens, music tracks in Last.fm), and in folksonomies about multiple domains (Web pages in Delicious), does have on personalization and recommendation strategies, together with an exhaustive analysis of the proposed semantic tag similarities, and an empirical comparison of different clustering methods, are some research lines to be addressed.

The distinction of the users' tagging purposes –describing content and context, making subjective opinions, and providing self-references– may be also taken into consideration to enhance the tag disambiguation/contextualization process [5].

Acknowledgements

This work was supported by the Spanish Ministry of Science and Innovation (TIN2008-06566-C04-02), and the Regional Government of Madrid (S2009TIC-1542).

References

1. Angeletou, S., Sabou, M., Motta, E.: Improving Folksonomies Using Formal Knowledge: A Case Study on Search. In: 4th Asian Semantic Web Conference, 276--290. Springer-Verlag (2009)
2. Au Yeung, C. M., Gibbins, N., Shadbolt, N.: Contextualising Tags in Collaborative Tagging Systems. In: 20th Conference on Hypertext and Hypermedia, pp. 251--260. ACM Press (2009)
3. Cantador, I., Bellogín, A., Fernández-Tobías, I., López-Hernández, S.: Semantic Contextualization of Social Tag-based Item Recommendations. In: 12th International Conference on Electronic Commerce and Web Technologies. Springer-Verlag (2011)
4. Cantador, I., Bellogín, A., Vallet, D.: Content-based Recommendation in Social Tagging Systems. In: 4th ACM Conference on Recommender Systems, pp. 237--240. ACM Press (2010)
5. Cantador, I., Konstas, I., Jose, J. M.: Categorising Social Tags to Improve Folksonomy-based Recommendations. Journal of Web Semantics 9(1), pp. 1--15. (2011)
6. García-Silva, A., Szomszor, M., Alani, H., Corcho, O.: Preliminary Results in Tag Disambiguation using DBpedia. In: 1st International Workshop on Collective Knowledge Capturing and Representation (2009)
7. Gemmell, J., Ramezani, M., Schimoler, T., Christiansen, L., Mobasher, B.: The Impact of Ambiguity and Redundancy on Tag Recommendation in Folksonomies. In: 3rd ACM Conference on Recommender Systems, pp. 45--52. ACM Press (2009)
8. Golder, S. A., Huberman, B. A.: Usage Patterns of Collaborative Tagging Systems. Journal of Information Science, 32(2), 198--208 (2006)
9. Markines, B., Cattuto, C., Menczer, F., Benz, D., Hotho, A., Stumme, G.: Evaluating Similarity Measures for Emergent Semantics of Social Tagging. In: 18th International Conference on World Wide Web, pp. 641--650. ACM Press (2009)
10. Newman, M. E. J., Girvan, M.: Finding and Evaluating Community Structure in Networks. Physical Review, E 69, 026113 (2004)
11. Niwa, S., Doi, T., Honiden, S.: Web Page Recommender System based on Folksonomy Mining for ITNG'06 Submissions. In: 3rd International Conference on Information Technology: New Generations, pp.388--393. IEEE Press (2006)
12. Sen, S., Vig, J., Riedl, J.: Tagommenders: Connecting Users to Items through Tags. In: 18th International Conference on World Wide Web, pp. 671--680. ACM Press (2009)
13. Shepitsen, A., Gemmell, J., Mobasher, B., Burke, R. 2008. Personalized Recommendation in Social Tagging Systems using Hierarchical Clustering. In: 2nd ACM Conference on Recommender Systems, pp. 259--266. ACM Press (2008)
14. Weinberger, K. Q., Slaney, M., Van Zwol, R.: Resolving Tag Ambiguity. In: 16th ACM Conference on Multimedia, pp. 111--120. ACM Press (2008)

Social Semantic Web Fosters Idea Brainstorming

Matteo Gaeta[1], Vincenzo Loia[2], Giuseppina Rita Mangione[3], Francesco Orciuoli[1], and Pierluigi Ritrovato[1]

[1] Dipartimento di Ingegneria Elettronica e Ingegneria Informatica
University of Salerno,
Fisciano, Salerno, Italy
[2] Dipartimento di Informatica
University of Salerno,
Fisciano, Salerno, Italy
[3] Centro di Ricerca in Matematica Pura e Applicata (CRMPA)
University of Salerno,
Fisciano, Salerno, Italy

Abstract. Generating and identifying promising ideas represent important challenges for any Enterprise that is focused on knowledge-intensive activities. The generation of new ideas, especially high-quality creative ideas, is vital to business success. Brainstorming is a didactic method that can be exploited to sustain the development of high order skills considered fundamental to foster innovation. On the other side, brainstorming sessions produce new ideas that have to be evaluated and possibly selected. In this paper the Social Semantic Web is exploited in order to define an approach for brainstorming that overcomes the limitations of the existing systems supporting groups in generating ideas. The Semantic Web-based structures organize, correlate and simplify the search for user-generated contents (e.g. ideas). Meanwhile, user-generated contents are analysed in order to elicit non-asserted correlations between them that are used to enrich the aforementioned structures.

Keywords: Social Semantic Web, Brainstorming, SIOC, Knowledge Forum, Knowledge Extraction, Idea Generation, Idea Selection, Innovation

1 Introduction and Motivations

Generating and identifying promising ideas represent recurrent and critical challenges for any Enterprise that is focused on knowledge-intensive activities and innovation. The generation of new ideas, especially high-quality creative ideas, is vital to business success. In order to foster the idea-related processes new strategies and environments to develop High Order Thinking skills (HOT skills) have to be re-thought. Critical thinking, reflection, problem-solving, etc. are fundamental skills for maintaining and improving innovation processes [9]. The research activities on Technology Enhanced Education (TEE), and in particular on Workplace Learning, point on e-Brainstorming as a didactic method guiding a learners' group to learn by progressive argumentation and idea development.

At the same time, e-Brainstorming allows developing and improving the thinking skills by exporting the identified promising ideas in order to further investigate them together with other groups to achieve a solid result in terms of feasibility and originality of the selected ideas. Moreover, e-Brainstorming allows to overcome the production blocking and conformity effect in teamwork [5], by doing so it improves comparison, negotiation and decision-making processes. Some consideration have to be expressed:

- The numerous existing Group Support Systems (GSSs) developed in order to assist people during the idea generation process are based on a vision known as Osborn's conjecture: *if people generate more ideas, then they will produce more good ideas*. Hence, these systems do not take care of the process transforming the quantity into quality with respect to the generation of ideas [14].
- The need for overcoming the limited vision of GSSs has conducted to the *Bounded Ideation Theory* [3] stating that an effective brainstorming model must sustain an iterative process that involves two mains strategies: idea exchange (sharing ideas within a brainstorming group) and generation (accumulating numerous ideas) at the social level and idea expansion (building new ideas starting from existing ones) and selection (identifying of most promising ideas) at the cognitive distributed level [18].
- Despite the brainstorming literature has agreed to support the discovery of connections among different ideas can be significant to effectively support the steps from idea generation (divergent thinking) to idea selection (convergent thinking), there exist few systems that support the automatic discovery of the aforementioned connections [11].

The present work proposes a *Brainstorming Model*, based on the Social Semantic Web approach, that takes care of the Bounded Ideation Theory to overcome the Osborn's conjecture. The used Semantic Web-based structures allow tool interoperability and simplify query and inference operations. On the other hand, the *Brainstorming Model* is based on the most common asynchronous communication/collaboration tool of the Social Web: the Discussion Forum. A language-independent keyphrase extraction algoritm is also applied to support correlation discovery between ideas coming from different groups. The work is organized as follows. In the Section 2 the *Brainstorming Model* is defined on the basis of the *Knowledge Forum Model* by extending Semantic Web-based ontologies. Furthermore, in Section 3.1 an approach, based on a keyphrase extraction algorithm, to automatically discover correlations between ideas coming from more than one brainstorming sessions is illustrated. In Section 4 some conclusion is provided.

2 Extending SIOC for Brainstorming

In this section a *Brainstorming Model* is defined. The approach proposed in the present paper is to exploit the *Knowledge Forum* in order to provide a suitable brainstorming environment. Morever, the defined *Brainstorming Model*

will be described by extending SIOC (Semantically-Interlinked Online Communities) [2]. SIOC is an attempt to link online community sites, to use Semantic Web technologies to describe the information that communities have about their structure and contents, and to find related information and new connections between content items and other community objects. SIOC is based around the use of machine-readable information provided by these sites. The adoption of SIOC provides the following benefits:

- fostering interoperability among different tools (also of different typologies like wikis, blogs, instant messaging, etc.);
- simplifying the link with external data sets, vocabularies, thesauri, folksonomies and with other Semantic Web-based schemes;
- improving and making cheaper the reuse of user-generated content;
- providing a semantic layer to be queried and inferred by using standard languages (SPARQL[4], OWL/OWL2[10]) and reasoners.

2.1 Brainstorming Model Definition

The brainstorming is a problem-solving technique defined by Osborn [12] based on a group discussion led by a moderator. The purpose of a brainstorming session is to make possible the growth of the biggest possible number of ideas about a specific issue. The brainstorming technique is also considered a relevant didactic method. In fact, it can be also classified as an argumentative practice [1]. A strong point of brainstorming is the ability to use the suggestions provided by all participants in the group, so that an idea proposed by a group member can suggest to another a new idea, perhaps more appropriate to reach the best solution. The focus, in the first phase is to produce the greatest number of ideas, which is initially more important than their quality, especially because the greater the number of ideas, the greater the likelyhood of finding some useful. In a second step, which is the more challenging phase of a brainstorming session, ideas should be evaluated, in relation to their effectiveness, selected and developed further. In the proposed approach, a brainstorming session prefigures the presence of a *moderator* while the other *participants* have no specific roles. The topic of discussion has to be not completely defined in order to unleash the power of idea generation, the ideas have to be freely expressed in the initial phase given that quantity is more important than quality at this stage. So, according to our model the brainstorming session consists of three different phases:

- *Activation.* In this phase the issue, on which the discussion has to take place, is presented and the participants have the possibility to socialize.
- *Production.* In this phase the moderator asks participants to speak freely on the subject, urges them to be active, asks questions, rewords questions. The participants freely express ideas, thoughts, opinions. Ideas are not subject to criticism during the meeting, in fact the adverse judgement of ideas must be withheld until later (*deferring judgement* [18]).

[4] http://www.w3.org/TR/rdf-sparql-query/

– *Synthesis.* The moderator summarizes the generated ideas, uses various criteria to stimulate participants to assess and select the best ideas. At this stage combinations and improvements of ideas are seeked. In addition, participants should suggest how the ideas of others can be turned into better ideas or how two ideas can be merged into new ones.

In order to define a digital environment able to support Brainstorming sessions as we have defined them above, the *Knowledge Forum* [4] can be exploited to support the creation and the continuous improvement of knowledge. To facilitate discussion, and therefore the transparency of the communicative intention of each author, the *Knowledge Forum* provides some predefined linguistic structures called *scaffolds*, through which it is possible to identify a set of descriptors of thought (*thinking types*), e.g. `my theory`, `need to understand` and so on.

In our model the use of three different *scaffolds* is proposed in order to sustain the main phases of a brainstorming session: *Idea Generation*, *Knowledge Construction* and *Revision Circle*. The first one covers the *Activation* and the *Production* phases of the Brainstorming session. While, the second one and the third one cover the *Synthesis* phase. Figure 1 shows the list of the *Thinking Types* for each considered scaffold.

Scaffold Name	***Thinking Types***	***Description***
Idea Generation	Issue, **Preliminary Idea**, Advancer, Question, Answer, Free Thought, **Updated Idea**	The moderator introduces the problem issue and provides advancers to smoothly guide the discussion. The participants take part freely in the discussion by proposing new preliminary ideas. The moderator encourages the participants' interventions supporting divergent thinking. Participants can also post free thoughts and update their preliminary ideas. Questions and answers are admitted in this phase in order to support the discussion.
Knowledge Construction	Evaluation, Binding, Explanation, Question, Answer, Example, Warning, **Evolved Idea**	The participants, led by the moderator, assess the ideas on the basis of the criterion of feasibility by describing plausible examples, bringing out the relations among the ideas, organizing ideas according to the identified relations, converge on the most promising ideas and, if necessary, make the ideas evolving. Questions, answers and explanations are admitted in this phase in order to support the discussion.
Revision Circle	Criticism, Promotion, Refinement, Synthesis, Decision, **Packaged Idea**	The moderator and the participants synthetize and refine the most promising ideas by developing convergent thinking (through criticisms, promotions) that brings to a final decision and to a set of packaged ideas.

Fig. 1. Scaffolds and Thinking Types for the proposed Brainstorming Model.

2.2 A SIOC Overview

The **SIOC** initiative aims to enable the integration of online-community information. For instance, users create posts (`sioc:Post`) organized in forums

(`sioc:Forum`), which are hosted on sites (`sioc:Site`). These concepts are sub-classes of higher-level concepts that were added to SIOC: `sioc:Item`, `sioc:Container` and (`sioc:Space`. The `sioc:has_reply` property links reply posts to the content to which they are replying, the `sioc:has_creator` property links user-generated content to its authors, and the `sioc:topic` property points to a resource describing the topic of content items. The SIOC Type module introduces new sub-classes for describing different kinds of Social Web objects in SIOC. In addition, the module points to existing ontologies suitable for describing details on these objects. For instance, a `sioc_t:ReviewArea` might contain reviews asserted by using **Review RDF** [5] that is a domain specific vocabulary used to describe the main properties of a review. The most important classes are `rev:Review`, `rev:Feedback` and `rev:Comment`, while the important properties are `createdOn`, `hasReview`,`rating` and `reviewer`. The link between an instance of a `sioc:Post` and a review (an instance of the `rev:Review` class) is realized by the property `rev:hasReview` (`rdfs:Resource` as range and `rev:Review` as domain). The ReviewRDF scheme is important for the BrainSIOC in order to handle ratings on ideas during the last phase of a brainstorming session (i.e. Synthesis) when the most promising ideas are evaluated, selected and packaged (described more formally).

SIOC can be used in combination with other Semantic Web-based schemes. First of all, **SCOT** (Social Semantic Cloud of Tags) [7] can be used to model tagging operations. SCOT aims to describe the structure and the semantics of *tagging data* and to offer social interoperability for sharing and reusing tag data and representing social relations amongst individuals across different sources. The `scot:Tag` class is used to manage tags. SCOT also enables the modeling of some aspects regarding *who* uses a specific tag. In fact, the property `scot:usedBy` links a tag to a specific user. An instance of `sioc:Post` can be tagged by using the `scot:hasTag` property, or conversely by using the `scot:tagOf` property with domain `scot:Tag` and range `sioc:Item` (a subclass of `sioc:Item`). SCOT can be also integrated with the **MOAT** (Meaning Of A Tag) [6] ontology that provides a mechanism to enrich data regarding tags by considering their *meaning*. Tagging ontologies are particularly useful in the context of BrainSIOC because they improve findability of ideas across brainstorming sessions. Moreover tagging ontologies allow to simply correlate ideas with any kind of user-generated content. The SIOC ontology follows this practice by reusing the **FOAF** vocabulary [7] to describe person-centric data. A person (described by `foaf:Person`) will usually have a number of online accounts (`sioc:UserAccount` that is a sub-class of `foaf:OnlineAccount`) on different online-community sites. FOAF allows to model a social network where persons' profiles are linked together by using the `foaf:knows` property between two instances of `foaf:Person` class. In the end, **SKOS** (Simple Knowledge Organization System)[8] is a Semantic Web

[5] http://vocab.org/review/terms.html
[6] http://moat-project.org/
[7] http://www.foaf-project.org/
[8] http://www.w3.org/TR/ skos- primer/

scheme used to build taxonomies and controlled vocabularies. For the aim of this work, SKOS will be used to model a controlled vocabulary of contexts of interest in a given organization using `skos:narrower` and `skos:broader` properties to relate instances of `skos:Concept`. SKOS can be used in order to construct controlled vocabularies and taxonomies for topics in SIOC to be linked to instances of `sioc:Post` or `sioc:Item` by means of the `sioc:topic` property. SKOS can improve knowledge sharing and correlation processes across different collaboration/communication sessions and tools. By linking FOAF, SIOC, SCOT/MOAT and SKOS it is possible to enrich a person's (a worker in the Enterprise context) profile with the generated ideas, the used tags, etc. in order to foster people search operations.

2.3 The BrainSIOC ontology

The **BrainSIOC** ontology extends the SIOC ontology to support the brainstorming sessions described in Section 2.1 and scaffolds and thinking types illustrated in Figure 1. In order to define the aforementioned extension, several schemes have been considered. In particular, the attention has been focused on *Argumentative Discussion* schemes [17]. Among the others, **IBIS OWL** and **DILIGENT** are relevant for the aims of this work. The IBIS OWL Model is a RDF representation of IBIS, providing URIs for terms regarding argumentations. DILIGENT is primarily a methodology for engineering an ontology; the acronym comes from certain letters in the phrase *DIstributed, Loosely-controlled and evolvInG*. Other interesting works are **Idea Ontology** [15] and **SWAN/SIOC** [17]. The first one introduces an ontology to represent ideas. This ontology provides a common language to foster interoperability between tools and to support the idea life cycle. Through the use of this ontology additional benefits like semantic reasoning and automatic analysis become available. With respect to the aforementioned work, BrainSIOC does not cover the whole idea life cycle management but it proposes a model to represent and support the activities in the context of brainstorming sessions by exploiting a modelling approach similar to those presented in [15]. The second one is a domain-dependent scheme modelling scientific discourses using Semantic Web-based approaches.

First of all, the BrainSIOC ontology considers two roles for the brainstorming activity, i.e. the generic participant and the moderator. In order to model the first one we need to define the `bsioc:Participant` class as a subclass of `sioc:Role`. While the class `bsioc:Moderator` is defined by subclassing `bsioc:Moderator`. An instance of `sioc:UserAccount` is linked to a specific role by using the `sioc:funcion_of` property (its inverse is `sioc:has_function`). The link between a moderator and a specific container (e.g. a forum) can be also asserted by using the `sioc:has_moderator` property with domain `sioc:Forum` and range `sioc:UserAccount`. Furthermore a brainstorming session is modelled by subclassing the `sioc:Forum` class and defining the `bsioc:Brainstorming` in order to reuse all the properties defined for `sioc:Forum`. Figure 2 provides the list of the other classes defined in the BrainSIOC ontology (`bsioc` namespace). In particular, there are correspondences between BrainSIOC classes and both IBIS

Class	Superclass	Subclasses	Description	Phase
bsioc:Advancer	*bsioc:Argument*		An advancer message anticipates the problems or provides additional information that should guide the discussion.	Idea Generation
sioc_t:Answer	*sioc:Post*		The moderator or the participants provide answers to previously asked questions.	Idea Generation, Knowledge Construction
bsioc:Argument	*sioc:Post*	*bsioc:Issue, bsioc:Advancer, bsioc:FreeTought*		
bsioc:Binding	*bsioc:Elaboration*		The participants find and express correlations among ideas.	Knowledge Construction
bsioc:Criticism	*bsioc:Position*		A criticism is the opposite of a promotion for a specifica idea. At this stage, a criticism can be brought out to refine or to reject an idea.	Revision Circle
bsioc:Decision	*sioc:Post*		The moderator takes into account the rating, the refinement and synthesis of ideas and provides a place for a decision (selected or rejected) on any single idea.	Revision Circle
bsioc:Elaboration	*sioc:Post*	*bsioc:Synthesis, bsioc:Binding, bsioc:Refinement*		
bsioc:Evaluation	*bsioc:Justification*		The participants evaluate a preliminary or a updated idea by providing a judgment.	Knowledge Construction
bsioc:EvolvedIdea	*bsioc:Idea*		The participants can make progress with respect to a definition of an idea.	Knowledge Construction
bsioc:Example	*bsioc:Justification*		The participants propose a real world example of an idea in order to demonstrate its feasibility.	Knowledge Construction
bsioc:Explanation	*bsioc:Justification*		The participants give further explanation about an idea, a binding, an example, etc. An explanation could be (or not be) inducted by a question.	Knowledge Construction
bsioc:FreeThought	*bsioc:Argument*		Free thoughts expressed by the participants in order to share intuitions, opinions, insights, etc. that are not yet formalized as ideas.	Idea Generation
bsioc:Idea	*sioc:Post*	*bsioc:PreliminaryIdea, bsioc:UpdatedIdea, bsioc:EvolvedIdea, bsioc:PackagedIdea*		
bsioc:Issue	*bsioc:Argument*		The issue (proposed by the moderator) to be faced in the specific brainstorming session.	Idea Generation
bsioc:Justification	*sioc:Post*	*bsioc:Evaluation, bsioc:Explanation, bsioc:Example*		
bsioc:PackagedIdea	*bsioc:Idea*		Selected ideas are better detailed and formalized to become packaged ideas.	Revision Circle
bsioc:Position	*sioc:Post*	*bsioc:Criticism, bsioc:Promotion*		
bsioc:PreliminaryIdea	*bsioc:Idea*		Preliminary ideas proposed by participants in response to an issue.	Idea Generation
bsioc:Promotion	*bsioc:Position*		A participant can promote a promising idea in order to stimulate other participants to refine it.	Revision Circle
sioc_t:Question	*sioc:Post*		The moderator or the participants ask for clarifications or deepening.	Idea Generation, Knowledge Construction
bsioc:Refinement	*bsioc:Elaboration*		The participants can provide some refinement to a specific idea. Typically, a refinement occurs after a promotion.	Revision Circle
bsioc:Synthesis	*bsioc:Elaboration*		Promotions, criticisms and refinements could be carried out to merge two or more ideas. This operation is realized by provide a synthesis.	Revision Circle
bsioc:UpdatedIdea	*bsioc:Idea*		Modifications to preliminary ideas bring to life updated ideas.	Idea Generation
bsioc:Warning	*sioc:Post*		The moderator brings out some problems or disputes related to the ideas already proposed, the correlations between ideas, and examples provided.	Knowledge Construction

Fig. 2. Classes of the BrainSIOC ontology.

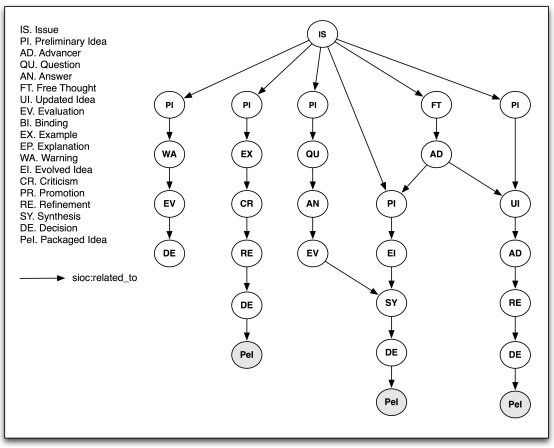


Fig. 3. A sample instance of the BrainSIOC ontology.

OWL and DILIGENT: `sioc_t:Question` is related to IBIS `ibis:Question`, `bsioc:Evaluation` is related to DILIGENT `Evaluation`, `bsioc:Example` is related to DILIGENT `Example`, `bsioc:Decision` is related to IBIS `ibis:Decision`, `bsioc:Idea` is related to IBIS `Idea`. Furthermore, we need to define new properties to be added to the BrainSIOC ontology. In SIOC, there exist several properties that are useful to link instances of `sioc:Item` (and hence of `sioc:Post`) to each other. In particular, the `has_reply` property is used to relate two items, while the `sioc:reply_of` property is its inverse. Both the aforementioned properties are defined as sub-properties of `sioc:related_to` that is adopted in the BrainSIOC. Another useful property is `sioc:next_version` that can be used to link two different versions of the same item. In the end, the `sioc:content` property (with domain `sioc:Item` and range `rdfs:Literal`) is used to store the text representing ideas, questions, answers and so on. Figure 3 illustrates an instance of the BrainSIOC ontology that shows the generation of some ideas in response to a proposed issue. The example illustrates how to the brainstorming takes place across several threads and how ideas evolve step by step until becoming a packaged idea or aborting.

3 Knowledge Discovery in Brainstorming Sessions

In this section, two knowledge discovery modalities in brainstorming sessions are described. The first one deals with discovering correlated ideas across brainstorming sessions. The second one concerns with the capability of BrainSIOC, being based on the Semantic Web stack, to provide high interoperability among people and applications while accessing, retrieving and sharing knowledge in standard way. Figure 4 shows both the modalities also explained in 3.1 and 3.2.

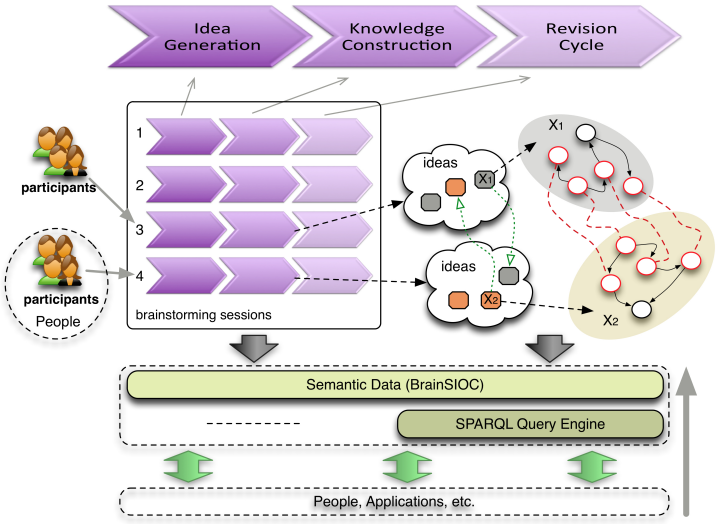


Fig. 4. Knowledge discovery in brainstorming sessions.

In Fig. 4, four brainstorming sessions are considered. For each session there is a group of participants taking part in the brainstorming. The sessions are disjoint except for the *Knowledge Construction* phase where correlations among ideas are discovered (see section 3.1) in order to unlock the independent sessions by providing external stimuli represented by similar ideas coming from other sessions.

3.1 Discovery of correlations among Ideas

In order to satisfy the requirement described in Section 1 regarding the need for correlating ideas, an approach to discover similar ideas across multiple brainstorming sessions (and to suggest these correlations to the participants) is proposed. During the *Knowledge Construction* phase, for a given idea A (an instance of the `bsioc:Idea` class), the literal associated with the `sioc:content` is compared with other ideas coming from other brainstorming sessions. The ideas A_1, A_2, ... , A_n more similar to A are suggested to the participants of the brainstorming sessions where A is emerged (in Figure 4, X_1 and X_2 are similar, so they are respectively suggested to sessions 4 and 3). The proposed approach is based on the application of the **DegExt** algorithm to build a graph representation of a single idea. In order to calculate the similarity, a distance measure that computes the distance between graphs is exploited. A threshold passing value must be considered in order to select only the most similar idea couples.

Furthermore, we suggest to rank the idea couples that pass the threshold using a measure of diversity between the two idea proposers. The bigger the diversity value, the greater the rank value. This approach is supported by scientific and methodological approaches concerning the team building approaches. In particular, in [13] and [6] it is emphasized that highly heterogeneous workgroups (diversity of competencies, skills, knowledge, culture, etc.) are more performant and effective with respect to the idea generation tasks. The diversity measure can be calculated by using the FOAF profiles of the idea proposers and applying some distance measure. The correlations, that are automatically elicited and accepted by participants after a discussion, can be asserted through the use of the new reflexive property `bsioc:correlated_to` that is defined by subclassing the `sioc:related_to` property. DegExt [8] is an unsupervised, graph-based, cross-lingual word and keyphrase extractor. DegExt uses graph representation based on the simple graph-based syntactic representation of text, which enhances the traditional vector-space model by taking into account some structural content features. The simple graph representation provides unlabeled edges representing order-relationship between the words represented by nodes. The stemming and stopword removal operations of basic text preprocessing are executed before constructing the graph. A single vertex is created for each distinct word, even if the word appears more than once in the text. Thus, each vertex label in the graph is unique. Edges represent order-relationships between two terms: there is a directed edge from A to B if an A term immediately precedes a B term in any sentence of the document. The syntactic graph-based representations were shown by Schenker et al. [16] to perform better than the classical vector-space model on several clustering and classification tasks. The most connected nodes in a document graph are assumed by DegExt to represent the keywords. When document representation is complete, every node is ranked by the extent of its connectedness with the other nodes, and the top ranked nodes are then extracted. Intuitively, the most connected nodes represent the most salient words. According to the above representation, words that appear in many sentences that diverge contextually will be represented by strongly connected nodes. DegExt is convenient for the aim of our work because it is relatively cheap in terms of processing time (linear computational complexity) and memory resources while providing nearly the best results for the two above text mining tasks and it does not require training. In order to exploit the result of the DegExt algorithm a distance measure between graphs has to be adopted. In particular, the measure proposed in [16] is considered:

$$dist_{MCS}(G_1, G_2) = 1 - \frac{mcs(G_1, G_2)}{\max(|G_1|, |G_2|)} \tag{1}$$

where G_1 and G_2 are graphs representing ideas (constructed by using DegExt algorithm applied on the `sioc:content` property of instances of the `bsioc:Idea` class), $mcs(G_1, G_2)$ is their maximum common subgraph, max(...) is the standard numerical maximum operation, and |...| denotes the size of the graph that can be taken as the number of nodes and edges contained in the graph. The

computation of *mcs* can be accomplished in polynomial time due to the existence of unique node labels in the considered application. The proposed method provides more accuracy with respect to traditional methods based on numerical feature vectors because it considers the order in which terms appear, where in the document the terms appear, how close the terms are to each other, etc.

3.2 Querying on BrainSIOC

In order to demonstrate the effectiveness of the Semantic Web stack to model, represent and integrate data, a simple SPARQL query able to find, across all brainstorming sessions, all packaged ideas annotated with the tag "Social Web"is listed here.

```
select ?title, ?content, ?topic
where
 {
    ?s a bsioc:PackagedIdea.
    optional { ?s dc:title ?title }.
    ?s sioc:content ?content .
    optional { ?s sioc:topic ?topic .
                  ?topic rdf:type skos:Concept .
                  ?topic skos:prefLabel "Social Web" }
  }
```

In particular, the above query foresees the use of the Dublin Core[9] property namely `dc:title` and the use of SKOS to define a shared (across all brainstorming sessions) controlled vocabulary in order to tag the posts. Moreover, this simple query envisages the capability of BrainSIOC to enable the integration of brainstorming sessions with collaborative working and learning scenarios in order to foster and improve knowledge maturing and knowledge sharing processes within the Organizations.

4 Conclusions and Future Works

This work proposes an approach consisting in *i*) a novel Brainstorming Model implemented by extending the SIOC ontology and defining BrainSIOC, *ii*) a technique based on the application of the DegExt algorithm to automatically discover correlations among ideas across multiple brainstorming sessions. The approach will be experimented and exploited in the ARISTOTELE project (which also foresees the development of a tool implementing the BrainSIOC) by also considering the competencies that may be developed by the participants during brainstorming sessions.

Acknowledgement

This research is partially supported by the EC under the Project ARISTOTELE "Personalised Learning & Collaborative Working Environments Fostering Social Creativity and Innovations Inside the Organisations", VII FP, Theme ICT-2009.4.2 (Technology-Enhanced Learning), Grant Agreement n. 257886.

[9] http://dublincore.org

References

1. Bonaiuti, G., Calvani, A., Ranieri, M.: Fondamenti di didattica. Teoria e prassi dei dispositivi formativi. Carocci (2007)
2. Breslin, J., Bojars, U.: SIOC Project Homepage (2008), http://sioc-project.org/
3. Briggs, R., Reinig, B.: Bounded Ideation Theory: A New Model of the Relationship Between Ideaquantity and Idea-quality during Ideation. In: 2007 40th Annual Hawaii International Conference on System Sciences HICSS07. pp. 16–16. Ieee (2007)
4. Chen, B., Chuy, M., Resendes, M., Scardamalia, M.: "Big Ideas Tool" as a New Feature of Knowledge Forum. In: 2010 Knowledge Building Summer Institute. Toronto, Canada (2010)
5. Girotra, K., Terwiesch, C., Ulrich, K.T.: Idea Generation and the Quality of the Best Idea. Manage. Sci. 56(4), 591–605 (Apr 2010)
6. Kerr, D.S., Murthy, U.S.: Divergent and Convergent Idea Generation in Teams: A Comparison of Computer-Mediated and Face-to-Face Communication. Group Decision and Negotiation 13(4), 381–399 (2004)
7. Kim, H.L., Yang, S.K., Song, S.J., Breslin, J.G., Kim, H.G.: Tag Mediated Society with SCOT Ontology. Business (2007)
8. Litvak, M., Last, M., Aizenman, H., Gobits, I., Kandel, A.: DegExtA Language-Independent Graph-Based Keyphrase Extractor. Advances in Intelligent and Soft Computing 86, 121–130 (2011)
9. Miri, B., David, B.C., Uri, Z.: Purposely Teaching for the Promotion of Higher-order Thinking Skills: A Case of Critical Thinking. Research in Science Education 37(4), 353–369 (2007)
10. Motik, B.: OWL 2 Web Ontology Language Document Overview (2009), http://www.w3.org/TR/2009/REC-owl2-overview-20091027/
11. Nijstad, B.A., Stroebe, W.: How the Group Affects the Mind: A Cognitive Model of Idea Generation in Groups. Personality and Social Psychology Review 10(3), 186–213 (2006)
12. Osborn, A.F.: Applied Imagination: Principles and Procedures of Creative Problem-Solving 3rd Edition. Creative Education Foundation (1993)
13. Pissarra, J., Jesuino, J.C.: Idea generation through computer-mediated communication: The effects of anonymity. Journal of Managerial Psychology 20(3/4), 275–291 (2005)
14. Reinig, B.A., Briggs, R.O.: On The Relationship Between Idea-Quantity and Idea-Quality During Ideation. Group Decision and Negotiation 17(5), 403–420 (2008)
15. Riedl, C., May, N., Finzen, J., Stathel, S., Kaufman, V., Krcmar, H.: An Idea Ontology for Innovation Management. International Journal on Semantic Web and Information Systems 5(4), 1–18 (2009)
16. Schenker, A., Last, M., Bunke, H., Kandel, A.: Classification of Web Documents Using a Graph Model. In: Seventh International Conference on Document Analysis and Recognition 2003 Proceedings. pp. 240–244. No. Icdar, IEEE Comput. Soc (2003)
17. Schneider, J., Passant, A., Groza, T., Breslin, J.G.: Argumentation 3.0: how Semantic Web technologies can improve argumentation modeling in Web 2.0 environments pp. 439–446 (2010)
18. Yuan, S.T., Chen, Y.C.: Semantic Ideation Learning for Agent-Based E-Brainstorming. IEEE Transactions on Knowledge and Data Engineering 20(2), 261–275 (2008)

Recommending #-Tags in Twitter

Eva Zangerle, Wolfgang Gassler, Günther Specht

Databases and Information Systems
Institute of Computer Science
University of Innsbruck, Austria
firstname.lastname@uibk.ac.at

Abstract. Twitter, currently the most popular microblogging tool available, is used to publish more than 140,000,000 messages a day. Many users use hashtags to categorize their tweets. However, hashtags are not restricted in any way in terms of usage or syntax which leads to a very heterogeneous set of hashtags occurring in the Twitter universe and therefore, decreases the search capabilities. In this paper, we present an approach for the recommendation of highly appropriate hashtags to the user during the creation process. The recommendations aim at encouraging the user to (i) use hastags at all, (ii) use more appropriate hashtags and (iii) avoid the usage of synonymous hashtags. Therefore the vocabulary of hashtags becomes more homogenous regarding both syntax and semantics.

1 Introduction

Social networks have gained significant importance on the web throughout the last years. The most popular microblogging tool, Twitter, has experienced tremendous success lately and has become very important as both a social network and a news media [13]. Twitter enables all registered users to post 140-character messages and follow other users. The users's personal timeline (home-view on the Twitter universe) basically includes all messages – the so-called tweets – of all followed users. The notion of a follower describes a user who follows another user. Vice versa, the notion of a followee describes a user who is followed by another user. Such a connection between users is not reciprocal - user A can follow any other user B without requiring user B to follow user A back. Additionally, all messages are fully accessible to the public. Nowadays, 140,000,000 messages - so-called *tweets* - are posted every day. As reported by Twitter[1], every day more than 400,000 new users join the Twitter network.
The basic motivation of users to join Twitter and participate is manyfold [11]. Millions of users use Twitter to keep track of friends and keep friends updated. Users may seek for advice on certain problems or participate in general discussions about certain topics. Some participants follow celebrities or companies in order to stay updated. Many of the active users - those who are not just following other users, but are also actively posting tweets - use Twitter as a medium to

[1] http://blog.twitter.com/2011/03/numbers.html

let the world know what they're up to or simply to share some information they consider useful. The probably most important feature of Twitter is the retweet functionality. It enables users to further broadcast tweets they consider worth spreading within the Twitter network. Mostly, the retweeted message remains unchanged. A retweeted message contains "`RT: @originaluser`" followed by the original message. This retweeting, which was also heavily analysed in [13], can spread an important message all over the world within minutes. Due to the ever increasing amount of Twitter messages and the resulting chaos within the Twittersphere, the microblogging community started to use so-called hashtags as a means for the manual categorization of tweets. The categorization can be used for either searching for certain topics based on the used hashtags or to be able to follow certain conversations about a certain topic on Twitter. The only requirement for hashtags is to start with a hash symbol #. Besides this fact, hashtags do not have to conform to any rules or regulations and can be seen as typcial tags as used in common Web 2.0 applications like e.g. Blogs. Hashtags may appear at any arbitrary position within the message and may consist of any arbitrary combination of characters. This makes them easy to use, but at the same time leads to a significant lack of structure and uniformity. During our research, we crawled a data set and analyzed it. In the process we found that that users utilized very different popular hashtags for their tweets about the same topic. For example, the Tour de France (a world-famous bicycle race in France) was very popular. Tweets about this topic contain different hashtags, such as `#tdf, #tourdefrance, #cycling` or `#procycling`. Twitter offers its users a search engine which is able to search for keywords, but also for hashtags. Therefore, when searching for discussions about the Tour de France by using the search hashtag `#tourdefrance`, the user might not be able to retrieve all tweets containing information about the Tour de France. This is due to the fact that other users used the hashtag `#tdf`, which the user did not specify in the search query. Certainly, tweets containing the hashtag `#tdf` would also have been a perfect match for the user's query. However, due to the heterogeneous hashtag vocabulary used by the active Twitter community, many synonymous hashtags are used for describing the same semantic information.

In this paper we introduce an approach for the recommendation of hashtags. Our approach computes recommendations based on an analysis of existing tweets by other users and recommends suitable hashtags for the currently entered message to the user. This recommendation mechanism aims at encouraging the user to make use of hashtags and creating a more homogeneous hashtag vocabulary in order to enhance the quality of search result. Additionally, we present general statistics about the use of hashtags within Twitter and an evaluation of our approach.

The remainder of this paper is organized as follows. Section 2 describes the basic concepts of Twitter and hashtags. Section 3 is concerned with the process of hashtag recommendations. Section 4 contains the experiments and evaluations of the presented approach. Subsequently, Section 5 describes important related work and Section 6 concludes the paper.

2 Hashtags

Hashtagging is a simple and convenient way for users to categorize their own tweets. Such a hashtag within a tweet can simply be specified by adding a hash - '#' - followed by the tag itself. One tweet may also contain multiple hashtags, like in the following example tweet: "`Don't forget! Only 7 days till the #SASWeb submission deadline #umap2011 http://bit.ly/dKgS82. #recsys #um #adaptivity #web3.0 #ontologies`" which was posted by the SASWeb workshop (@sasWeb2011).

The most popular hashtags are either related to long-term popular topics or to current events or topics, e.g. the hashtag `#tdf` was extensively used during the crawling period as the Tour de France was taking place during this time. Typical long-term topics are e.g. `#Apple` or `#Obama` which are featured in thousands of messages a day [13].

2.1 Data Set and Hashtag Analysis

In order to be able to analyse the hashtagging behaviour of Twitter users and to build up a database which forms the basis for all recommendation computations, we had to crawl tweets. Overall, we collected about 16,000,000 tweets from July 2010 until February 2011 via the Twitter Application Programming Interface.

In order to retrieve a diverse and highly representative data set to base our evaluations and analysis on, we decided to use Twitter's API[2]. The basis for our search queries was an English dictionary containing more than 32,000 words. We iterated over the words contained in the dictionary and used them as search keywords for the Twitter Search API. All search results were stored whereby only tweets containing hashtags were used for further analysis. Another approach was to retrieve the public timeline, which basically consists of the ten latest tweets. The timeline is displayed on the Twitter website and is also available via the API. However, these tweets are only updated once a minute. Therefore, only 600 tweets could be retrieved per hour and considering the fact that only 20% of all tweets contain hashtags, this approach was not feasible for crawling a sufficiently large dataset.

After having crawled the data, we had to perform multiple preprocessing steps. This included removing all non-english messages (based on Twitter's language classification mentioned in the metadata of every tweet) and all messages not containing hashtags at all. Furthermore, all messages were transformed to lower-case. Table 1 contains an overview about the crawled data set and its characteristics. Out of the crawled tweets, more than 3 million tweets contained at least one hashtag, which marks 20% of all crawled tweets. The hashtags filtered from all tweets were further analysed in regards to their usage and popularity. Figure 1 displays the long tail distribution of hashtags and their usage. The fact that stands out about this distribution is that 86% of all hashtags within the data set were used within less than five tweets. On the other

[2] `http://search.twitter.com/search`

hand, the most popular hashtags within the data set (`#jobs`, `#nowplaying`, `#zodiacfacts`, `#news` and `#fb`) were used in 8% of all messages containing hashtags. Another interesting fact is the distribution of the number of hashtags used per tweet which can be seen in Figure 2. We expected the number of hashtags per message to be decreasing steadily. This is mostly the case for messages contains less than 15 hashtags. However, the sudden amplitude at 17 hashtags per message is somewhat surprising. We therefore examined these messages and discovered that these were spam tweets which only contained hashtags and a URL, like e.g. "`RT @Bhupesh_tweet: #Quad #loop-http://bit.ly/ciHX2U #retweet #India #Jobs #World #news #canada #ad #win #USA #tdf #oea #hacking #icantstop #sdcc #game`". Such tweets typically also feature a high retweet-rate by using a spam network consisting of many Twitter users created for spam purposes.

Characteristic	**Value**	**Percentage**
Crawled messages total	16,034,195	100%
Messages containg at least one hashtag	3,209,281	20%
Messages containing no hashtags	12,824,914	80%
Retweets	2,556,617	16%
Direct messages	3,073,948	19%
Hashtags usages total	5,097,545	–
Hashtags distinct	510,170	–
Average number of hashtags per message	1.5884	–
Maximum number of hashtags per message	23	–
Hashtags occurring < 5 times in total	437,266	–
Hashtags occurring < 3 times in total	328,348	–
Hashtags occuring only once	384,187	–

Table 1. Overview about the Crawled Tweets

3 Hashtag Recommendations

The aim of the approach presented in this paper is to find a set of hashtags suitable for any tweet the user enters. These hashtags are then recommended to the user during the creation process of the new tweet. Recommendations are basically be computed by performing the following steps:

1. finding the most similar messages in the crawled data set for the tweet just entered by the user

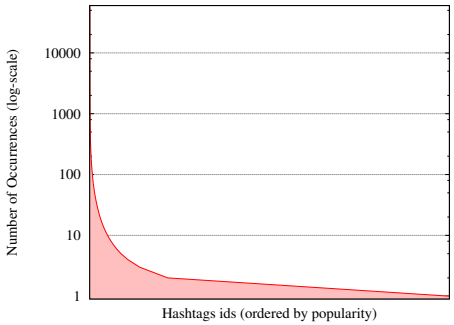


Fig. 1. Long tail of hashtag popularity

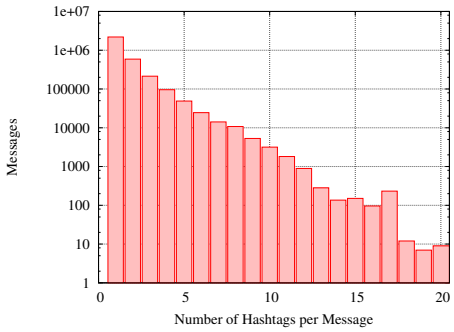


Fig. 2. Number of hashtags per message

2. retrieving the set of hashtags used within these most similar messages
3. ranking the computed set of hashtag recommendation candidates

These steps for the computation of hashtag recommendations are discussed in the following sections.

3.1 Similarity of Tweets

In order to be able to determine similar tweets, a similarity measure for the comparison of two at most 140 character long messages has to be introduced. This metric is used to rank the results gathered from searching similar tweets within the crawled data set. These similar messages are subsequently considered to contain valuable hashtag recommendation candidates. A straightforward solution is to use the term frequency - inverse document frequency measure for the comparison of tweets. In order to be able to use tf/idf for the computation of the similarity of tweets, the formula stated in Equation (1) is used.

$$tf\_idf_{t,d} = tf_{t,d} * idf_t \tag{1}$$

$$tf_{t,d} = n_{t,d} \tag{2}$$

$$idf_t = log \frac{|D|}{|\{d : t \in d\}|} \tag{3}$$

In the case of searching a set of tweets, the set D of documents which have to be searched is the set of tweets in the system. The term frequency basically is the number of occurrences of a term t within a given document d (tweet). The inverse document frequency (idf) constitutes the importance of a term t within the whole set of documents which are searched. This is computed by taking the number of all documents ($|D|$) within the index and dividing it by the number of documents which contain the searched term ($|\{d : t \in d\}|$). The computation of the tf/idf measure for a given search query (in our case the tweet inserted by the user), is subsequently accomplished by computing the sum of all tf/idf of all terms t occurring within the search query d: $\sum_{t\ in\ d} tf\_idf(t)$. Futhermore, the final score is increased if more of the terms of the query are matched. The final set of similar tweets (those obtaining the highest tf/idf-based score ratings) is restricted to a set of tweets having a score above a certain treshold corresponding to the total number of results and the specified limit of total results.

3.2 Ranking

After having obtained the set of the most similar messages to the tweet the user just entered, the hashtags are extracted from these tweets. These hashtags are referred to as hashtag recommendation candidates throughout the remainder of this paper. The ranking of these hashtag recommendation candidates is crucial for the success of recommendations. This is due to the fact that both the cognition of the user and the space available for displaying the recommendations is limited. In most cases a set of 5-10 recommendations is most appropriate which also correspond to the capacity of short-term memory (Miller, 1956). Therefore the top-k recommendations are shown to the user, where k denotes the size of the set of recommended hashtags presented to the user. This restricted set is based on the set of all hashtags which were extracted from the most similar messages to the newly created tweet. To present the most suitable top-k hashtags to the user, the recommendation candidates have to be ranked. For our approach, we evaluated three ranking methods, which can be summarized as follows:

- *OverallPopularityRank:* This ranking approach is based on the popularity of the hashtag recommendation candidates. It basically considers the number of occurrences of the respective hashtag within our data set. The more popular a hashtag is overall, the higher the resulting rank of the hashtag.
- *RecommendationPopularityRank:* This ranking method basically counts the occurrences of each hashtag within the set of recommendation candidates. The higher the number of occurrences, the more (similar) messages contain this hashtag. Therefore, it is likely that the hashtag is suitable for the tweet the user just entered.

– *SimilarityRank:* This ranking method is based on the similarity value between the tweet entered by the user and the tweet which provides a hashtag recommendation candidate. The more similar the messages are, the more likely it is that the hashtags contained in this similar message are suitable for the tweet entered by the user. In the case that multiple tweets contain the hashtag which has to be ranked, the similarity of the most similar tweet is used. As a metric for the similarity of tweets, we used tf/idf as described in 3.1.

4 Evaluation

A recommendation engine prototype implementing this approach has been developed based on Apache's Lucene[3] fulltext index. We used the fulltext index to store the crawled tweets which enabled us to find the most similar messages by using Lucene's Search Index.

4.1 Test Setup

The evaluation was done on a CentOS release 5.1 machine with 8 GB of RAM. The evaluation of the hashtag recommendation approaches was conducted by performing a leave-one-out test. This test was based on the data set described in Section 2.1. Based on the crawled data set, we built a fulltext index comprising all 3.2 mil. cleaned messages without hashtags of this data set. From this index, we randomly chose 10,000 messages with less than six hashtags for each test run. For each of these messages, the contained hashtags were removed from the message and the resulting string was used as the input tweet for the recommendation engine. Naturally, the currently used tweet was removed from the Lucene Index and was not considered for the computation of recommendation candidates. Additionally, no retweets were used as test input tweets as search for similar messages would return an identical retweeted message which would obviously distort the evaluation results.

Based on the hashtag recommendations computed by the recommendation engine, we evaluted the three ranking methods described in 3.2. This was done by computing the precision and recall values of the top-k recommendations with $k = 1$, $k = 2$,..., $k = 10$ as described in the next section.

4.2 Precision and Recall

For the evaluation of the quality of the computed recommendations, we chose to use the precision and recall values of the recommendations. These metrics are defined as follows:

$$precision\ (\mathcal{H}_{rec}) = \frac{|\mathcal{H}_{rec} \cap \mathcal{H}_{orig}|}{|\mathcal{H}_{rec}|} \tag{4}$$

[3] `http://lucene.apache.org/`

$$recall\,(\mathcal{H}_{rec}) = \frac{|\mathcal{H}_{rec} \cap \mathcal{H}_{orig}|}{|\mathcal{H}_{orig}|} \tag{5}$$

where $\mathcal{H}_{original}$ is the set of original hashtags which were removed from the original tweet and $\mathcal{H}_{recommended}$ is the set of top-k recommendations. We performed ten test runs for each ranking method with $k = 1$, $k = 2$,..., $k = 10$. Each test run computed the respective average recall and precision value of 10,000 test tweets. Thus, the evaluation is based on the computation of 100,000 top-k recommendation sets for each ranking method.

4.3 Results

The experiments conducted showed that the approach is feasible of recommending suitable hashtags. The recall values for the top-k recommended hashtags can be seen in Figure 3. In this figure, the recall values for k (the number of recommended hashtags) being between 1 and 10 has been evaluated for the three considered ranking methods. This Figure shows that ranking based on the overall popularity of the hashtag (OverallPopularityRank) and also based on the popularity of the hashtag within the hashtag recommendation candidates (RecommendationPopularityRank) do not perform well. In contrast, SimilarityRank (ranking based on the similarity of the original tweet and the tweet containing the recommendation candidate) is able to perform significantly better. This ranking method leads to promising recall values which are well above the 40% mark for $k > 2$.

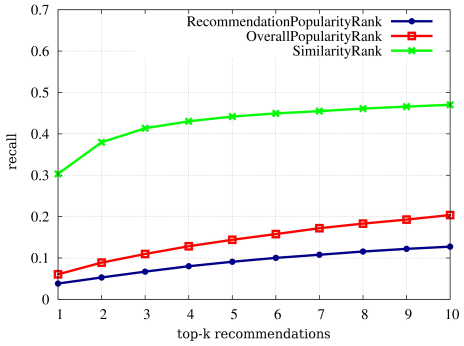


Fig. 3. Recall depending on number of Recommendations

The precision values for the computed recommendation sets decrease with an increasing k. This is due the fact, that we only use test tweets with at most 5 hashtags per message. Therefore even a set of 10 recommendations featuring a recall value of 100% only results in a precision of 50% as five of the ten

recommended hashtags are not applicable as the original message only features five hashtags.

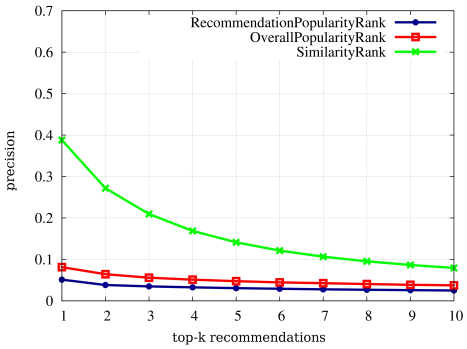


Fig. 4. Precision depending on number of Recommendations

Overall, the evaluations showed that our approach is suitable for the recommendation of hashtags. Another fact which can be derived from the evaluations is that our approach shows the best performance when restricting the set of recommended hashtags to $k = 5$, as the recall value does not improve much with additional recommendations and the precision value is still reasonable.

5 Related Work

The recommendation of Twitter hashtags can benefit from various other fields of research. These areas are (i) tagging of online resources, (ii) traditional recommender systems, (iii) social network analysis and (iv) Twitter analysis. However, to the best of our knowledge, there is no other approach aiming at recommending hashtags to Twitter users.

The recommendation of tags of online resources like images, bookmarks or bibliographic entries is directly related to our approach. Such approaches can be based on the co-occurrence of tags, like e.g. in [14, 20]). The notion of co-occurrence of tags describes the fact that two tags are used to tag the same photo. Therefore, only partly tagged photos can be subject to tag recommendations. Based on these relatively simple approaches, the paper by Rae *et al.* [17] proposes a method for Flickr tag recommendations which takes different contexts into account. Rae distinguishes four different contexts for the computation of recommendations: (i) the user's previously used tags, (ii) the tags of the user's contacts, (iii) the tags of the users which are members of the same groups as the user and (iv) the collectively most used tags by the whole community. A similar approach has also been facilitated by Garg and Weber in [6]. Another example for recommendations of tags is based on the BibSonomy platform which

basically allows its users to tag bibliographic entries [14]. This approach extracts tags which might be suitable for the entry from the title of the entry, the tags previously used for the entry and tags previously used by the current user. Based on these resources, the authors propose different approaches for merging these sets of tags. The resulting set is subsequently recommended to the user. Jäschke *et al.* [10] propose a collaborative filtering approach for the computation of tag recommendations. This computation is based on a graph consisting of the users, their tags and the tagged resources. After having constructed this graph, a PageRank-like ranking algorithm (called FolkRank) is applied. Furthermore, [2,15] are mainly concerned with the motivation of users to tag resources. John Hannon *et al.* [7] developed the Twittomender system which facilitates an approach for the recommendation of followees. This is done by creating profiles of users and applying a collaborative filtering approach to these profiles. The Twittomender system also provides search functionality (based on arbitrary keywords) which returns profile information about the found users like e.g. the latest popular keywords used by the specific user or his latest tweet.
Another approach directly connected to Twitter and recommendations is described by Phelan *et al.* [16]. In this approach, Twitter is used for the recommendation of news articles. In particular, Twitter is used to rank the news stories originating from various RSS feeds based on the user's tweets, the user's friends tweets or the public most recent tweets. Also, Jilian Chen *et al.* [5] focused on recommendations based on tweets. In this case, interesting URLs are recommended to the user. Romero *et al.* [19] analyzed how hashtags spread within the Twitter Universe. The hashtags were analyzed with regards to how a hashtag might be used by a user who is exposed to this hashtag by his followers and followees. The authors categorized the top-500 hashtags used within their data set and found that the adoption of hashtags is dependent on the category of the hashtags. E.g. multiple exposure to a hashtag for political or sports topics lead to the adoption of the hashtag with a higher probability than in any other hashtag category.
Kwak *et al.* [13] did a thorough analysis of the Twitter universe focusing on information diffusion within the network. Further analysis of Twitter messages are also contained in [3,11,12,21]. There have been numerous papers throughout the last years addressing the social aspects of Twitter and social online networks in general. Huberman *et al.* [9] found that the Twitter network basically consists of two networks: one dense network consisting of all followers and followees and one sparse network consisting of the actual friends of users. Huberman defines a friend of a user as another Twitter user with whom the user exchanged at least two directed messages. [4] contains an analysis of the retweet messages and [8] is concerned with how Twitter might be suitable for collaboration by exchanging direct messages.
As for the recommender system facilitated in our approach, many publications are focused around collaborative filtering. The papers by Resnick [18] and Adomavicius [1] provide a very good overview about the field of collaborative filtering.

6 Conclusion

In this paper, we presented an approach for the recommendation of hashtags within the Twitter microblogging application. The presented algorithm is based on the analysis of similar tweets and the hashtags contained in these tweets. Our evolutions were based on a self-crawled data set consisting of 12 million tweets. The preliminary evaluations showed promising results as the recall values of the recommendations are about 45-50%. Future work will include integrating the social graph of Twitter users for the recommendation. Furthermore, the ranking of hashtag recommendation candidates is also subject to further research and improvements. The enhancement of the recommendations of synonymous hashtags based on a semantic analysis for the exclusion of synonymous hashtags and their recommendation is also part of future work.

References

1. G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749, 2005.
2. M. Ames and M. Naaman. Why we tag: motivations for annotation in mobile and online media. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '07, pages 971–980, New York, NY, USA, 2007. ACM.
3. S. Asur, B. Huberman, G. Szabo, and C. Wang. Trends in Social Media: Persistence and Decay. *Arxiv preprint arXiv:1102.1402*, 2011.
4. D. Boyd, S. Golder, and G. Lotan. Tweet, tweet, retweet: Conversational aspects of retweeting on twitter. In *hicss*, pages 1–10. IEEE Computer Society, 1899.
5. J. Chen, R. Nairn, L. Nelson, M. Bernstein, and E. Chi. Short and tweet: experiments on recommending content from information streams. In *Proceedings of the 28th international conference on Human factors in computing systems*, pages 1185–1194. ACM, 2010.
6. N. Garg and I. Weber. Personalized, interactive tag recommendation for flickr. In *Proceedings of the 2008 ACM conference on Recommender systems*, RecSys '08, pages 67–74, New York, NY, USA, 2008. ACM.
7. J. Hannon, M. Bennett, and B. Smyth. Recommending twitter users to follow using content and collaborative filtering approaches. In *RecSys '10: Proceedings of the fourth ACM conference on Recommender systems*, pages 199–206, New York, NY, USA, 2010. ACM.
8. C. Honeycutt and S. C. Herring. Beyond microblogging: Conversation and collaboration via twitter. In *HICSS*, pages 1–10. IEEE Computer Society, 2009.
9. B. Huberman, D. Romero, and F. Wu. Social networks that matter: Twitter under the microscope. *First Monday*, 14(1):8, 2009.
10. R. Jaeschke, L. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme. Tag Recommendations in Folksonomies. In J. Kok, J. Koronacki, R. Lopez de Mantaras, S. Matwin, D. Mladenic, and A. Skowron, editors, *Knowledge Discovery in Databases: PKDD 2007*, volume 4702 of *Lecture Notes in Computer Science*, pages 506–514. Springer Berlin / Heidelberg, 2007.
11. A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st*

SNA-KDD 2007 workshop on Web mining and social network analysis, pages 56–65. ACM, 2007.
12. B. Krishnamurthy, P. Gill, and M. Arlitt. A few chirps about twitter. In *Proceedings of the first workshop on Online social networks*, pages 19–24. ACM, 2008.
13. H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600. ACM, 2010.
14. M. Lipczak and E. Milios. Learning in efficient tag recommendation. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 167–174, New York, NY, USA, 2010. ACM.
15. C. Marlow, M. Naaman, D. Boyd, and M. Davis. HT06, tagging paper, taxonomy, Flickr, academic article, to read. In *Proceedings of the seventeenth conference on Hypertext and hypermedia*, page 40. ACM, 2006.
16. O. Phelan, K. McCarthy, and B. Smyth. Using twitter to recommend real-time topical news. In *Proceedings of the third ACM conference on Recommender systems*, pages 385–388. ACM, 2009.
17. A. Rae, B. Sigurbjörnsson, and R. van Zwol. Improving tag recommendation using social networks. In *Adaptivity, Personalization and Fusion of Heterogeneous Information*, RIAO '10, pages 92–99, Paris, France, France, 2010. Le Centre de Hautes Etudes Internationales d'Informatique Documentaire.
18. P. Resnick and H. Varian. Recommender systems. *Communications of the ACM*, 40(3):58, 1997.
19. D. M. Romero, B. Meeder, and J. M. Kleinberg. Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter. In S. Srinivasan, K. Ramamritham, A. Kumar, M. P. Ravindra, E. Bertino, and R. Kumar, editors, *WWW*, pages 695–704. ACM, 2011.
20. B. Sigurbjörnsson and R. Van Zwol. Flickr tag recommendation based on collective knowledge. In *Proceeding of the 17th international conference on World Wide Web*, pages 327–336. ACM, 2008.
21. S. Ye and S. Wu. Measuring Message Propagation and Social Influence on Twitter. com. In *Social Informatics: Second International Conference, Socinfo 2010, Laxenburg, Austria, October 27-29, 2010, Proceedings*, page 216. Springer-Verlag New York Inc, 2010.