Automatic Categorization of Tags in Collaborative Environments

Qihua Wang[1,⋆], Hongxia Jin[2], and Stefan Nusser[2]

[1] Purdue University, West Lafayette, Indiana
[2] IBM Almaden Research Center, San Jose, California

Abstract. Tagging allows individuals to use whatever terms they think are appropriate to describe an item. With the growing popularity of tagging, more and more tags have been collected by a variety of applications. An item may be associated with tags describing its different aspects, such as appearance, functionality, and location. However, little attention has been paid in the organization of tags; in most tagging systems, all the tags associated with an item are listed together regardless of their meanings. When the number of tags becomes large, finding useful information with regards to a certain aspect of an item becomes difficult. Improving the organization of tags in existing tagging systems is thus highly desired. In this paper, we propose a hierarchical approach to organize tags. In our approach, tags are placed into different categories based on their meanings. To find information with respect to a certain aspect of an item, one just needs to refer to its associated tags in the corresponding category. Since existing applications have already collected a large number of tags, manually categorizing all the tags is infeasible. We propose to use data-mining and machine-learning techniques to automatically and rapidly classify tags in tagging systems. A prototype of our approaches has been developed for a real-word tagging system.

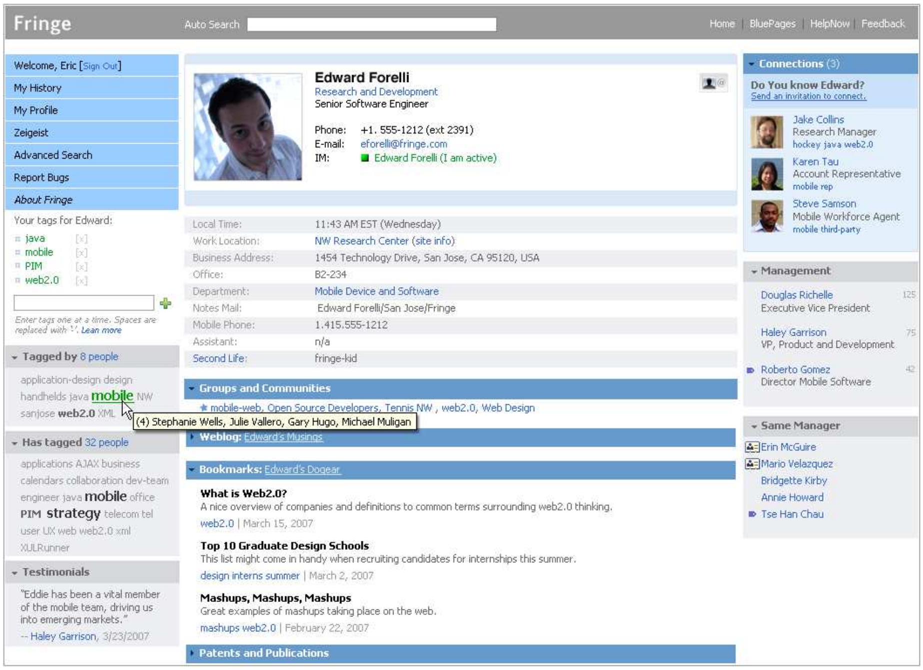
Keywords: Tagging, Web 2.0, Social Network, Categorization, Machine Learning.

1 Introduction

Tagging has gained popularity as a lightweight and flexible approach to classifying and retrieving information. It enables individuals to use whatever terms they think are appropriate to describe an item without the burden of selecting the term from a pre-defined taxonomy. Tagging has been applied in a variety of applications ranging from desktop applications for photo organization (F-Spot [3]) to email systems (Gmail [6]). Tagging becomes most compelling when it is used in a collaborative environment, where tags from different people can be aggregated and combined. This social tagging approach has been used to manage bookmarks (Del.icio.us), images (Flickr), and products (Amazon.com).

In addition to characterizing items, tagging has also been applied to describing people. Fringe Contacts (or Fringe for short) [4] is a reference system designed to augment

⋆ This work was done when the author was an intern at IBM Almaden Research Center.


Fringe | Auto Search | Home | BluePages | HelpNow | Feedback

Welcome, Eric [Sign Out]
My History
My Profile
Zeigeist
Advanced Search
Report Bugs
About Fringe

Your tags for Edward:
java [x]
mobile [x]
PIM [x]
web2.0 [x]

Enter tags one at a time. Spaces are replaced with '_'. Learn more

Tagged by 8 people
application-design design handhelds java mobile NW sanjose web2.0 XML
(4) Stephanie Wells, Julie Vallero, Gary Hugo, Michael Muligan

Has tagged 32 people
applications AJAX business calendars collaboration dev-team engineer java mobile office PIM strategy telecom tel user UX web web2.0 xml XULRunner

Testimonials
"Eddie has been a vital member of the mobile team, driving us into emerging markets."
-- Haley Garrison, 3/23/2007

Edward Forelli
Research and Development
Senior Software Engineer

Phone: +1. 555-1212 (ext 2391)
E-mail: eforelli@fringe.com
IM: 🟩 Edward Forelli (I am active)

Local Time:	11:43 AM EST (Wednesday)
Work Location:	NW Research Center (site info)
Business Address:	1454 Technology Drive, San Jose, CA 95120, USA
Office:	B2-234
Department:	Mobile Device and Software
Notes Mail:	Edward Forelli/San Jose/Fringe
Mobile Phone:	1.415.555-1212
Assistant:	n/a
Second Life:	fringe-kid

Groups and Communities
mobile-web, Open Source Developers, Tennis NW, web2.0, Web Design

Weblog: Edward's Musings

Bookmarks: Edward's Dogear

What is Web2.0?
A nice overview of companies and definitions to common terms surrounding web2.0 thinking.
web2.0 | March 15, 2007

Top 10 Graduate Design Schools
This list might come in handy when recruiting candidates for internships this summer.
design interns summer | March 2, 2007

Mashups, Mashups, Mashups
Great examples of mashups taking place on the web.
mashups web2.0 | February 22, 2007

Patents and Publications

Connections (3)
Do You know Edward?
Send an invitation to connect.

Jake Collins
Research Manager
hockey java web2.0

Karen Tau
Account Representative
mobile rep

Steve Samson
Mobile Workforce Agent
mobile third-party

Management
Douglas Richelle 125
Executive Vice President
Haley Garrison 75
VP, Product and Development
Roberto Gomez 42
Director Mobile Software

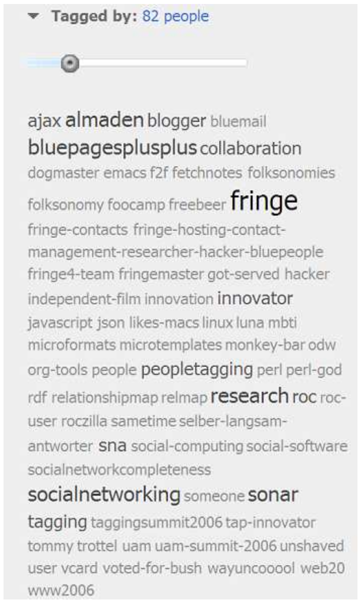
Same Manager
Erin McGuire
Mario Velazquez
Bridgette Kirby
Annie Howard
Tse Han Chau

Fig. 1. Fringe profile page of Edward Forelli. Note the tagger widget on the left hand side, as well as incoming (received) and outgoing (assigned) tag clouds.

employee profiles with tagging in a corporate enviornment. In Fringe, people are allowed to tag each other with arbitrary terms. A person's incoming tags and outgoing tags are viewable in the form of tag clouds on a user's employee profile page. Figure 1 shows a Fringe profile page. The tagging functionality of Fringe is useful in keeping certain profile information such as projects, expertise, and interests, up-to-date. According to system records, most employees do not update their profile in a timely manner [5]. Fringe enables a small group of active users to update the information of the people they know through tagging. It has been shown that, in most cases, the tags one received adequately describes oneself [5]. Thus, we may make use of tags to learn more about a person, such as her projects and skills, or search for expertise in a certain field within a corporate environment.

As of May 14, 2008, 53844 people has been tagged by a total of 170137 tags in Fringe. In particular, 557 people have been tagged by 20 or more distinct terms [1]. Currently, the distinct terms from tags are listed alphabetically in a tag cloud on one's Fringe profile (see Figure 2). These terms may describe different aspects of a person. For example, the terms "fringe" and "sonar" are project names, which probably indicate the projects the person works on; the terms "ajax" and "perl" describe the programming skills the person has; the terms "blogger" and "innovator" describe the person himself.

[1] In Fringe, a user may be tagged with the same term by multiple users. For example, a software developer may be tagged the term "Java" ten times.



Tagged by: 82 people

ajax almaden blogger bluemail bluepagesplusplus collaboration dogmaster emacs f2f fetchnotes folksonomies folksonomy foocamp freebeer fringe fringe-contacts fringe-hosting-contact-management-researcher-hacker-bluepeople fringe4-team fringemaster got-served hacker independent-film innovation innovator javascript json likes-macs linux luna mbti microformats microtemplates monkey-bar odw org-tools people peopletagging perl perl-god rdf relationshipmap relmap research roc roc-user roczilla sametime selber-langsam-antworter sna social-computing social-software socialnetworkcompleteness socialnetworking someone sonar tagging taggingsummit2006 tap-innovator tommy trottel uam uam-summit-2006 unshaved user vcard voted-for-bush wayuncooool web20 www2006

Fig. 2. Tag cloud showing tags used by other people on a person. Tags with the same term are shown only once; and the larger the font size of a term, the more times the person is tagged by the term.

To learn about all the skills the person has, one has to go over all the terms in the tag cloud on the person's profile one by one and collect those terms that are relevant to skills. When the number of terms one has been tagged is large, scanning all the terms in the tag cloud is time-consuming. Other popular websites/applications that provide tagging functionality, including Flickr and Amazon.com, do no better in organizing and displaying tags to users.

Grouping tags into categories such as "Project-related" and "Skill-related" will facilitate information retrieval from tags and enhance the usability of tagging systems. Tag categorization can also help alleviate the problem of naming collisions, which is considered to be an open issue in existing tagging systems [5]. For example, some users in our corporate are working on a project codenamed "vista", which is also the name of an operating system. There are 30 tags with the term "vista" in Fringe, and it is hard to tell the actual meaning of these tags without referring to other information. If such tags were classified into categories such as "Internal Projects" and "Operating Systems", the meaning of each tag would be clarified by its category.

Unfortunately, the importance of tag-organization is largely ignored by existing tagging systems.

In this article, we propose to organize tags in a hierarchical manner. Similar to creating a tag, users may create a category or a sub-category under another category using whatever term they consider appropriate. They may then place new or existing tags into the created categories. The effect of categorization activities from different users will be combined and shared in the system. Allowing users to create categories preserves the spirit of freedom in tagging and avoids the need of a pre-defined taxonomy.

Since tagging systems such as Fringe have collected a large number of tags, it is infeasible to manually classify the existing tags into different categories. An automatic approach is needed to categorize the tags with minimum human-intervention. A challenge for automatic categorization is that, unlike words in an article, tags are individual words with very limited context information. Also, users may use different terms to refer to the same thing. For example, "tim", "itim", and "tivoli-identity-manager" all mean the same product. A categorization approach has to find out the synonyms used in tagging activities. In this article, we propose an approach that makes use of data-mining and machine-learning techniques to reliably and rapidly categorize tags based on user-inputs and feedbacks. Even though our discussion will be mainly based on Fringe, our techniques can be applied to other tagging systems as well.

The contributions of this paper are summarized as follows.

- We propose to organize tags in a hierarchical manner. Our approach allows users to create categories and sub-categories, which preserves the spirit of freedom in tagging systems.
- We design an automatic method to categorize tags using data-mining and machine-learning techniques. Our approach is efficient and effective in practice.
- We propose to solve the problem of naming collisions using user-connection graphs generated from tagging relations. We observed that tags with the same term in the same connected component of the connection graph normally mean the same thing; while tags with the same term from different connected components could have different meanings.
- We developed a prototype of our approach for a real-world tagging system, Fringe. Our prototype demonstrates the efficiency and effectiveness of our approach. Our prototype also enables one to learn about a tag whose meaning is unknown by referring to categorized related tags.

The rest of this paper is organized as follows. We describe the problem of tag categorization in detail in Section 2 and present our solutions in Section 3. After that, we describe our prototype for Fringe in Section 4. Finally, we discuss related work in Section 5 and conclude in Section 6.

2 Problem Description

In this section, we describe the problem of tag-categorization in details. We first give the representation of tags and categories, and then we define the categorization problem and discuss goodness metrics for its solutions.

Definition 1 (Tagging Relation). The *tagging relation* of a system is represented as $TR \subseteq \mathcal{U} \times \mathcal{U} \times \mathcal{W}$, where $\mathcal{U}$ is the set of all users in the system and $\mathcal{W}$ is the set of all possible terms. A *tagging tuple* (or a *tag* for short) $(u_1, u_2, t) \in TR$ indicates that user u_1 has tagged u_2 using the term t.

Intuitively, the tagging relation is the set of all the tags in the system. In Fringe, a term used in a tag may consist of a single word, such as "java" and "perl", or multiple words connected by the symbol "-", such as "java-developer" and "tivoli-identity-manager". A term consisting of a single word is called a *simple term*, while a term consisting of multiple words is called a *complex term*. Among the 170137 tags in Fringe, 22540 distinct terms have been used and 12033 (or 53%) of them are complex terms.

Definition 2 (Category). A *category* is represented as a term $c \in \mathcal{W}$. Let C be the set of categories in the system, the hierarchical relation of categories is denoted as $CH \subseteq C \times C$, where $(c_1, c_2) \in CH$ indicates that c_2 is a *subcategory* of c_1.

For example, the category "research areas" may contain subcategories such as "theory" and "database". We assume that there is a special category $unknown$. Those tags that have not been classified are placed in the category $unknown$.

Next, we define the tag-categorization problem.

Definition 3 (Tag-Categorization Problem). A state of a *tagging system* is given as $\langle TR, C, CH, FC \rangle$, where TR is a tagging relation, C is a set of categories, $CH \subseteq C \times C$ is the category hierarchy, and FC is a function that maps every tag in TR to a category in C.

The input to the *tag-categorization problem (TCP)* is an initial state $\langle TR, C, CH, FC \rangle$ such that $unknown \in C$ and $\forall_{t \in TR} FC(t) = unknown$. We are asked to output $\langle TR, C', CH', FC' \rangle$ such that $\forall_{t \in TR} FC'(t) = c$, $c \in C'$ and $c \neq unknown$.

The above definition does not require $C' = C$. In other words, new categories can be created in solutions to the categorization problem.

Definition 3 only defines what is a solution to an instance of TCP without stating what is a good solution. When the set of possible terms is infinite, we we may have infinitely many solutions for a TCP instance. It is clear that all the solutions are not equally good. For example, we may come up with a trivial solution by creating a category "xyz" and map all tags to such a category. But such a trivial solution is useless in practice. Intuitively, the categorization of tags should match the expectation of human users so as to be useful. A naive approach to get a good solution is to ask a human user to categorize all the tags in the system. But when the number of tags is large, manual categorization is infeasible.

3 A Tag-Categoration Approach

In this section, we describe our approach for tag categorization. The overview of our approach is as follows.

- First of all, we find the synonyms used in tags based on statistics and the similarity between words. When a term is placed into a category, its synonyms should be placed into that category as well.
- Second, the system learns how to categorize tags from human users in a collaborative environment. Users train the system by creating categories and placing tags into such categories. The knowledge from different users are combined, and conflicts are resolved using a voting mechanism. The system also derives rules from classified tags and applies such rules to categorize other tags.

Finding Synonyms. As we have mentioned in Section 1, since tagging systems grant users the freedom to use whatever terms they like in tags, different terms may be used to refer to the same thing. A common practice is to use initials instead of full project or title names in tags. For example, in Fringe, "de" is short for "distinguished-engineer" and both terms are used in tags. For another example, "itim" is short for "ibm-tivoli-identity-manager", which is equivalent to "tivoli-identity-manager". Finding the synonyms used in tags can speed up the categorization process, as once the category of a term is known, tags using the synonyms of the term can be automatically placed in that category as well.

A natural approach to determine if the meanings of two terms are equivalent is to see if the two often appear together. Intuitively, if users who have been tagged t_1 are usually tagged t_2 as well, and vice versa, then it is likely that t_1 and t_2 are synonyms. Let $|t_1|$ and $|t_2|$ be the number of tags using term t_1 and t_2, respectively, and $|(t_1, t_2)|$ be the number of users being tagged both t_1 and t_2. The function $g(t_1, t_2) = \langle |(t_1, t_2)|/|t_1|, |(t_1, t_2)|/|t_2| \rangle$ computes the *statistical likelihood* that t_1 and t_2 are synonyms. If both $|(t_1, t_2)|/|t_1|$ and $|(t_1, t_2)|/|t_2|$ are close to 1, we have high confidence that t_1 and t_2 mean the same thing. For instance, "bsa" is the initial of "business-solutions-architect" and the two terms are synonyms. Let t_1 = "bsa" and t_2 = "business-solutions-architect". We have $|(t_1, t_2)| = 9$ and $g(t_1, t_2) = \langle 100\%, 90\% \rangle$ in Fringe.

Even though a large number of pairs of synonyms in Fringe have high statistical likelihood, we found that using statistical likelihood alone to determine synonyms will lead to relatively high false positive rate. For example, "ns-staff" and "buddylist" appear together 206 times and $g(\text{ns-staff, buddylist}) = \langle 99\%, 94\% \rangle$, which is higher than most other pairs of terms. But apparently, "ns-staff" is not a synonym of "buddylist". Also, many pairs of synonyms have relatively low statistical likelihood. For instance, the statistical likelihood of "wempe" and "websphere-everyplace-mobile-portal-enable" is only $\langle 60\%, 60\% \rangle$, but they are probably equivalent as the former is the initial of the latter and no other terms in Fringe has initial "wempe".

To complement the statistical approach, we take the syntactically similarity between words into account when finding synonyms. For example, two terms are syntactical similar if one is the initial of the other, one is the prefix/suffix of the other, or they are complex terms sharing some single terms. In general, we compute the confidence score measuring the likelihood that two terms are synonyms by combining the statistical likelihood of the pair and the level of similarity between the words used in the terms.

As we will see in Section 4, our approach is effective in finding synonyms used in tags.

Learning in a collaborative environment. In our system, categories are created by users. The system learns how to place tags into different categories by observing how human users classify tags. For example, assume that a user is browsing an employee's profile and finds that a tag with the term "business-solutions-architect" remains unclassified. Then, the user may drag the tag to the category "Job Title", which was created earlier by another user. The system learns that other tags with the same term probably belong to the same category (except in the cases of name collisions, which will be discussed later). Also, tags with the term "bsa", which is found to be a synonym of "business-solutions-architect", should be placed in the category "Job Title" as well.

Fringe works in a collaborative environment, where a large number of users browse employee profiles through Fringe every day. Every user may contribute to the learning process of tag categorization. Conflicts may occur when we combine the opinions of all users, as different users may have different opinions on how to categorize a term. For instance, some users think that tags with term t should be placed in category c_1, while some others consider category c_2 to be more appropriate. Sometimes, users may make mistakes as well, like placing t into c_1 while t should belong to c_2. To resolve conflicts and correct errors, we employ a voting mechanism, i.e. if the number of users who prefer t to be in c_1 is larger than the number of those who prefer c_2, then t will be put in the category c_1.

Fringe has collected 170137 tags, and 22540 distinct terms are used in those tags. To speed up the categorization process, the system must be able to guess the categories of those tags whose terms have not been categorized by human users based on what it has learned. The system thus needs to link terms that are likely to belong to the same category together. Unlike words extracted from articles, which can be linked by topics, tags are individual terms and there are not obvious semantic links between them. Here, we consider linking tags syntactically through complex terms.

As we have mentioned in Section 2, more than 50% of the terms used in Fringe tags are complex terms, which consist of more than one words. An important observation is that the category of a complex term is determined by its subterms. For example, if "tivoli" is a product, then complex terms such as "tivoli-identity-manager" and "tivoli-workload-scheduler" are likely to be products too. Furthermore, by knowing that "tivoli-workload-scheduler" is a product, the system may infer that other terms containing the word "workload" and/or "scheduler" could be a product as well. In general, the system may derive categorization rules from what it has learned from human users.

Definition 4 (Categorization Rule). A *categorization rule* is represented as $\langle t \rightarrow c, T\rangle$, where t is a term, c is a category, and T is a set of terms called the *supports* of the rule. We say that $\langle t \rightarrow c, T\rangle$ is a rule for term t.

The system maintains the following two types of rules for tag classification.

– *User-specified rules*: if the combined effort of human users indicates that term t (which can be a simple or complex term) belongs to category c, then a user-specified rule $t \rightarrow c$ is added to the system and $\{t\}$ is the supports of the rule.
– *Derived rules*: if the combined effort of human users indicates that complex term t belongs to category c, then for every single subterm t' of t, a derived rule $t' \rightarrow c$ is added to the system (if such a rule does not exist yet) with $\{t\}$ as its supports. This indicates that it is possible that t' belong to category c, because one of its superterms belongs to c. If the rule $t' \rightarrow c$ is derived again from another complex term t_1 in future, then t_1 will be added to the supports of the rule. Derived rules with larger supports will be given more weights when they are applied.

  For instance, if the system is told that "tivoli-identity-manager" is a product, then derived rules "tivoli $\rightarrow$ product", "identity $\rightarrow$ product" and "manager $\rightarrow$ product" are added to the system (if they do not exist). And "tivoli-identity-manager" is added to the supports on these rules.

Note that a single term may have multiple derived rules targeting different categories. For example, in addition to the rule $\langle$manager $\rightarrow$ product, $\{$tivoli-identity-manager$\}\rangle$, we may have another rule $\langle$manager $\rightarrow$ job title, $\{$sales-manager, people-manager$\}\rangle$. Intuitively, the two rules indicate that a complex term containing "manager" could be a product or a job title, and the latter case is more likely than the former as the latter has more supports. Also, a term may be removed from the supports of a rule in the future, if its category changes later. Term t will be removed from the supports of rule $t \rightarrow c$ when its category is no longer c (for example, more users vote that t should be in c'). A rule will be removed if its supports becomes empty.

Next, we describe how to apply the categorization rules to guess the category of a new term. Given a term t, let $S_t = \{t_1, \ldots, t_m\}$ $(m \geq 1)$ be the set of single terms in t. The algorithm that outputs a category for t is described in Figure 3. The high-level idea of the algorithm is that, for every sub-term $t_i \in S_t$, we compute the probability that t_i falls into a certain category c based on the supports of its rules (if t_i does not have a rule regarding c, it is ignored). Note that user-specified rules have higher priority over derived rules. We then determine the probability that t falls into c by combining the results of its subterms. Finally, we select the category c' with the largest probability that t falls into it [2]. Furthermore, when guessing the category of t, we prefer to use rules whose terms are closer to t. For example, assume that we have rules for both "manager" and "identity-manager". To guess the category of "tivoli-identity-manager", rules for "identity-manager" will be used while rules for "manager" will not. This is because "identity-manager" is a subterm of "tivoli-identity-manager" that subsumes "manager", which indicates that the rules for "identity-manager" should be more precise with respect to "tivoli-identity-manager" than those for "manager".

The following example illustrates how the algorithm works.

Example 1. Assume that we are given a term "social-network-computing", which has not been categorized by human users. We have the following rules in the system.

- $\langle$social $\rightarrow c_1, S_1\rangle$, where $|S_1| = 30$.
- $\langle$social-network $\rightarrow c_1, S_2\rangle$, where $|S_2| = 9$.
  $\langle$social-network $\rightarrow c_2, S_3\rangle$, where $|S_3| = 1$.
- $\langle$computing $\rightarrow c_1, S_4\rangle$, where $|S_4| = 2$.
  $\langle$computing $\rightarrow c_2, S_5\rangle$, where $|S_5| = 2$.
  $\langle$computing $\rightarrow c_3, S_6\rangle$, where $|S_6| = 6$.

According to our algorithm, the rule for "social" will not be used as we have rules for "social-network" which subsumes "social". We then compute the probabilities that "social-network-computing" does not fall into c_1, c_2, and c_3, using the rules for "social-network" and "computing". We have

- $M(c_1) = (1 - 9/(1+9)) \times (1 - 2/(2+2+6)) = 0.1 \times 0.8 = 0.08$
- $M(c_2) = (1 - 1/(1+9)) \times (1 - 2/(2+2+6)) = 0.9 \times 0.8 = 0.72$
- $M(c_3) = 1 \times (1 - 6/(2+2+6)) = 1 \times 0.4 = 0.4$

[2] In the algorithm described in Figure 3, we actually select the category with the smallest probability that t does not fall into it.

```
Input: A set $R_u$ of user-specified rules; a set $R_d$ of derived rules; a term $t$.
Output: A category $c_t$.

Begin
    Let $T = \{t_1, \ldots, t_m\}$ be the set of single subterms of $t$;
    Let $M$ be a hashmap that maps a category to a real number;
      (Initially, $M$ maps every category to 1)
    For every term $t_i \in T$ do
        Find term $t'$ with the largest number of subterms that satisfies the follows:
          - There is at least one rule for $t'$ in $R_u$ or $R_d$
          - $t_i$ is a subterm of $t'$
          - Every single subterm of $t'$ is in $T$
        If such a $t'$ does not exist, continue to process the next subterm in $T$;
    EndFor;
    Let $R$ be the set of rules for $t'$ in $R_u$;
    If $R$ is empty, let $R$ be the set of rules for $t'$ in $R_d$;
    Let $x$ be the sum of the size of supports of rules in $R$;
    For every rule $r_j = \{t' \rightarrow c_k, S\} \in R$ do
        $M(c_k) = M(c_k) \times (1 - |S|/x)$;
    EndFor;
    Return the category $c_t$ such that $M(c_t)$ is the smallest;
End
```

Fig. 3. The algorithm for guessing the category of a term t based on a set of categorization rules

Since $M(c_1)$ is the smallest, the algorithm returns c_1 as the most likely category for "social-network-computing".

As shown in the experimental results in Section 4.1, our learning approach allows the system to categorize a large number of tags with a small number of rules. And the correctness rate of the categorization is satisfactory.

Handling Naming Collisions. As stated in Section 1, naming collisions is an open issue in tagging systems. With naming collisions, it is possible that tags with the same term should belong to different categories, depending on their context. In our approach, we detect and handle naming collisions using tagging-relation graphs.

Our observation is that, when naming collisions occur, it is almost always the case that two disjoint sets of people use the same term to mean different things. It is very rare that a person would use the same term for different meanings as that would confuse herself. Also, it is unlikely that a term has different meanings to a group of closely connected users in the tagging system. We say that a group of users are closely connected if they often tag each other. In particular, if a group of users tag each other using term t, then t probably means the same thing in the group. A tagging-relation graph with respect to t can be constructed in such a way: the nodes of the graph represent users, and there is an edge from node n_1 to n_2 if and only if user u_1 has tagged u_2 using term t.

A potential naming collision on term t is detected when the system finds that several users vote category c_1 for some tags with term t, while several others vote category c_2

for some other tags with the same term. When this happens, we construct a tagging-relation graph with respect to t. If those tags being voted as c_1 and those as c_2 are associated with users in different connected components of the graph, we may conclude that t means something belonging to c_1 in some connected components, while it means another thing belonging to c_2 in other connected components. In this case, the system uses different internal representations of t to distinguish it from one connected component to another. The difference of internal representations of term t in different tags is invisible to end-users. In that case, instead of categorizing all tags with t into either c_1 or c_2, we may now place some of the tags in c_1 and the others in c_2. And placing a tag under a category makes it less likely for users to confuse its meaning with a tag with the same term in another category.

4 A Prototype for Fringe

We have implemented a prototype for our tag-categorization approach in Fringe. In this section, we first show that how categorization may facilitate the retrieval of information from tags by presenting screenshoots of our tool, and then we present experimental results to demonstrate the effectiveness of our method.

As we can see in Figure 4, tags are grouped into different categories for display in our tool, which makes it much easier to learning about different aspects of a person, when compared to the current tag cloud showed in Figure 2. We may find out the skills the user has and the projects he is working on at a glance, instead of having to go over all the tags to pick out relevant information.

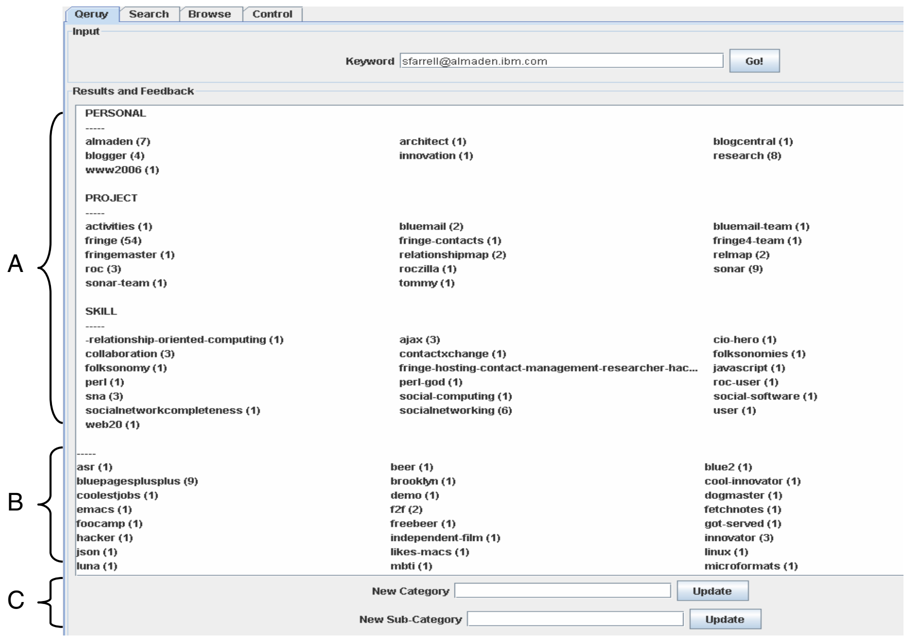
Tag categorization also makes it easier to learn about those tags that are unknown to a user who is browsing a profile. By double-clicking a tag on an employee's profile, our tool pops up a window containing the tags that may be related to the targeted tag. Such related tags are selected from those tags, which often appear together with the targeted tag on employ profiles. For instance, Figure 5 shows the tags related to "fringe". We can quickly find out that "fringe" is probably developed in Almaden Research Center (as it is related to the term "almaden" in the category "Location"), it is about collaboration and social-network (as it is related to such terms in the category "Skills"), and the personnel related to Fringe also work on projects such as Bluemail and Dogear.

In general, in a people-tagging system such as Fringe, a person's tags describe the person, while a tag's related tags describe the tag itself. Tag categorization makes it easier to learn about both people and tags in the system.

4.1 Experimental Results

In this subsection, we demonstrate the effectiveness of our tag-categorization approach by presenting experimental results. Our implementation is on Fringe and we use the dataset collected by Fringe for experiments. Fringe is a tagging system used in our corporate and its dataset represents a real-world scenario.

First of all, we would like to evaluate our approach on finding synonyms used in tags. As stated in Section 3, the approach we designed takes into account statistical likelihood and word similarities.



Input

Keyword sfarrell@almaden.ibm.com

Results and Feedback

PERSONAL
-----

almaden (7)	architect (1)	blogcentral (1)
blogger (4)	innovation (1)	research (8)
www2006 (1)		

PROJECT
-----

activities (1)	bluemail (2)	bluemail-team (1)
fringe (54)	fringe-contacts (1)	fringe4-team (1)
fringemaster (1)	relationshipmap (2)	relmap (2)
roc (3)	roczilla (1)	sonar (9)
sonar-team (1)	tommy (1)	

SKILL

-relationship-oriented-computing (1)	ajax (3)	cio-hero (1)
collaboration (3)	contactxchange (1)	folksonomies (1)
folksonomy (1)	fringe-hosting-contact-management-researcher-hac...	javascript (1)
perl (1)	perl-god (1)	roc-user (1)
sna (3)	social-computing (1)	social-software (1)
socialnetworkcompleteness (1)	socialnetworking (6)	user (1)
web20 (1)		

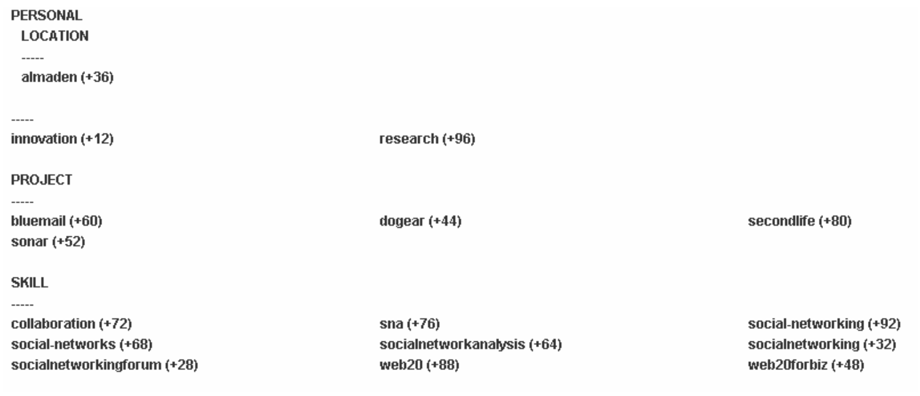
-----

asr (1)	beer (1)	blue2 (1)
bluepagesplusplus (9)	brooklyn (1)	cool-innovator (1)
coolestjobs (1)	demo (1)	dogmaster (1)
emacs (1)	f2f (2)	fetchnotes (1)
foocamp (1)	freebeer (1)	got-served (1)
hacker (1)	independent-film (1)	innovator (3)
json (1)	likes-macs (1)	linux (1)
luna (1)	mbti (1)	microformats (1)

New Category

New Sub-Category

Fig. 4. The tags on an employee's profile as displayed in our tag categorization tool. (A) Tags that have been classified into three categories, "Personal", "Skills", and "Projects". (B) Tags that have not been categorized yet. (C) Feedback area where users can create a category, categorize an unclassified tag, or try to correct the category of a tag.



PERSONAL
  LOCATION
  -----
  almaden (+36)

-----

innovation (+12)	research (+96)	

PROJECT
-----

bluemail (+60)	dogear (+44)	secondlife (+80)
sonar (+52)		

SKILL
-----

collaboration (+72)	sna (+76)	social-networking (+92)
social-networks (+68)	socialnetworkanalysis (+64)	socialnetworking (+32)
socialnetworkingforum (+28)	web20 (+88)	web20forbiz (+48)

Fig. 5. Tags related to "fringe" as displayed by our tag categorization tool

In Fringe, there are 22540 distinct terms used in the 170137 tags. We set the selection threshold of our algorithm to a such value that it outputs 312 pairs of potential synonyms. To evaluate the correctness of the results, we ranked the pairs based on their confidence scores and manually examined the first 50 and the last 50 pairs. Out of the first 50 pairs, 37 pairs are known to be synonyms, 8 pairs are closely related terms (such as “soma-workshop” and “soma-workshop-dk2007”), 1 pair appears to be unrelated, and there are 4 pairs that we are not sure about their relation as we do not know the meaning of the terms. In other words, the correctness percentage of the algorithm is 74%, and if we take closely related terms into account, the correctness percentage is 90%. In contrast, in the last 50 pairs, 6 pairs are synonyms, 21 of them are related, 7 are unrelated, and we are not sure about the remaining 16. If we take pairs of related terms into account, the correctness rate of the algorithm is 54%. Since two closely related terms are likely to be placed in the same category, it does not do much harm on the correctness of categorization, even if the algorithm takes closely related terms as synonyms and automatically classifies them together. Note that we can always increase the threshold to make the results more accurate (in that case, less pairs will be outputted). Also, our tool provides an interface that allows users to correct the mistakes made by the system in synonym discovery.

Next, we present experimental results of our categorization algorithm. Our prototype has not been deployed in Fringe yet, so we are unable to evaluate our approach in a real collaborative environment. The major objective of the experiments reported here is to evaluate the effectiveness of the rule-based categorization approach proposed in Section 3.

In our experiment, we created three categories; they are “Personal” (which indicates personal information), “Skills” (which includes programming and research skills), and “Projects” (which are name of projects). To train the tool, we manually categorize 36 tags with distinct terms. Among the 36 tags we classified, 8 of them are placed into “Personal”, 16 of them are “Skills”, and 12 belong to “Projects”. We then ask our tool to classify all the tags in the system. It took about 15 seconds for our tool to finish the classification. Among the 170137 tags, 141557 (or 83%) of them have been categorized by the tool. And those tags that have been classified have 21664 distinct terms (i.e. 96% of the total 22540 distinct terms). Of all the classified terms, 4371 (or 20%) of them are categorized as “Personal”, 9145 (or 42%) of them are “Skills”, and 8148 (or 38%) of them are “Projects”. To evaluate the correctness rate of the approach, we examined 10 employee profiles, 5 of which were selected manually as they contain a large number of tags (more than 60), and the other 5 were picked randomly. Among the 1107 tags with 476 distinct terms in the selected profiles, 467 distinct terms (or 98%) are classified. Through manual examination, we estimate that 300 (or 64%) distinct terms have been classified correctly. Most of the incorrectly-categorized terms have been mistakenly placed into the category “Skills”. The major reason is that several terms we manually classified as “Skill”, such as “machine-learning” and “social-network-analysis”, are complex terms, while all the terms we manually classified as “Personal” and “Projects” are single terms. Since the categorization of complex terms would lead to more derived rules, the tool tends to place terms into the category “Skills”.

In general, the performance our rule-based category algorithm is satisfactory. We have only classified 36 tags in the training process, but our rule-based algorithm is able to spread such limited categorization knowledge to most other tags in the system with a decent correctness rate. After our tool is deployed in our corporate websites, it will be able to learn and derive a lot more categorization rules from many users. We expect the accuracy of tag categorization will improve significantly with the growing number of rules.

5 Related Work

Tagging in collaborative environments has attracted significant amount of interests in the research community [8,2,7,11,4,5,10].

Research has been done on tags visualization [8,2] so as to facilitate the browsing of resources with tags, or allow users to figure out the important tags associated with an item quickly. However, existing tag visualization work does not provide automatic tag-classification functionality. In order to select tags describing a certain aspect of an item, users still have to go over all the tags and do the selection themselves.

In tagging systems, users have the freedom to choose whatever terms they like in tags. As we have seen earlier, this results in non-uniform usage of terminology (e.g. synonyms), naming collisions, and other problems. There has been work on improving the quality of tags in collaborative environments. In [11], Xu et al. defined a set of general criteria for a good tagging system. They also proposed a collaborative tag suggestion algorithm using those criteria to spot high-quality tags. Similar to [11], the goal of our paper is to improve the usability of tagging systems. However, instead of suggesting tags based on existing ones, we focus on automatic organization of tags so as to make it easier for users to find the information they want. In [10], Razavi and Iverson proposed to allow users to control access to their information using people-tagging. But they did not study the categorization of tags.

Our tag categorization work is related to research on documentation clustering [1]. In documentation clustering, topics are extracted from text-documents, and those documents are classified based on their topics. The major difference between our tag-categorization work and documentation clustering is that text-documents contain a large number of words while tags are individual terms. When given a text-document, people may extract frequently-appeared words in the document to serve as its topic (or category). However, tags do not have an article as its context, and thus there is little contextual information we can use to link together different terms used in tags.

6 Conclusion and Future Work

Tagging has gained popularity in collaborative systems, such as online photo sharing, webpage bookmarking, and employee profiling systems. Since different tags may describe different aspects of an item, placing tags into different categories will improve the usability of tagging systems, especially when the number of tags in the system is large. In this paper, we have proposed an approach to automatically categorize tags in tagging systems. Our approach look for synonyms used in tags by combining statistical data and the syntactical similarity between words. Whenever human users classify a tag, categorization rules may be derived to classify more tags. Also, we have proposed to use tagging-relation graph to handle the issue of naming collisions. Finally, we have implemented a prototype of our approach for Fringe, which is a real-world tagging system. Our experiments on the data collected by Fringe demonstrated the effectiveness of our approach.

Our next step is to develop a user-friendly interface for our tool and deploy it on Fringe. We will perform user-study with the deployed tool and make improvements. Also, we will explore other ways, such as searching and recommendation, to make use of the tags collected by Fringe. We would also like to test our tag-categorization approach on other enterprise-oriented tagging systems for webpages and resources, such as Dogear [9].

References

1. Andrews, N., Fox, E.: Recent developments in document clustering. Technical Report TR-07-35 (2007)
2. Dubinko, M., Kumar, R., Magnani, J., Novak, J., Raghavan, P., Tomkins, A.: Visualizing tags over time. In: International World Wide Web Conference (WWW), Edinburgh, Scotland (2006)
3. F-Spot, `http://www.gnome.org/projects/f-spot/`
4. Farrell, S., Lau, T.: Fringe contacts: People-tagging for the enterprise. In: Workshop on Collaborative Web Tagging in conjuction with WWW 2006, Edinburgh, Scotland (2006)
5. Farrell, S., Lau, T., Nusser, S., Wilcox, E., Muller, M.: Socially augmenting employee profiles with people-tagging. In: UIST 2007: Proceedings of the 20th annual ACM symposium on User interface software and technology, pp. 91–100. ACM, New York (2007)
6. GMail, `http://www.gmail.com`
7. John, A., Seligmann, D.: Collaborative tagging and expertise in the enterprise. In: Workshop on Collaborative Web Tagging in conjuction with WWW 2006, Edinburgh, Scotland (2006)
8. Kaser, O., Lemire, D.: Tagcloud drawing: Algorithms for cloud visualization. In: International World Wide Web Conference (WWW), Banff, Canada (2007)
9. Millen, D.R., Feinberg, J., Kerr, B.: Dogear: Social bookmarking in the enterprise. In: CHI 2006: Proceedings of the SIGCHI conference on Human Factors in computing systems, pp. 111–120. ACM, New York (2006)
10. Razavi, M.N., Iverson, L.: Supporting selective information sharing with people-tagging. In: CHI 2008: CHI 2008 extended abstracts on Human factors in computing systems, pp. 3423–3428. ACM, New York (2008)
11. Xu, Z., Fu, Y., Mao, J., Su, D.: Towards the semantic web: Collaborative tag suggestions. In: Workshop on Collaborative Web Tagging in conjuction with WWW 2006, Edinburgh, Scotland (2006)