Automated Tag Clustering: Improving search and exploration in the tag space

Grigory Begelman
Technion - Israel Institute of Technology
Computer Science Dpt
gbeg@cs.technion.ac.il

Philipp Keller
Citrin Informatik GmbH
phred@citrin.ch

Frank Smadja
RawSugar
frank@rawsugar.com

1. INTRODUCTION

In this paper we discuss the use of clustering techniques to enhance the user experience and thus the success of collaborative tagging services. We show that clustering techniques can improve the user experience of current tagging services. We first describe current limitations of tagging services, second, we give an overview of existing approaches. We then describe the algorithms we used for tag clustering and give experimental results. Finally, we explore the use of several techniques to identify semantically related tags.

2. MOTIVATION

The success of tagging services like Flickr[1], del.icio.us[2] and technorati[3] has shown that tagging is a great collaboration tool. Tagging seems to be the natural way for people to classify objects as well as an attractive way to discover new material. Tagging services provides users with a repository of tagged resources (a.k.a tagspace) that can be searched and explored in different ways. More and more people use at least one tagging service and enjoy them as discovery tools. Indeed, tagging is simple, it does not require a lot of thinking and it is very useful to find the tagged objects later. People tag pictures, videos, and other resources with a couple of keywords to easily retrieve them in a later stage. However, looking for information in the tag space has a number of hard limitations.

The difficulty comes from the fact that several people usually use different tags for the same document. In fact, even a single user's tagging practice may vary over time. Usually, this variability is compensated by looking at many users' tags; which is only possible when the page has been tagged many times. However, for less popular pages the problem remains. Currently tagging services still provide a relatively marginal value for information discovery and we claim that with the use of clustering techniques this can be greatly improved. We first discuss the main limitations of the current tagging services.

2.1 Limited Search

Let us imagine that you would like to tag the picture in Figure 1. It is a picture of a piece of sushi called *nigiri* (hand formed) sushi as opposed to other types of sushi like *maki*, *futomaki* or *temaki sushi*. A person not aware of this classification of sushi would tag this picture with any combination of the following tags: *food, fish, raw fish, rice, Japanese*. However a more expert person would use: *nigiri sushi*, or *toro*. Whithout delving on the psychological aspects of tagging[4] (nor on the nuances of sushi); we cleary see that people think and tag differently. This creates a noisy tagspace and thus makes it harder to find material tagged by other people.

Figure 1: How would you tag this?

In summary, if you tag the above picture as *toro*, people searching for sushi or food will not find it. This type of problem is rooted in the language, words are often related and do not stand in isolation. Such relations among words are called *lexical relations*. We refer the reader to WordNet[5] for a thorough treatment of semantic and syntagmatic relations among words.

Our point is that, without accounting for lexical relations, searching in a tag space in which many people of various background collaborate is bound to be very limited.

2.2 Limited Subscription

The availability of RSS and ATOM feeds has recently created a new information discovery paradigm which we call here the subscription paradigm. An increasing number of Internet users discover information with the use of these tools.

The motivation of the subscribing user is to stay informed on a certain topic. It is important to receive all documents related to the topic but it is less important if some received documents are less relevant. In other words, the subscribing user is expecting a high recall and will accept a lower precision.

[1]http://www.flickr.com, now part of Yahoo!
[2]http://del.icio.us, now part of Yahoo!
[3]http://www.technorati.com

WWW2006, May 22–26, 2006, Edinburgh, UK.

[4]See http://www.rashmisinha.com/archives/05_09/tagging-cognitive.html by Rashmi Sinha for a good discussion on the subject
[5]http://wordnet.princeton.edu/

With tagging services you can subscribe to a number of tags, and all items tagged with these tags will show up in your subscription. However, in practice if you subscribe on to the tags *java* and *article*[6], you will miss articles that are tagged with the words *blog* or *essay* instead of *article*. Recognizing lexical relations is crucial to be able to provide an effective subscription service.

2.3 Limited Exploration

The whole promise of collaborative tagging is that by exploring the tag space you can discover a lot of useful information you would not find with traditional search engines. When your information need is not well defined, the idea that you can explore and see what other people tagged with certain tags is very attractive. We believe that tagging will be able to reach a very wide audience only when exploration techniques will be effective.

Currently, there are two main ways the tag space can be explored: using search/refine, and using some kind of tag space visualization such as a tag cloud.

Say you are looking for a restaurant in your area, searching on del.icio.us for *restaurants*, your results look like the image in Figure 2.

Figure 2: Searching for a restaurant with del.icio.us

You definitely get a few useful links at the top, *Yelp* is a very useful site to find restaurants. However, if you want to explore and see what types of restaurants are available in what locations or in what kind of price range, the tag list on the right is not useful. At this point, it is better to continue the exploration at *yelp* for example. This is very similar to what you get by searching for restaurants with Google. The first links are very relevant and are probably directories or hubs that contain information on restaurants. The true exploration will be pursued at one (or several) of these hubs. We believe that searching is only the first step in exploring, and the user wants to continue exploring in a way they would do in a directory like *Open Directory* or with shopping sites. This is only possible if tags are grouped in clusters.

The other way to explore the tag space is to look at popular pages or tags, for example, a tag cloud in which the size of the tags is proportional to their popularity[7]. Although a great visualization paradigm, we believe that with today's tagclouds it is hard to find more than one or two tags to click on. Tags are not grouped, there is too much information, so that you find lot of related tags scattered on the tag cloud. One or two popular topics and all their related tags tend to dominate the whole cloud. For example, looking at the del.icio.us tagcloud, one would mostly see tags related to web design and technologies. This is because these topics are overwhelmingly more frequent than anything else. There are some 4,500 links tagged with *chocolate* and some 61,000 links tagged with *food*. However, these hardly show up on the tag clouds or the popular pages. We claim that accounting for tag clusters by, for example, showing five semantically more cohesive tag clouds is much more informative.

The key in building an effective exploration space seems to be able to group and show related items and to explain how the items are related. In hierarchical classification systems like dmoz[8] it is easy to present related items, namely the parent, siblings and children items. However, in tagging spaces, such relations don't exist. Some tagging systems present lists like "this tag often occurs together with the following tags" (related tag list in del.icio.us) or "this item is tagged x, here are other items tagged x". This information is too raw to build an exploration space upon.

We claim that if we could automatically and dynamically cluster tags whithout putting more burden on the user, we could provide a much stronger service. Searching, subscribing and exploring would be much more effective.

3. RELATED WORK

There is a lot of relevant work to discuss and we will briefly mention some here. First, we should mention the taxonomy projects such as Open Directory (dmoz.org) and Yahoo! Directories who in fact recognized the issue of tagging even before it existed. Their solution was flawed, however, because they put the burden on the tagger which in their case was either some Yahoo! Employee or a volunteering librarian.[9] Along these lines are the shopping sites (shopping.com[10], Yahoo! Shopping[11], Froogle[12]) who use somehow semi-automated techniques for tagging and are based on controlled vocabulary.

With RawSugar[13] taggers can specify tag hierarchies in their own accounts (saying that sushi is a subtag of food for example). The system uses these hierarchies to provide a strong exploration and search experience. Figure 3 is the RawSugar tag box extracted from a search page for *restaurants* on a user's account. The tags are grouped according

[6]http://del.icio.us/rss/tag/java+article, or http://rawsugar.com/rss/search/java/articles

[7]As in http://del.icio.us/tag/ or http://www.flickr.com/photos/tags/ for example.

[8]http://dmoz.org/

[9]See http://wiki.osafoundation.org/bin/view/Journal/... HierarchyVersusFacetsVersusTags for a disccusion on the topic

[10]http://shopping.com

[11]http://shopping.yahoo.com/

[12]http://froogle.google.com/

[13]http://rawsugar.com

to the user defined hierarchy and thus present a more powerful exploration space. This solves the problem in specific user's directories but not on the global tagspace because the clusters or tag groups are still too sparse.

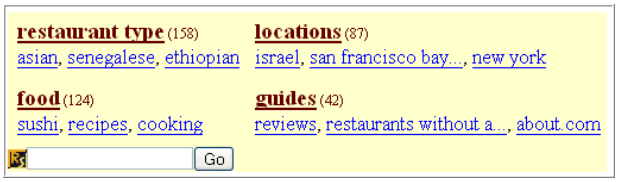

Figure 3: Searching for restaurants on RawSugar

Flickr has *Flickr clusters*, which, provided a popular tag, give related tags grouped into clusters. For example, looking at the clusters for the word *Jaguar*[14], we see that the clusters neatly fall into several semantic categories of Jaguars: animal, car and plane. The hereby presented guidepost is what makes the difference. Clustering makes it possible to present a guidepost, to provide the means that allow the user to explore the information space. In addition, Flickr also has an *interestingness* exploration technique which they define as a factor of several parameters including the pageviews, the comments left by users, the specific users, etc.

Rashmi Sinha[15] has published a number of entries on tagging and clustering. Bielenberg and Zacher [1] mention tag clustering[16].

One should also mention a growing number of tag visualization techniques in various stages of development that are currently available on the Web[17].

4. CLUSTERING ALGORITHMS

4.1 Introduction

Data clustering is a common technique for statistical data analysis. Clustering provides partitioning of a dataset into subsets of similar objects or data clusters.

Before actually using a clustering technique the first task one has to do is to transform the problem at hand into a numeric representation that can be used by clustering algorithms. In our case, the goal is first to provide a similiarity measure among tags and then to run clustering techniques on the tag space represented like this. Below we first discuss our proposed technique to find similar tags and then we discuss the use of clustering techniques.

4.2 Finding Strongly Related Tags

In this section, we present an algorithm to find strongly related tags. The algorithm is based on counting the number of co-occurrences (tags that are used for the same page) of any pair of tags and a cut-off point is determined to decide when the co-occurrence count is significant enough to be used. This results in a sparse matrix that represents tags, so that the value of each element is the similarity of the two tags.

Say a user tags an article about *African trees* that is written by an XHTML expert with the following tags: *xhtml, standard, trees, biology, africa, toread, resource*. Then *(xhtml, standard)* and *(xhtml, trees)* would each get one count as co-tags. After processing the whole tagspace, we use the frequency counts of all the co-tag pairs and attempt to identify the significant co-tags. In order to do that, we determine the pairs of tags that co-occur significantly more frequently then expected. We look for a cutoff point above which the co-tags are considered strongly related. Frequency graphs we examined usually exhibit a general "relatedness distribution" as shown in Figure 5. Let's look for example at the tags related to tag *rss* below:

tag	*count*	*tag*	*count*
feed	310	web2.0	77
blog	298	home	65
feeds	246	wikipedia	59
search	219	blogs	57
news	173	biography	53
google	103	preview	48
xml	102	learn	33
web	81	sitemap	30

Table 1: Co-tags of "RSS" and their counts

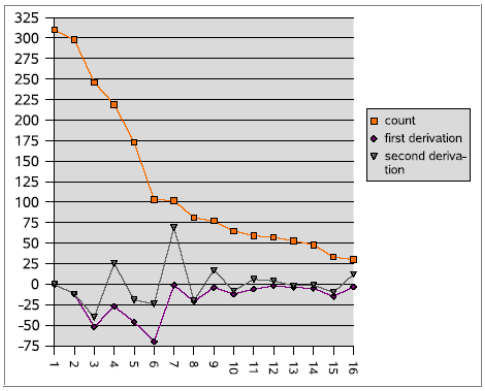

Figure 4: Tags related to "RSS"

In the table 1 and the graph 4, we see that "feed" occurred 310 times together with rss. In the graph, the y-axis is the $count_i$ (how many times a certain tag is used together with *rss*), as well as its 1^{st} and 2^{nd} derivative, the x-axis is i.

In the mentioned example about African trees, the tag combination *xhtml, africa* is just accidental and related to this example and thus would not be selected. We see a clear change in the shape of the plot for the $count_i$, and to determine this cutoff point, we consider the 1^{st} and 2^{nd} derivative of the count, we start from the tail on the right end and seek the point where the 1^{st} derivative has its first high peak (that is when the second derivative goes from positive to negative) and check if the peak was high enough.

[14]http://www.flickr.com/tags/jaguar/clusters
[15]http://www.rashmisinha.com/, http://www.rashmisinha.com/archives/05_02/tag-sorting.html
[16]See also http://group.us and http://laurie.informatik.uni-bremen.de/clusty/
[17]http://www.newzingo.com, http://hublog.hubmed.org/archives/001049.html, http://www.corante.com/many/archives/2005/01/26/visualizing_the_collective_brain.php, http://www.quasimondo.com/tagnautica.php, http://www.ivy.fr/revealicious/, http://www.tagcloud.com/

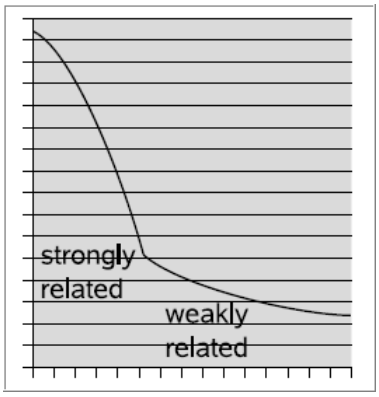

Figure 5: Typical distribution of related tags

If these two conditions are fulfilled then this is the cutoff, the tags on the left hand side of the cutoff relate strongly to tag *rss*. The only parameter that has to be optimized is the "minimal peak hight". However, sometimes the distribution doesn't have this disruption point or we simply don't have enough data to compute this point, therefore the tag_j with the most co-occurences of tag_i is always considered strongly related to tag_i.

If we do this for every tag in the tagspace we obtained an undirected graph $G(V, E, W)$ consisting of nodes V, a set of edges E and a weight matrix W. Each vertex v_i of the graph corresponds to a tag tag_i. There is an edge between v_i and v_j if the tag tag_i relates strongly to tag tag_j or vice versa according to the described algorithm. The weight $w_{i_1 i_2}$ corresponds to the number of times tag_{i_1} occurred together with tag_{i_2} within the same item.

We tested this algorithm on data from del.icio.us gathered from their RSS-feed[18]. To simplify computations we pruned all relations with a count smaller than 30; at the date of the experiments the table contained 1100 tag connections and we found 23 independent clusters, whereas a cluster is a set of tags that are connected.

Figure 6 is a part of one of the big clusters for *design*.

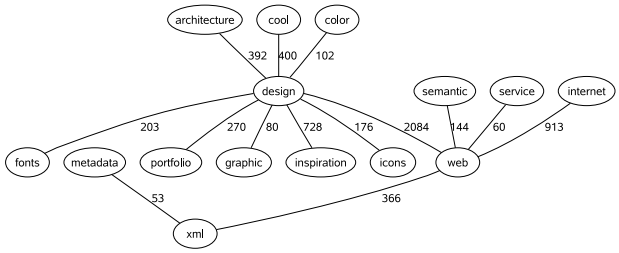

Figure 6: part of cluster "design"

The clusters were computed regularly and the results seem to be fairly stable, that is there was no tendency towards one big cluster. Some clusters seem too big, i.e. the cluster above should be split into a "design" and a "web" cluster. The spectral clustering algorithm described by Scott White [5] helped splitting up into handier clusters.

4.3 Clustering Algorithm

[18]http://del.icio.us/rss/

The input for the tags clustering algorithm consists of:

- Tags t_i, $i = 1 \dots I$
- Users u_j, $j = 1 \dots J$
- Tagged resources (web resources were used in our experiments) r_k, $k = 1 \dots K$
- A 3D tensor $A \in \mathbf{R}^{I \times J \times K}$ of boolean values. The tensor A contains tagging information: if a user u_i tagged a resource r_k with a tag t_j then $A_{ijk} = 1$, otherwise $A_{ijk} = 0$. Normally the tensor A is sparse.

Our goal is to partition the set of tags into non-intersecting groups of semantically-related tags. We show here how a graph is built from the input of the tag clustering algorithm.

Let $G(V, E, W)$ be an undirected weighted graph consisting of nodes V, the set of edges E, and a symmetric weight matrix $W \in \mathbf{R}^{I \times I}$, where I is the number of vertices. Each vertex v_i of the graph G corresponds to a tag t_i. First, we compute matrix $B \in \mathbf{R}^{I \times K}$, collecting the tagging information from all users: $B = \bigvee_j A_{ijk}$, where $\bigvee_j(\cdot)$ denotes the "logical OR" performed on the second dimension of the tensor A. The rows of the matrix B correspond to the tags, while the columns of the matrix B correspond to the tagged resources. Thus, if a resource r_k is tagged by a tag t_i by some user, then $B_{ik} = 1$.

The weight $w_{i_1 i_2}$ of the edge between the vertex v_{i_1}, corresponding to the tag t_{i_1} and the vertex v_{i_2}, corresponding to the tag t_{i_2} is the number of resources, tagged by both tags t_{i_1} and t_{i_2}. Thus we take the rows i_1 and i_2 of B, and calculate the number of resources shared by the tags t_{i_1} and t_{i_2}: $w_{i_1 i_2} = \left\| (B)_{i_1} \bigwedge (B)_{i_2} \right\|_1$, where $\bigwedge$ denotes the "logical AND" and $\| \cdot \|_1$ stands for L_1 norm of a boolean vector, or the number on "ones" in the vector.

There are many algorithms for graph clustering [2]. Recently, [3] introduced the "modularity function" Q, which measures the quality of a particular clustering of nodes in a graph. Consider a particular division of a graph into k groups. The modularity function is defined as:

$$Q(P_k) = \sum_{c=1}^{k} \left[\frac{A(V_c, V_c)}{A(V, V)} - \left(\frac{A(V_c, V)}{A(V, V)} \right)^2 \right] \qquad (1)$$

where P_k defines a partitioning of the vertices into k groups, $A(V', V'') = \sum_{i \in V', j \in V''} w(i, j)$, and V_c is the set of vertices belonging to the partition c (see [5] for the discussion of the modularity function properties). Our algorithm for graph clustering uses the modularity function as a measure of the quality of partitioning.

The graph clustering algorithm is based on the spectral bisection [4]. First, we build the Laplacian matrix L_G of the graph G. The Laplacian matrix is an $I \times I$ symmetrical matrix, defined by:

- $L_G(i, i)$ equals to the degree of vertex v_i (the number of graph edges touching the vertex v_i)
- $L_G(i, j) = -1$ if there is an edge between the vertices v_i and v_j
- $L_G(i, j) = 0$ otherwise

Second, we compute the eigenvector v_2 **of** L_G corresponding to the second largest positive eigenvalue, $\lambda_2(L_G)$. The

vertices of the graph are bisected based on the sign of the corresponding component of v_2.

We combine the spectral bisection algorithm and the modularity function to the recursive greedy algorithm. Our greedy algorithm takes as input an simple connected undirected graph and performs the following steps:

1. Use spectral bisection to split the graph into two clusters.
2. Compare the value of the modularity function Q_0 of the original unpartitioned graph to the value of the modularity function Q_1 of the partitioned graph. If $Q_1 > Q_0$ accept the partitioning, otherwise reject the partitioning.
3. Proceed recursively on each accepted partition.

4.4 Experimental Results

The experiments were performed on the RawSugar database as of January 2006. The data at this point was about 200,000 pages and 30,000 tags. The results of the clustering can be accessed in the RawSugar lab page[19]. The number of clusters was chosen manually. Below are some example clusters for a few query tags.

- Query tag: *health*:
  - shopping, research
  - nutrition, food, diet
  - fitness, workout, running
  - article, science
  - life, lifehack, product,howto, gtd, reference, tip
  - esport, sport
- Query tag: *sports*
  - hockey, nhl
  - basebal, mlb, triple
  - basketbal, nba, nbdl, wnba
  - footbal, nfl
  - alcohol, beer, tv, food, bar
  - computer game, action game, free game

4.5 Using Clusters to Find Semantically Related Tags

Related tags can also help the user by suggesting interesting tags while tagging, searching, exploring or subscribing. For example a user subscribing to the tag *music* would be suggested to try the also add the tag *mp3* to his subscription. We present here a technique to automatically discover related tags based on the clusters we obtained previously. Table 2 shows a few examples we obtained with this technique on the RawSugar tag space. The algorithm works as follows:

1. For each tag t_i that is frequent enough in the tagspace:
   - Build a graph of its cotags.
   - Partition the graph to different number of clusters using the clustering algorithm described before.
   - Increase the similarity count for each pair of tags t_j t_k belonging in the same cluster.
2. Sort all the pairs of tags t_j t_k thus produced by their decreasing count.
3. Select the top N similar tags.

[19]http://www.rawsugar.com/lab

Tag	related tags
Apple	mac, osx, macosx, tiger
Art	cool, design, fun, graphics, images
javascript	ajax, dhtml, programming languages
music	audio, media, mp3, ipod, itunes
photography	galleries, photo, hi-res, sexy, flickr, images
software	computers, hardware, acorn, internet, linux, open source software, mambo, programming, technology, web howto, tips, reference, tutorials, tools
free	download, freeware, opensource

Table 2: Some related tags

5. CONCLUSION AND FUTURE WORK

We have presented in this short paper what we believe is convincing evidence that clustering techniques can and should be used in combination with tagging. Clustering can improve the tagging experience and the use of the tagspace in general. We have presented several clustering techniques and provided some results we obtained on the del.icio.us or RawSugar tagspace. We are currently investigating several other techniques including similarity measurements using mutual information and other statistical measures such as Chi-square or the Dice coefficient. We are also looking at the problems of tag spamming and inherently ambiguous tags.

6. REFERENCES

[1] K. Bielenberg and M. Zachera. Groups in social software: Utilizing tagging to integrate individual contexts for social navigation. 2005.

[2] U. Brandes, M. Gaertler, and D. Wagner. Experiments on graph clustering. In *Proceedings of the 11th Annual European Symposium on Algorithms (ESA'03)*, volume 2832 of *Lecture Notes in Computer Science*, pages 568–579. Springer-Verlag, 2003.

[3] M. E. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2), 2004.

[4] A. Pothen, H. D. Simon, and K.-P. Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM J. Matrix Anal. Appl.*, 11(3):430–452, 1990.

[5] S. White and P. Smyth. A spectral clustering approach to finding communities in graphs. In *SIAM International Conference on Data Mining*, 2005.