TagAssist: Automatic Tag Suggestion for Blog Posts

Sanjay C. Sood
Northwestern University
2133 Sheridan Road
Evanston, IL 60208
(847) 467-4786
sood@cs.northwestern.edu

Sara H. Owsley
Northwestern University
2133 Sheridan Road
Evanston, IL 60208
(847) 467-4786
sowsley@cs.northwestern.edu

Kristian J. Hammond
Northwestern University
2133 Sheridan Road
Evanston, IL 60208
(847) 467-1012
hammond@cs.northwestern.edu

Larry Birnbaum
Northwestern University
2133 Sheridan Road
Evanston, IL 60208
(847) 491-3640
birnbaum@cs.northwestern.edu

Abstract

In this paper, we describe a system called TagAssist that provides tag suggestions for new blog posts by utilizing existing tagged posts. The system is able to increase the quality of suggested tags by performing lossless compression over existing tag data. In addition, the system employs a set of metrics to evaluate the quality of a potential tag suggestion.

Coupled with the ability for users to manually add tags, TagAssist can ease the burden of tagging and increase the utility of retrieval and browsing systems built on top of tagging data.

General Terms

Algorithms, Design, Human Factors, Languages

Keywords

Tags, Blogs, Case Base Reasoning, Tag Suggestions, Text Classification

1. Introduction

The explosion of user-created content on the web has given rise to tagging systems aimed at annotating this content with meta-information, usually in the form of keywords to help in organizing, browsing, and searching. From image tagging, (Flickr), to web page tagging (Del.ico.us), to social tagging (Facebook), these system have become popular and are heavily utilized across the Web.

Different types of tagging systems have emerged for different types of content. Social/Collaborative tagging systems have allowed resources to be tagged by multiple people and shared amongst a group or community of people. Others only allow the owner of the content to define the set of tags that will be associated with the content (YouTube). The focus of our system is the latter type.

Many would argue that the power of tagging lies in the ability for people to freely determine the appropriate tags for a resource without having to rely on a predefined lexicon or hierarchy [10]. The dynamism of tagging systems allows the creators of content to quickly adapt and incorporate novel concepts and changes in terminology without having to rely on a standardized tag corpus. Others argue that large user generated tag corpora, or folksonomies, will converge on a shared vocabulary that can assist people in finding and browsing information. The power of the vocabulary is based on the collaborative nature of its creation, where individual contributors organically learn and extend the domain language.

Unfortunately, since tagging systems do not enforce fixed or controlled vocabularies for tag selection, the tag space suffers from many of the same problems of traditional free text Information Retrieval systems. Golder et. al., [6] identified three major problems with current tagging systems: polysemy, synonymy, and level variation.

Polysemy, in tagging systems, refers to instances where a single tag can have multiple meanings. For example, a blog post tagged with “caterpillar” could indicate that the post is about etymology or could be interpreted as containing information about construction equipment.

Multiple tags having the same meaning is referred to as synonymy. Cases of synonymy may be morphological variation (“blog” versus “blogs”) or semantic similarity (“news” versus “current events”). In blog post tagging systems, synonymy is particularly problematic as authors must rely on their own intuition to pick the appropriate tag to represent the content of their post, with no guarantee that two users who have posts on the same topic will chose the same tag to describe their content.

The third problem identified is level variation. This refers to the phenomenon of users tagging content at differing levels of abstraction. Content can be tagged at a “basic level” or at varying

ICWSM’2007 Boulder, Colorado, USA

levels of specificity which is often based on a blogger's expertise or requirements; a post may be tagged as "car" (basic) but an enthusiast might find "Volkswagen MKIV 2001 Golf" more appropriate.

Another factor that complicates current blog tagging systems is the lack of clear functional pressure to make tagging consistent, stable, and complete for use in applications dealing with collaboration/community, clustering, and search. Some authors tag their posts to make them visible to a larger community, using general categorical descriptions such as "politics" and "shopping". Some bloggers use tags to organize their posts for their own consumption and interpretation, using non-content descriptive tags such as "random" or "toRead". Other may choose to use very specific or niche tags that are highly descriptive of the content of the post (i.e., "why I hate best buy", "instructions for cooking grandma's apple pie").

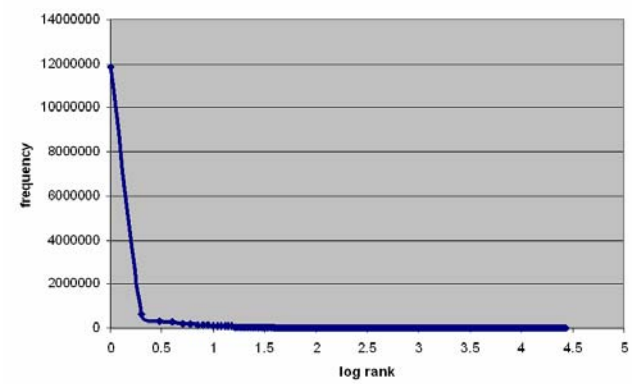

Figure 1: *The distribution of tags and their frequencies*

The lack of a shared or controlled vocabulary has resulted in the explosion of unique tags in the blogosphere. At last glance, Technorati [16], one of the leading blog aggregators, was tracking over 60 million blogs and nearly 11.5 million tags. A sample of English blog data provided by Technorati from a 16 day period in late 2006 shows nearly 403,000 unique tags with a mean frequency of 343.1, median of 8, and mode of 1. The most frequently occurring tag is "Weblog" with 6,695,762 occurrences. Nearly 22% (88,212) of tags in the system only occurred once and only 5.7% of the tags occur more frequently than that average frequency (343.1). A sample of the distribution of tags and their frequencies in the sample is illustrated in Figure 1. Because of the size of the dataset, we ordered the tags by frequency and sampled every 15 tags (using log distribution to present the data). The data show that a small percentage of all existing tags are actually reused by bloggers. The data also show that there is a very large portion of existing tags that are used rarely, making up a significant "long tail" [17] distribution. In practice, incredibly rare tags (that are assigned to posts very infrequently), often referred to as "meta-noise", are unlikely to be used for retrieval due to their idiosyncratic nature. For example, consider the likelihood of a user utilizing the tags shown in Table 2, a small random sample of tags that only occur once in our data sample, in order to browse or search for content.

Content creator-owned tagging systems (those without a collaborative component), especially suffer from inconsistent and idiosyncratic tagging. It is not that people are uninterested in tagging as they often do tag their posts, but when given no insight into how other bloggers tag, the task of tagging becomes difficult and the results are less than useful for retrieval and browsing.

For this reason, systems need to be built that can *suggest* appropriate tags for content. The goal of such a system is that users can see how other users are tagging content and choose to leverage the shared vocabulary or create new tags when necessary. The overall results would be much more useful tagging systems without undercutting the prospect or the power of folksonomies.

Given the state of this tag space, we aimed to build a system that would assist users in tagging their own blog posts by providing them with a set of relevant tags. The approach we take is that of a mediated suggestion system. That is, the tagging system does not apply the suggested tags automatically, rather it suggests a set of appropriate tags and allows the user to select tags from the set they find appropriate. The selected tags are then applied to the post and incorporated into a larger corpus of post to tag associations. The system also provides a text box where users can add additional tags not suggested by the system, allowing new and emerging tags to be introduced and utilized in the system. This approach is appealing as it is able to leverage large scale data processing, while manually checking the suggestions using minimal human intervention. This type of approach also fundamentally changes the tagging process from generation to recognition -- requiring less cognitive effort and time [18][19].

In addition, by providing consistent suggestions to user, we provide the opportunity for the tag space to organically converge on a consensus for tag selection. Such a convergence would help alleviate the issues of synonymy and level variation as users would have an indication as to the types of tags that other bloggers are using to describe similar content. Convergence would also help increase recall by reducing the number of idiosyncratic tags, reducing the meta-noise in the tag distribution.

offshoreman 'the people', no way!!.., black colored lilies, manila pride, circuit asia, console customization, eyelash perming, shadow watcher, miss yah, all female horror, scripture snippets, insomnia due to quail wailing, streetball china, marriage age, Wresler, could this possibly be a poem..., Coritsol, goodbye highbury, 1.2 glossary of terms

Table 2: *A sample of tags that only occur once on the blog post corpus*

2. Related Work

There have been several system developed that automatically generate appropriate tags for a given blog post.

The first of these systems, named *TagIt* [1], uses Naïve Bayes text classification methods to determine the appropriate tag to apply to a new post. While the results of the system were promising, it was not proven to scale beyond the 330 tags in the training set or evaluated against blog posts with multiple tags.

Brooks et al, [4] developed another system that automatically tagged blog posts based on the top three terms extracted from the post, using TFIDF scoring. While this technique resulted in closer, more focused clusters of posts it only provides unigram tags that literally appear in the blog post.

Most similar to our system is the collaborative filtering *AutoTag* system developed by Gilad Mishne [11]. *AutoTag* finds similar tagged posts and suggests some set of the associated tags to a user for selection. While our system uses a similar technique, we have improved on *AutoTag's* performance by introducing tag compression and case evaluation to filter and rank tag suggestions.

3. The System

To help define the task and guide the development of our system, we instituted a functional framework for blog tags. Functionally, we wanted tags to help users to retrieve and browse posts based on a contextual relationship to a tag or set of tags. Although the tag space is currently used for browsing and retrieval, the lack of consistency is the tag space leaves a "long tail", a significant but rarely seen portion of the tagged blogosphere.

Our solution to this problem takes the form of a recommendation system that leverages tags previously associated with content to recommend tags for new content. The system take a new, un-tagged post, finds other tagged posts that are similar to it, aggregates the tags associated to those posts, and then recommends a set of those tags to the end user. In practice, the system considers several factors when selecting tags to suggest, including the frequency of occurrence of the tags in previous posts. To increase the effectiveness of our system we did not treat every unique tag in the tag space as an atomic symbol, but rather looked for areas that we could automatically group morphologically related tags in a lossless compression. Discovering similar sets of tags allows the system to utilize a larger portion of the tagged posts in order to provide recommendations.

To adapt the constant flow of new blog content being created and to prevent the data from becoming stale, the system also allows for the continuous update of content in the training system. This allows the system to react to changes in the blogosphere including the addition of new tags and the drift of existing tag senses.

3.1 Tag Compression

The tag space compression stage of the system has two primary phases. The first phase, referred to as the tag normalization phase, takes each tag in the system and performs a set of operations aimed at reducing it to its root form. The second phase, called the compression validation phase, validates the normalization done in the first phase.

3.1.1 Tag Normalization

Because of the uncontrolled nature of the tag data, token scrubbing is performed to trim white space and punctuation from each tag. Each tag is also stemmed to a morphological root using Porter's stemmer [12]. In addition, tags that contain more than one atomic word are tokenized, stemmed, and placed in alphabetic order. This helps resolve tag variations such as "news and politics" and "politics and news" which both resolve to "and new polit". The resulting buckets of morphologically related tags (i.e., those with the same root form) are used as the hypothesis set of final compression. The first round of tag normalization reduced the overall tag set by 18.581% (402638 unique tags to 327820 unique roots).

3.1.2 Compression Validation

The second phase of the tag space compression, called validation, aims to confirm each grouping from the normalization phase to ensure that the system has not grouped tags with different meanings under the same normalized root. Morphological normalization is a relatively aggressive technique that poses the risk of over-stemming, where two terms that share the same root but not the same meaning are collapsed together. While "production", "product", and "producers" share the same morphological root "produc", they each have distinct meanings. Techniques that validate morphological normalization choices using dictionaries and thesauri have been developed, but fail to adapt to novel word senses and lack entries for current technological and/or blogging terminology [4]. We chose, instead, to validate our normalization choices by leveraging relationships between tags as they are used within our blog corpus.

To perform the validation, the set of related tags is generated for each tag within the corpus. The related tags for tag *t* is defined as the set of tags, *rel(t)*, that co-occur alongside *t* in posts in the corpus. Along with the set of related tags, we also collect the number of times the co-occurrence appeared within the corpus. The related tag set provides a reasonable set of related or similar concepts to the usage of tag *t* in the corpus. Cattuto el al [5] statistically analyzed collaborative tagging data and determined the non-trivial nature of co-occurrence relationships amongst tags. They further demonstrated that the relationship between co-occurring tags and how the frequent grouping of "generic" tags with "narrow" tags may encode semantic hierarchical organization. The use of co-occurring tags has also been used to some extent in tag clustering [3] and tag visualization systems [7]. Table 3 shows the top 10 related tags for "Iraq", illustrating the effectiveness of related tags to help define a topic space.

related tag	**count**
Politics	462
Bush	410
War	357
Terrorism	275
Iran	230
News	193
Middle East	171
War on Terror	146
Republicans	141
Military	133

Table 3: *List of the top 10 related tags and their co-occurrence frequency for "Iraq"*

To perform the validation, each set of tags that share a common normalized root is placed in a bucket, B_i, with *n* total buckets, where *n* is the total number of unique roots. The most frequently occurring tag in each bucket is assigned as the centroid ($centroid_i$) of B_i and its related tags $rel(centroid_i)$ are retrieved and normalized. For each of the remaining candidate tags {t_1 $t_{2...,}$ $t_{k,}$} in B_i the related tags $rel(t_k)$ are retrieved and normalized and an overlap score P is calculated between $rel(centroid_i)$ and $rel(t_k)$. The frequency of co-occurrence for each tag is used to weight the intersection score to favor frequently co-occuring pairs. If P is above an acceptable a tunable threshold F, the compressed relationship is maintained. If P is undefined, meaning that t_k did not have any related tags, the compressed relationship is also maintained. If P is less than F, t_k is labeled an outlier and placed in a new bucket, B_{n+1}. The algorithm is then recursively invoked until all tags have been placed in an appropriate bucket with similar tags. At this point, each bucket is assigned a representative that is the most common variant (*mcv*) of the particular morphological root based on its frequency within the training corpus. The *mcv* is subsequently used as the actual tag suggestion to the user, representing the most common use over the entire corpus. For any tag t, the $mcv(t)$ represent the most common variant from B_i which contains t.

The end result of the validation phase was a modest reduction of overall compression. The reduction of the uncompressed raw tag data went from 18.581% (327820 unique roots) to 17.001% (333790 unique roots), but still a large improvement over the total number of unique raw tags (402638 unique tags).

tag	**related tag**	**count**
apple	Mac	333
apple	Technology	240
apple	iPod	217
apple	Software	190
apple	Microsoft	143
apple	iTunes	135
apples	Fruit	60
apples	Apple	50
apples	Recipes	33
apples	Food	31
apples	Cooking	26
apples	Oranges	20

Table 4: *A sample of the tags and their co-occurrence frequency for "apples" and "apple"*

More interesting was this technique's ability to identify the actual context in which a tag is used in the corpus, which may be different than information contained within a standard dictionary or thesauri. For example, the tags "apple" and "apples" were combined during the first phase of tag compression, as they share a common morphological root. A dictionary may very well tell us that "apples" is a plural form of "apple" or even that "Apple" is the name of a computer and software manufacturer, but does not say anything about how the tag "apple" or "apples" is used by most users in the blogsphere. The related tag set, however, provides clear evidence that the tag "apple" almost exclusively refers to the technology firm, while "apples" refers to the fruit. The differences between these two related tag sets are illustrated in Table 4. Conversely, this strategy was able to verify many more compression decisions by proving the semantic relationship between the two variants. An example of this type of validation is illustrated in Table 5.

The end result of the compression stage of our system is the creation of a collapsed tag space that condenses the various morphological variants and promotes one variant to represent each set during tag suggestion.

tag	**related tag**	**count**
dogs	Pets	364
dogs	Dog	108
dogs	Puppies	100
dogs	Cats	82
dogs	Puppy	74
dogs	Dog Training	71
dog	Dogs	108
dog	Pets	93
dog	Puppy	83
dog	Puppies	76
dog	Dog Training	72
dog	Dog Clothes	69

Table 5: *A sample of the tags and their co-occurrence frequency for "dogs" and "dog"*

3.2 Tag Suggestion Engine

Once the tag space has been normalized and compressed, the other component of the system, the Tag Suggestion Engine (TSE) is used to suggest a set of tags to a user. The TSE operates on the principal of leveraging existing tagged data to provide appropriate tag suggestions for new content. This approach is very similar to Case-Based Retrieval Systems [8][13][14] (CBR) where solutions for new cases are determined by retrieving similar, solved cases from a large corpus of labeled examples and applying those solutions (or transformations of those solutions) to the new problem. Mishne's *AutoTag* system takes a very similar approach to tag recommendation, comparing his system to a recommender system, a successor to the classic CBR systems.

The TSE contains three main components: the case-base, the case retriever, and the case evaluator which are all implemented as web services so they can easily be deployed and integrated with existing blog post authoring tools.

3.2.1 The Case Base

In order to leverage previously tagged blog posts, they had to be available for retrieval from a repository. For this purpose, we use the off-the-shelf Lucene search engine. We have had success in the past [15] using Lucene, as it was an easy-to-use and configure repository for our previous text classification system. We used Lucene's default content analyzers to index each tagged post in our corpus along with a unique post identifier so we can retrieve the associated tags for the post. Once indexing has been completed, Lucene is able to take a text-based query and provide a relevance ranked list of posts that contain one or more of the specified query terms using a simple vector space comparison model.

3.2.2 *The Case Retriever*

The second component of the TSE is the case retriever. The main purpose of this component is to take a new post (target post) to be tagged and to retrieve other similar posts from the case-base. To generate a compressed representation of the target post, we employ a simple TFIDF unigram scoring schema using the corpus to determine the document frequency component of each term's score. In addition, we set minimum and maximum selection thresholds (St_{min} and St_{max}) for term inclusion in the query. The use of St_{max} helps in filtering out common corpus-wide unigrams from the query and St_{min} aides in identifying cases where unigrams are misspelled or non-English. In addition to the unigram-based term vector, we also identify salient bi-grams (using TFIDF scoring) from the target post and include those in our final query. To be included, the bigram must not contain a term with score below the minimum scoring threshold and must occur at least twice in the post. To prevent favoring the vocabulary of any one blogger, we only process the first two posts from any particular blogger. We experienced the best performance by using lengthy queries (a maximum length of 30) and retrieving the top 35 results from Lucene for evaluation.

3.2.3 *The Case Evaluator*

Once similar posts have been retrieved from the case base, the unique post identifier is used to retrieve information about the blog the post was contained in as well as the tags that have been assigned to that post. For each tag that is retrieved, the most-common variant, *mcv(t)*, is found, utilizing the tag compression. Each tag is then scored and/or filtered using five metrics that evaluate the relative usefulness of the tag *t*. The five different scoring/filtering parameters for tag evaluation are as follows:

Frequency – *freq(t)* is the number of times that *t* appears as an associated tag in the top 35 results returned by the case evaluator. A tag is discarded if *freq(t)* < 2. This is effective under the assumption that the stronger the consensus of the tag across different bloggers, the higher the potential utility of the tag.

Text Occurrence – whether the raw tag *t* or the *mcv(t)* appears in the actual target post. The appearance of the tag in a post may be an indicator of relevance.

Tag Count – *count(t)* is the number of times tag *t* (and all of its variants) have been used in the training corpus. The tag is discarded if *count(t)* < 2. Tags that are utilized more in the blogosphere have a higher potential of being useful to a user.

Rank – the relative rank of the blog that contained the post that was assigned tag *t*. The rank of a blog is analogous to its overall popularity in the blogosphere as determined by the number of inbound links.

Cluster – using the same clustering technique that we use to validate the tag compression, we determine whether any of the candidate tags are members of topically related clusters by comparing the pair-wise similarity of each tag's related tag set. This allows us to find the semantic relationships between tags that are not morphological variants. Topical agreement amongst disparate tags in the results set is an indication of their potential utility.

After each tag has been processed and scored, the individual scores are weighted and combined to form an aggregate tag score. The tags are ordered by score and filtered once again by score. The goal is to provide only the best tag suggestions to the user. To this end, we only return tags that have an aggregate score greater than the mean score for all the tag candidates.

4. Evaluation

To evaluate *TagAssist*, we used data provided to use by Technorati, a leading authority in blog search and aggregation. Technorati provided us a slice of their data from a sixteen day period in late 2006. The data contains only English content with 8.1M blog posts from 2.7M unique blogs. Out of these posts, 1.9M posts are tagged with an average of 1.75 tags per post.

To gauge the effectiveness of our system compared to other similar systems, we developed a version of our tagging suggestion engine that was integrated with the raw, uncompressed tag data and did not use the case-evaluator for scoring, aside from counting frequency of occurrence in the result set. This baseline system returned the top 10 tags ordered by frequency. In addition to comparing our system's performance against the baseline, we were also interested in examining how our system compared to the original tags that were assigned to the post in our training corpus.

Tag Set	Accuracy
Original Tags	48.85%
TagAssist	42.10%
Baseline	30.05%

Table 6: *Accuracy values for human evaluation of the three tag sets*

Our study used human judges to evaluate the appropriateness of tags for a post. For testing data, we randomly selected posts, with 2 or more originally assigned tags, from our blog corpus and presented them to a human judge along with a list of tags generated by our system, a list of tags generated by a baseline system, and the tags originally assigned to the post in our corpus. The post was presented in a web page along with a list of tags and corresponding checkboxes. The judges were asked to read each post and then check the boxes next to tags they thought were appropriate for the post.

In all, we collected and analyzed 225 responses from a total of 10 different judges. Table 6 lists the precision values for each of the tag sets, that is, the average percentage of tags in each set that the judges found appropriate. As the results show, 48.85% of tags originally assigned to a post were determined to be relevant by our judges. Our method resulted in a precision of 42.10% and the baseline came in third with a precision of 30.05%. While *TagAssist* did not outperform the original tag set, the performance is significantly better than the baseline system without tag compression and case evaluation.

Given that less than 50% of tags originally assigned to a post are not deemed as relevant by third party judges, we found it less useful to perform automatic evaluation of our system by calculating precision/recall values for our system against the original tags. It also makes it difficult to automatically tune the system when there is not reliable data to gauge the system's performance. For the sake of comparison to other systems, we performed the evaluation by processing 1000 posts through our system and the baseline system and then comparing the suggested tags against those originally assigned. We did not use string distance to compare tags, but chose instead to use exact string equality for comparison. As a result, the precision/recall values are much lower than the results of human evaluation. Table 7 shows the results of automated evaluation for both our method and the baseline. The results show almost identical recall values between both systems with our system outperforming the baseline in precision.

Suggestion Method	Precision	Recall
TagAssist	13.11%	22.83%
Baseline	7.66%	23.14%

Table 7: *Precision and recall values for automated testing over 1000 posts using exact tag matching*

5. Discussion

Our evaluation shows that *TagAssist* is able to provide relevant tag suggestions for new blog posts. The novel tag compression algorithm and case evaluation component helped increase the precision of the system without reducing recall. A system that can effectively propose relevant tags has many benefits to offer the blogging community.

Firstly, shifting the tagging process from a purely generative process to one that require users to recognize appropriate tags significantly reduces the cognitive burden and increases performance of blog post tagging. If we work to make the tagging process easier, we are more likely to increase the number of bloggers who tag their posts. If more users tag posts, we are likely to increase the richness of the folksonomy and provide more content to tag search interfaces.

Secondly, providing users with tag choices based on their actual usage in the blogosphere can accelerate the convergence of tag vocabulary to be more consistent and useful for retrieval and browsing.

6. Future Work

One of the interesting results from our human evaluation is the relevance score for the original tags assigned to a blog post. On average, less that 50% of these original tags were deemed as relevant by third-party judges. Given that our system is trained off this data, we believe we can drastically improve the performance of our system by identifying the blog posts have been *effectively* tagged, meaning that the tags associated with the post are likely to be considered relevant by other users. We are currently investigating techniques to identify these effectively tagged blog posts and hope to incorporate it into future versions of *TagAssist*.

Automatic evaluation of tag suggestion engines is also critical to building effective systems. Given issues with the perceived relevance of user-generated tags, it is important that we have a ground truth testing corpus to evaluate system performance. But with no functional constraints on the use of tags, it is difficult to build a gold standard that everyone can agree on.

User feedback is another component that we would like to add to *TagAssist*. Given that we are providing a list of tags to a user and having them select the most appropriate ones give a blog post, we can use their feedback to help tune our system. A more involved scenario might ask users to evaluate each tag, providing the system with explicit feedback on the utility of each tag. In addition, given that we allow users to freely enter additional tags, we can use that information to improve *TagAssist*.

We are also exploring novel way of presenting the suggestion list, besides using plain text. We would like to explore ways of presenting the tags that reflect their usage in the larger blogosphere. The interface to our system will be critical in making it effective and usable by various bloggers.

Acknowledgements

Our thanks to Technorati for providing us with the datasets used to create *TagAssist*. Particularly, we'd like to thank Peter Hirshberg, Adam Hertz, and Ian Kallen for their support on this project. Also, thanks to Shannon Bradshaw for his insight on this project.

References

[1] Al-Shurafa, H. Blog Post Labeling using Naïve Bayes Text Classification Methods.

[2] Bateman, S., Brooks, C. and McCalla, G. Collaborative Tagging Approaches for Ontological Metadata in Adaptive ELearning Systems. In *Proceedings of the Fourth International Workshop on Applications of Semantic Web Technologies for E-Learning (SW-EL 2006) in conjunction with 2006 International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH2006)* (June 20, 2006). Dublin, Ireland, 3-12.

[3] Begelman, G., Keller, P., and Smadja, F. Automated Tag Clustering: Improving search and exploration in the tag space. In *Proceedings of the Fifteenth International World Wide Web Conference (WWW2006)* (Edinburgh, Scotland, May 22-26, 2006). ACM Press, New York, NY, 2006.

[4] Brooks, C., and Montanez, N. Improved Annotation of the Blogopshere via Autotagging and Hierarchical Clustering. In *Proceedings of the Fifteenth International World Wide Web Conference (WWW2006)* (Edinburgh, Scotland, May 22-26, 2006). ACM Press, New York, NY, 2006.

[5] Cattuto, C., Loreto, V., and Pietronero, L. Semiotic Dynamics and Collaborative Tagging. http://www.isrl.uiuc.edu/~amag/langev/paper/cattuto06semioticDynamics.html, 2006.

[6] Golder, S.. Huberman, B. Usage Patterns of Collaborative Tagging Systems. In *Journal of Information Science (2006)*, 32(2). 198-208.

[7] Hassan Montero, Yusef, and Herrero-Solana, Víctor. Improving Tag-Clouds as Visual Information Retrieval Interfaces. In *Proceedings of the International Conference on Multidisciplinary Information Sciences and Technologies (InSciT '06)* October 25-28, 2006, Mérida, Spain.

[8] Leake, D. CBR in Context: The Present and Future. In *Case-Based Reasoning: Experiences, Lessons, and Future Directions*. AAAI Press/MIT Press, 1996, 1-30.

[9] Marlow, C., Naaman, M., Davis, M., and Boyd, D. HT06, Tagging Paper, Taxonomy, Flickr, Academic Article, ToRead. 2006. In *Proceedings of Hypertext 2006*. ACM Press, New York, NY, 2006.

[10] Mathes, A. Folksonomies: Cooperative classification and communication through shared metadata. In *Computer Mediated Communication*, 2004.

[11] Mishne, G. AutoTag: A Collaborative Approach to Automated Tag Assignment for Weblog Posts. In *Proceedings of the Fifteenth International World Wide Web Conference (WWW2006)* (Edinburgh, Scotland, May 22-26, 2006). ACM Press, New York, NY, 2006.

[12] Porter, M. F. An Algorithm for Suffix Stripping. In *Readings in Information Retrieval*. Morgan Kaufmann Publishers Inc., San Francisco, CA, 1997, 313-316.

[13] Riesbeck, C., and Schank, R. *Inside Case-based Reasoning*. Erlbaum, Northvale, NJ, 1989.

[14] Schank, R. *Dynamic Memory: A Theory of Learning in Computers and People.* Cambridge University Press, New York, NY, 1982.

[15] Sood, S., Owsley, S., Hammond, K. and Birnbaum, L. Reasoning Through Search: A Novel Approach to Sentiment Classification. In Submission.

[16] Technorati, www.technorati.com, 2006.

[17] Anderson, S. "The Long Tail", *Wired*, Oct. 2004

[18] Goldstein, D. G., and Gigerenzer, G. The Recognition Heuristic: How ignorance makes us smart. In *Simple heuristics that make us smart.* Oxford University Press, Oxford, England, 1999.

[19] Tulving, E. *Elements of Episodic Memory*. Oxford University Press, New York, New York, 1983.