Automatic Tag Recommendation for the Web 2.0 Blogosphere Using Collaborative Tagging and Hybrid ANN Semantic Structures

SIGMA ON KEE LEE and ANDY HON WAI CHUN
Department of Computer Science
City University of Hong Kong
Tat Chee Avenue, Kowloon Tong
HONG KONG

Abstract: - This paper proposes a novel approach to automatic tag recommendation for weblogs/blogs. It makes use of collective intelligence extracted from Web 2.0 collaborative tagging as well as word semantics to learn how to predict the best set of tags to use, using a hybrid artificial neural network (ANN). The use of "tags" has recently become very popular as a mean of annotating and organizing everything on the web, from photos, videos and music to blogs. Unfortunately, tagging is a manual process and limited to the users' own knowledge and experience. There may be more accurate or popular tags to describe the same content. Collaborative tagging is a recent technology that creates collective intelligence by observing how different users tag similar content. Our research makes use of this collective intelligence to automatically generate tag suggestions to blog authors based on the semantic content of blog entries.

Key-Words: - Web 2.0, Blog, Collaborative Tagging, Intelligent Systems, Machine Learning.

1 Introduction

Web 2.0 represents the "second generation" of Web applications with new technologies that allow people to work, collaborate and share knowledge in innovative manners. An important characteristic of Web 2.0 is that it embraces the power of the web to harness collective intelligence of its users. In particular, the rise of blogging is one of the most highly touted phenomena of the Web 2.0 era. Weblog or blog is an important innovation that makes it easy to publish information, engage discussion and form communities on the Internet. Blogs are web sites consisting of content (or "entries") that are dated and displayed in reverse chronological order. Many people think of blogs as online public journals. Its easy-of-use has made it the leading decentralized publishing technology in the Web 2.0 world. Basically anyone with access to the Internet can now publish content, allowing anyone to quickly and easily disseminate their opinions to a very wide audience. The contents of blogs may vary from personal journals, markets or product commentaries, to news and current affairs. In addition, the number of blogs has also grown exponentially to estimated tens of millions to over a hundred million blogs by the end of 2006. Therefore, creating technologies that allow people to easily and quickly find high quality blog content that they are interested in is a very important but difficult task. Our research in automatic tag recommendation is a way to maximize the chances that blog contents will reach those potentially interested in it through more accurate tagging that makes use of collective intelligence of the billion Internet users.

The tens of millions of blogs in the world are interlinked to form what is known as the blogosphere. To support this Web 2.0 phenomenon, special technologies such as custom blog search, analysis engines, and systems that employ specialized information retrieval techniques were invented, all with the aim to making finding information in the gigantic blogosphere easier. In particular, tagging is a popular technique to facilitate the organization of blog entries. Tags can be thought of as key words or key phrases attached to documents or objects (blog entries, photos, music, or videos) to help describe those objects. The use of keywords is of course not new. It has been used in categorizing or indexing in the traditional library systems. Keywords provide an easy way to categorize, search, and browse content. Tagging is a term to describe the new set of Web 2.0 technologies to support keywords online, such as collaborative tagging.

One of the characteristics of Web 2.0 collaborative tagging is the ingenious use of "open vocabularies" instead of a formalized ontology. Tags are not selected by professional annotators, but by the average content authors themselves. Although this may sound counter-intuitive, but tags created organically without any centralized control is more interesting that a formalized ontology as it harnesses the collective intelligence of hundred of millions of

people! With a rich pool of tags, tags can group documents into broad categories [5] that can solve the problem of synonyms, pluralization and misspelling by using the shared knowledge of other users. The use of tags has organically produced a “folksonomy” [17], [12], short for “folk taxonomy”, a system in which the meaning of a tag is determined by its use among the community as a whole. Technorati.com is one of the most popular sites related to the tagging of blogs, while sites like furl.com and del.icio.us help users collaborate on tagging webpages. Flickr.com is an example of using tags to describe photos.

In this paper, we describe a novel approach to automatic tag suggestion that makes use of collective intelligence from collaborative tagging combined with semantic-driven ANN learning to produce a set of most relevant tags for the user to select from. The result of ANN learning is a network that encodes richness and subtleties in mapping content to tags. The results produced will be a list of weighted or prioritized tags that are most relevant to the given blog. In simple terms, our system basically learns how to tag by observing how other humans tag their own blog content. This learned knowledge is then used to automatically generate tag suggestions for new blog entries.

2 Research Background

Tagging is a way to organize content through labeling. It tries to associate meaning to online content such as blogs, photos, videos and music. Tags are keywords or key phrases that can be associated with content as a simple form of metadata. To a computer, tags serve as a set of atomic symbols that are associated with an object. Unlike the keyword systems used in libraries in which users select keywords from a predefined list, users can choose any string to use as a tag. The idea of using tags to annotate content recently become quite popular within the blogging community. The idea of tagging is not new, photo-organizing tools have used tagging for ages, and HTML has had the ability to allow META keywords to describe a document since HTML 2.0 [4] since 1996.

In a tagging system, an item of content will typically have one or more "tags" associated with it. Tagging software automatically provides links to other items that share the same tag, or even to specified collections of tags (via AI clustering). This allows multiple "browseable paths" through the content to facilitate search and retrieval of related items.

While using tags is flexible and easy, tagging is not without its drawbacks. Tags are just strings without any semantic meaning. For example, the tag "apple" might refer to the fruit, or Apple Computer. The lack of semantic distinction in tags can lead to inappropriate connections between items. In addition, selection of tags is highly dependent on the individual. Different people may use drastically different terms to describe similar content. A case in point, items related to a version of Apple Computer's operating system might be tagged both "OSX", "Tiger", and possibly many other terms. Users of tagging systems have to make “intelligent guesses” to determine the most appropriate tag to use or search for.

Collaborative tagging offers an interesting alternative to current efforts. Collaborative tagging is portrayed as a kind of shared knowledge. It allows users to share their tags with other users. It allows users to publicly tag and share content, so that they can categorize information for themselves, and they can browse the information categorized by others.
Tag classification, and the concept of connecting sets of tags between web/blog servers, has lead to the rise of folksonomy classification over the internet. Larger-scale folksomonies have the benefit of using tagging as astute users of tagging system will monitor/search the current use of "tag terms" within these systems. They tend to use existing tags in order to easily form connections to related items. In this way, evolving folksomonies define a set of tagging conventions through eventual group consensus.

In collaborative filtering, patterns in user preferences are mined to make recommendations based on like users’ opinions—individuals who have shared taste in past will continue to do so. Examples include Ringo [16] and GroupLens [13] as well as e-commerce sites such as Amazon.com. Fab [2] combined content-based and collaborative recommendation. However, collaborative filtering suffers from some well-known limitations [14], such as, the sparsely of user profiles, the latency associated with pre-computing similarity information, and the difficulty in generating predictions about new items. Some of these limitations will also apply to the system presented here.

3 Auto Tag Suggestion Algorithm

Our AT:tag algorithm, consists of a Training Phase which involves ANN learning, and an Execution Phase which is responsible for tag suggestion generation.

3.1 Training Phase

In the Training Phase, we first use robots to crawl the web to collect blogs that have already been manually tagged. Some of these blogs will become part of the training set while others will be used for testing. The main objective of the training phase is to learn how blog content is associated to tags. To keep our experiments manageable, we will limit our robots to focus on subsets of the blogosphere. For example, blogs related to "hiking" only or blogs related to "rock climbing."

The algorithm for the Training Phase consists of 3 main stages:

- Stage 1: Keyword Extraction
- Stage 2: Semantic Processing
- Stage 3: ANN Learning

3.2 Stage 1: Keyword Extraction

We use both statistically method and the lexical resources method to perform keyword extraction. This is further divided into 3 steps:

- Step 1: extract single keywords using TFIDF score. (statistically based)
- Step 2: compute co-occurrence frequency (statistically based)
- Step 3: check bigrams using WordNet (lexical resources based)

Step1: extract single keywords using TFIDF score. The TFIDF score [15] is calculated by the following formula (1):

$$TFIDF(word) = termFreq(word) x \log\left(\frac{|corpus|}{DocFreq(word)}\right)$$

where:

- *termFreq(word)* indicates the number of times that a word occurs in the blog entry being processed. It is computed using:

$$termFreq(word) = \frac{n_i}{\sum_k n_k}$$

- $|corpus|$ indicates the total number of message in each user.
- *DocFreq(word)* indicates how frequently a word appears in that corpus.

The TFIDF will score individual words within text documents in order to select concepts (represented by keywords) that accurately represent the content of the document. This will cause commonly used words to have a very low TFIDF score, and rare words to have a high TFIDF score. Because the TFIDF score is based purely on how frequent a single word appears in the text, we will need to supplement this with information on a word's relevance in terms of other words.

Step2: compute co-occurrence frequency in the same blog. In our AT:tag algorithm, the keyword extraction stage will also consider bigrams selection where two continuous words are considered as one item. Co-occurrence frequencies are computed for the extracted keywords. In our experiments, we filter out word-pairs that have frequency less than 5. In particular, we try to extract special bigrams that do not appear in our dictionary. Higher frequency bigrams will have higher weightings in our algorithm.

Step3: check bigrams using WordNet. WordNet [10] is a freely available electronic dictionary developed by the Cognitive Science Laboratory at Princeton University. It has been used for text summarization [3] and other natural language processing tasks. In this project, we use WordNet to help with our bigram selection [9]. When bigrams are extracted from the blog, we search WordNet to check if the bigrams are common phrases or not.

The result of "Stage 1: Keyword Extraction" is a set of keywords or key phrases to represent the blog content. In "Stage 2: Semantic Processing," we further enrich this representation by supplementing the keywords/phrases with semantics.

3.3 Step 2: Semantic Processing

After generating a set of keywords/phrases for a blog, our AT:tag algorithm then use WordNet to extract semantic information. This process helps provide lower-level semantics to our representation and allow us to relate blog with different set of keywords but with similar "meanings."

The design of WordNet, was inspired by current psycholinguistic theories of human lexical memory. Words are organized into synonyms sets (i.e. synsets) each representing one underlying lexical concept. For example: the set of lexical items {car, automobile auto, machine, motorcar} constitutes one synset representing the concept corresponding to the gloss/definition: "4-wheeled motor vehicle; usually propelled by an internal combustion engine". Different semantic relations link synsets together into different hierarchies (e.g. IS-A and PART-OF relations).

For each keyword/phrase generated from our Stage 1 processing, we select the first synset produced from WordNet. The resulting synset information is used as additional semantics to describe a blog. For example, the keyword "computer" is related to this synset: {computer, computing machine, computing device, data processor, electronic computer, information processing system}. The collection of synset produced from our keywords/phrases is used to represent the semantic content of a blog.

3.4 Step 3: ANN Learning

Learning in AT:tag is performed using an artificial neural network (ANN). The structure of the network is shown below:

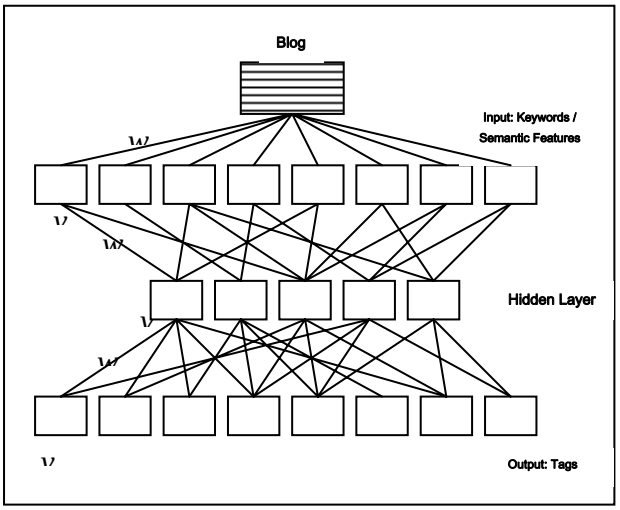

Fig. 1. The structure of the ANN used for learning

There are three layers in our ANN - input layer is the feature layer with weighting, one hidden layer, and an output layer which represents the predicted tags. ANN learns the association of keywords/phrases and semantic features to tags. Learning is needed as the selection of tags can be influenced by several different features. The weights learned determine the contribution of each feature to the selection of a tag. We used backpropagation [18] for learning.

"Stage 3: ANN Learning" is further divided into:

- Step 1: Initialize Network
- Step 2: Compute Errors
- Step 3: Back propagate the errors
- Step 4: Adjust weightings (learning)

Step 1: Initialize Network. The following are the ANN initialization procedures. The learning algorithm is described after that.

Procedure 1: w_i initial value: normalized feature occurrence frequency

- keyword/key phrase node:

$$\begin{cases} 0 : b\log\_does\_not\_have\_feature \\ 1 : b\log\_have\_that\_feature \end{cases}$$

$$y_i = g(w_i) = \begin{cases} 0 : no\_feature \\ w_i : have\_feature \end{cases}$$

Procedure 2: w_j initial value: random number value [0,1]

- $y_j = f\left(\sum w_j y_i\right)$

Procedure 3: w_k initial value: random number value [0,1]

- $y_k = f\left(\sum w_k y_j\right)$

where f(x) = Sigmoid function = $\frac{1}{1+e^{-x}}$

In our AT:tag algorithm, the output of our ANN is a set of suggested tags, each with a priority weighting. Since there are more than one output nodes, we modify the standard backpropagation algorithm using a hybrid approach.

To produce a prioritized list of tags to suggest, the output is not only a single node. After evaluating the activation function, each output node (each representing one tag) will have an activation value. The higher the value, the higher the ranking will be for a tag in our prioritized suggestion list. Since there are multiple outputs, the backpropagation error calculation will be different. We map the multiple errors to a single error using a regression function. The number of actual tags present in a blog is of course fixed. However, the predicted outputs from the ANN consist of the entire set of tags stored in AT:tag. Therefore, we need to select the same amount of outputs for both the predicted output and actual output. We select the highest N predicted outputs for the N actual outputs for comparison. We then normalize using the highest activation value node in the predicted output, because when there is single actual output, no weighting need to be changed. For multiple outputs, except the highest value node, other nodes must below the activation level of 1. Therefore, although the number of predicted outputs is same as the actual, an error may still exist since the activation level may be too low. If so, the network will continue with its training cycle.

Therefore, we have some basis for matching predicted output versus actual. If the predicted output exists in the actual output, then the error is positive. Otherwise, if the predicted output does not exist in the actual output, then the error is negative. The reason for having different sign of error is to adjust the learning point to move to a more reasonable direction. We then add up all the N error to produce the overall error. Using this regression function, if all tags matched, the error will be small. We will skip the learning progress if the error is below a pre-defined threshold. Only significant errors will trigger learning and the changing of link weights.

Step 2: Compute Errors. Procedures involved in computing the errors during back propagation:

Procedure 1: top $\vec{x}$ predicted outputs nearest to 1 (highest ranking nodes).

Procedure 2: compare to original $\vec{z}$ tags for that blog

Procedure 3: select same number (N) of output nodes $\vec{x}$ as actual $\vec{z}$.

Procedure 4: normalize output node value (using highest value as 1)

- to avoid changing weighting if only have 1 actual output, i.e. N = 1

Procedure 5: actual output of the tag

- $= \begin{cases} 1 : if\ \_tag\_exists \\ 0 : if\ \_tag\_not\_exists \end{cases}$

Procedure 6: error is then calculated by: $\delta_k = \vec{\gamma}(\vec{x} - \vec{z})$

where

$$\gamma = \begin{cases} +1 : if\ \_desire\_tag\_match\_actual\_tag \\ -1 : if\ \_deaire\_tag\_not\_match\_actual\_tag \end{cases}$$

Procedure 7: if error is less than a threshold T, learning procedure will be skipped.

Step 3: Back propagate the error. Based on the above the learning parameters are computed as:

$$\delta_i = \sum w_j \delta_j$$

Step 4: Adjust weightings (learning). The weights of the links are then adjusted according to these formulae for each layer of the ANN:

$$w_i' = w_i + \eta \delta_i \frac{\partial w_i}{\partial \sum w_i} x_i$$

$$w_j' = w_j + \eta \delta_j \frac{\partial y_j}{\partial \left(\sum w_j y_i \right)} y_i$$

3.5 Execution Phase

After the Training Phase has been completed, our algorithm then makes use of the resulting ANN to automatically suggest tags. In this phase, the user submits a completed blog entry to AT:tag and gets a list of prioritized tag suggestions in return. When a blog entry is received by AT:tag, it first extracts keywords/phrases and semantic features to represent that blog entry. The extraction method is the same as the knowledge extraction in the Training Phase. After that, AT:tag uses the extracted features to activate the ANN. Results from the ANN is presented to the user as prioritized tag suggestion.

4 Results and Comparisons

In our experiment, we first used Technorati API (http://technorati.com/) to search for the following keywords: {ai, ajax, alone, apple tag, apple, art, baby, book, bush, car, card, cat, christmas, comedy, computer, crazy, dairy, dog, dressing, education, environment,fire,fish,friends,games,google,government,happy,health,hiking,home,house,idol,internet,job,kiss,law,life,lonely,love,mobile,money,mountain,mp3,music,nature,news,play,pop,popular,robot,rock,sad,school,science,sleepy,snow,song,sport,star,sweet,tag,tagging,technology,telephone,tools,universe,web 2.0,web tag, web, weblog, windows, word, world, youtube}. The search is restricted to English blogs. The results were analyzed to retrieve the first 500 permalinks for each of the target keywords. It is because our experiment requires full content of the blogs and their corresponding tags. For each of the permalinks, we extract blog content and tags. Out of the 35417 links, we found 4401 pages with tags. We further divided these pages into training set and testing set. We performed preprocessing on these data files, such as removing special characters and html tags and comments. Finally, blog content is split into a series of keywords. For each of the training data and testing data, we extract keywords using our keyword extraction method and calculate frequency of keywords to prepare the input for ANN. The following is an example data extracted from a blog:


##Contents:
directx
3d
graphics
apis
madison
lockwood
designed
microsoft
highest
versions
introduction
vista
operating
direct3d
developers
hardware
introduced
windows
games
development
version
sophisticated
video
cards
ati
card
produce

Currently, we are still fine-tuning our algorithm and parameters. We believe the main reason that our approach works is that it makes use of collective intelligence provided in Web 2.0 collaborative tagging. Tags suggested are learned from this collective intelligence and will be more acceptable to the user. In addition, we capture subtleties and richness in blog content using semantic information provided in WordNet. The same mechanism allows us to handle differences in how people tag similar blogs as well as how people express similar ideas with different wordings. The collective intelligence of millions of blogs also allows us to reduce the chance of human errors in tagging. In comparison, there is an existing weblog tagging systems called AutoTag [8] which finds the most similar blogs and then collect all the tags in those blogs for ranking and filtering. The disadvantage of this approach is that tags that are not in the similar blog entries will not be considered. It cannot suggest new tags if the tags are not already used in one of the similar blogs. In our method, all tags related to the semantic content of the blog will be proposed regardless of whether that set of tags have been used in another blog before or not.

Another related work that parsing each post's content for tags are [6] and [1], which is done by analyzing blog content. In our method, we relate blog content to their corresponding tags.

Yahoo! [19] has a different approach to collaborative tag suggestions using a greedy heuristic approach. Their algorithm emphasizes correlation within tags and reputation of user. Our method uses online dictionary to enrich the information extracted from blogs. Our system adjusts rating between the networks since we don't need to store the score of every tag with each object. ANN provides another possible approach for collaborative tag suggestion since ANN has a long history of success for similar problems. There may be possible for combining ANN is heuristic method for further improvement.

5 Future Plans

We believe our approach can also be used in other situations where tags or metadata need to be generated. For example, it can be used to automatically generate metadata for HTML files by analysis the semantic content of a webpage. As an enhancement, we plan to investigate the benefits of analyzing the generated tag suggestion list using WordNet with the possibility of reducing the number of tags with similar meanings. In addition, we plan to investigate whether the use of common sense knowledge bases, such as OpenCYC, might further improve the quality of tags produced. Other potential areas for research include exploring the use of data mining and clustering to our learning algorithm.

6 Conclusion

This paper presented a new and power feature for the Web 2.0 blogosphere – automatic tag suggestion generation. Our novel AT:tag approach uses a hybrid ANN to learn subtle mappings between rich semantic features and tags. Our algorithm leverages on the vast amount of collective intelligence that is available in Web 2.0 collaborative tagging to produce results that are in resonance with other users. The result is that the generated tags are similar to those produced by humans.

References:

1. A. Beard, "Bumpzee Adds Amazing Feature – Autotagging."http://andybeard.eu/2007/02/bumpzee-adds-amazing-feature-autotagging.html, 2007.
2. Balabanovic, M., and Shoham, Y. "Content-based, collaborative recommendation." Comm. ACM 40(3):67-72, 1997.
3. Barzilay R., Elhadad M. "Using lexical chains for test summarization," In Proc of Intelligent Scalable Text Summarization Workshop (ISTS), 1997.
4. Berners-Lee, T., and Connolly, D. "Hypertext markup language specification – 2.0." Technical Report RFC 1866, MIT/W3C, 1996
5. Christopher H. Brooks and Nancy Montanez. "An Analysis of the Effectiveness of Tagging in Blogs," Proc 2006 AAAI Conf, 2006
6. Christopher H. Brooks and Nancy Montanez. "Improved Annotation of the Blogosphere via Autotagging and Hierarchical Clustering.", Proc. of the 15th Int'l WWW Conf, Edinburgh, Scotland, 2006
7. Fellbaum C. (ed.) "WordNet: An electronic lexical database", MIT Press, 1998
8. Gilad Mishne, "AutoTag: a collaborative approach to automated tag assignment for weblog posts", Proc. of the 15th Int'l Conf on World Wide Web, 2006
9. L. Bentivogli and E.Pianta. "Beyond lexical units: Enriching wordnets with phrasets.", In Proc. of the Research Note Sessions of the 10th Conf of the European Chapter of the Assoc for Computational Linguistics (EACL'03), pp.67-70, Budapest, Hungary, April 2003
10. Miller, George A., Richard Beckwith, Christiane Fellbaum, Derek Gross, & Katherine J. Miller, "Wordnet: An on-line lexical database." Int'l Journal of Lexicography, 3(4):235–312, 1990.
11. Miike S., Amano S., Uchida H., Yokoi T. "The Structure and Function of the EDR Concept Dictionary." Terminology and Knowledge Eng. (Vol. 1), 1990.
12. Quintarelli, E. "Folksonomies: power to the people," ISKO Italy-UniMIB meeting. http://www.iskoi.org/doc/folksonomies.htm, 2005.
13. Resnick, P.; Iacovou, N.; Suchak, M,; Bergstorm, P.; and Riedl, J. "GroupLens: An Open Arch for Collaborative Filtering of Netnews," In Proc. ACM Conf. on Computer Supported Cooperative Work, 175-186, 1994.
14. Sarwar, B. M.; Karypis, G.: Konstan, J.; and Ridel, J. "Analysis of recommender algorithms for e-commerce.", In Proc. 2nd ACM E-Commerce Conf. (EC'00), 2000.
15. Salton, G., and McGill, M. J. "An Introduction to Modern Information Retrieval." New York: McGraw-Hill, Inc., 1983
16. Shardanand, U., and Maes, P. "Social Information Filtering: Algorithms for Automating "Word of Mouth", In Proc. ACM CHI'95 Conf., 210-217, 1995.
17. Shirky, C. "Folksonomy" http://www.corante.com/many/archives/2004/08/25/folksonomy.php, 2004.
18. Tariq Samad. "Back-propagation is significantly faster if the expected value of the source unit is used for update" In Int'l Neural Network Society Conf Abstracts, 1988.
19. Zhichen Xu, Yun Fu, Jianchang Mao, and Difu Su Yahoo! Inc. "Towards the Semantic Web: Collaborative Tag Suggestions" Proc of the Collaborative Web Tagging Workshop at the WWW, 2006