Automatic Tag Recommendation for Web 2.0 Blogosphere
by Extracting Keywords from Similar Blogs

SIGMA ON KEE LEE and ANDY HONWAI 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 Vector Space Model (VSM), comparing similar blogs from the web and statistical methods. Tags are popular means of annotating and organizing content on the web, from photos, videos and music to blogs. Unfortunately, tagging is just 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.

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. Weblogs or 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 [4] 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” [12], [5], 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. It uses the VSM to find similar document in the Internet and using extra information to extract the potential tags for user to select. The result is using the users’ behavior on the Internet by the statistical method, and the heuristic function to extract the possible tags from other users. The results produced will be a list of prioritized tags that are most relevant to the given blog. This system is based on the giant social network to collect the common knowledge from different Internet users. The other users in the Internet will help our system 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 [2] 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 [11] and GroupLens [8] as well as e-commerce sites such as Amazon.com. Fab [1] combined content-based and collaborative recommendation. However, collaborative filtering suffers from some well-known limitations [10], 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 Our Automatic Tag Suggestion Algorithm
Our Algorithm consists of 2 key parts – expanding the original blog by finding similar blog in the internet, and using the extra information (the similar blog) to enrich the tag suggestion by similar content.

3.1 Part I: Expanding the original blog by finding similar blog in the internet
In this part, we use robots to grab similar blogs from the web and analyze contents using both statistical and lexical methods. This analyzed content is stemmed and using a simple Vector Space Model (VSM) we locate contents that are closest to the original input blog. Very often, a few documents topically close to the specified document can be retrieved from a large corpus though search engines, and these neighbour documents are believed beneficial to evaluate and extract useful information from the document. The underlying assumption is that the topic-related documents can provide more knowledge and clues for the single document. From human’s perception, users would better understand a document if they read more topic-related documents. In the experiment, we use a blog search engine to process the large corpus. In ways, the web can be considered as a large worldwide database. The selected similar blogs will be further analyzed to enrich the information in Part II.

Stage 1: Keyword Extraction

Step 1: Using WordNet to select the noun
In our research, we use WordNet [7] to divide English into groups: nouns, verbs, adjective and adverbs. This allows us to categorize every word into groups. For every keyword/phrase, we remove the white space and punctuation and then trim the strings. Each word is further processed using WordNet to identify whether it is a noun or not.

Step 2: Calculate the TFIDF score for every noun
Before running the system, a large corpus is created by grabbing blogs though Internet using a robot. We grab blogs to create a corpus for different topics; removing common words that appear in different topics. Using TFIDF [9], we score individual words within text documents 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.

Stage 2: Select query
We first sort the TFIDF (noun) list in descending order. Ideally, the highest TFIDF scored keywords will be used for blog search engine queries. However, the highest TFIDF score may contain some rare words that are not suitable to use as keywords to describe blog content. We use Yahoo! search API to retrieve a web count for the keywords. The larger the web count number, the more informative is the search query. We further sort the web count list for every noun extracted from the previous stage, combining the highest TFIDIF score and the highest web count together. The resulting nouns are found to be accurate representations of the blog content.

We only use nouns to represent blog content. We use TFIDF to extract topics because TFIDF can identify the most important words in blog entry. However, different people use different wordings to represent similar concepts. We found that some of the words used in a blog are rare. Our purpose is to search the web to find any possible similar blog that matches the current blog. If we only use rare words for query, we might not retrieve any blog that are similar to the current blog. Therefore, the Yahoo! search API is used to identify which words are more informative. The web count from the Yahoo! search API represents how popular is the keyword. The higher the web count, the more blogs will be retrieved. Therefore, we combine the TFIDF score together with the web count from Yahoo! to select the most informative and representative words for the query to search for similar blogs.

Stage 3: Retrieve potential blogs
From the previous stage, we get keywords that are most informative and representative. In our experiments, we select the top 3 keywords to
compose the query. We then use the Technorati blog search engine to search for similar blogs using these 3 keywords.

In our experiments, we selected the top 15 blog links that are returned from Technorati. We assumed that blog search engine will retrieve the most similar blogs sorted with the most related query. The content of those 15 blogs are extracted using RSS or exploring its HTML content.

Stage 4: Select similar blogs by VSM
We first remove all special characters from the blogs as well as white spaces and punctuation, and then trimmed. The blogs extracted from Technorati form a blog pool. The original blog is the query blog in VSM, and is used to select the most similar blog from the blog pool.

Procedure 1: We first produce the Vector Space Model. Every keyword in the query and the blog pool formed the dimension m of the vector.

\[
\begin{align*}
1 \times m \text{ Query Vector : } & [tfidf \ldots \ldots 0 \ldots \ldots tfidf] \\
15 \times m \text{ Blog pool matrix: } & \begin{bmatrix}
tfidf & \ldots & \ldots & 0 & \ldots & \ldots & tfidf \\
\vdots & \ldots & \ldots & \ldots & \ldots & \ldots & \vdots \\
\vdots & \ldots & \ldots & \ldots & \ldots & \ldots & \vdots \\
0 & \ldots & \ldots & 0 & \ldots & \ldots & 0 \\
\vdots & \ldots & \ldots & \ldots & \ldots & \ldots & \vdots \\
0 & \ldots & \ldots & \ldots & \ldots & \ldots & 0 \\
\vdots & \ldots & \ldots & \ldots & \ldots & \ldots & \vdots \\
0 & \ldots & \ldots & \ldots & \ldots & \ldots & 0 \\
\vdots & \ldots & \ldots & \ldots & \ldots & \ldots & \vdots \\
0 & \ldots & \ldots & \ldots & \ldots & \ldots & 0 \\
\vdots & \ldots & \ldots & \ldots & \ldots & \ldots & \vdots \\
0 & \ldots & \ldots & \ldots & \ldots & \ldots & 0 \\
\vdots & \ldots & \ldots & \ldots & \ldots & \ldots & \vdots \\
0 & \ldots & \ldots & \ldots & \ldots & \ldots & 0 \\
\vdots & \ldots & \ldots & \ldots & \ldots & \ldots & \vdots \\
0 & \ldots & \ldots & \ldots & \ldots & \ldots & 0 \\
\end{bmatrix}
\end{align*}
\]

We take the TFIDF scores and compute the term weights. These columns can be viewed as a sparse matrix in which most entries are zero. We treat weights as coordinates in the vector space, effectively representing documents and the query as vectors. To find out which document vector is closer to the query vector, we use the similarity analysis introduced in the next step.

Procedure 2: First for each document \(D_i\) and query \(Q\), we compute all vector lengths (zero terms ignored) by:

\[
|D| = \sqrt{\sum w_{ij}^2}
\]

and

\[
|Q| = \sqrt{\sum w_{ij}^2}
\]

Procedure 3: We compute all dot products (zero products ignored) between the query and the documents (blogs).

\[
Q \cdot D_i = \sum w_{Q,j} w_{D,j}
\]

Procedure 4: We calculate the similarity values by cosine similarity

\[
\cos \theta_i = \frac{\text{Sim}(Q, D_i)}{|D_i| |Q|} = \frac{\sum w_{Q,j} w_{D,j}}{\sqrt{\sum w_{Q,j}^2 \sum w_{D,j}^2}}
\]

Procedure 5: Finally we sort and rank the documents in descending order according to the similarity values. In our experiment, we choose the top 2 similar blogs from the similarity values.

3.2 Part II: Using the extra information to enrich the tag suggestion by similar content
After finding the similar blogs from the Internet, we further analyze the blog content to enrich the information in tag suggestion. We analyze and extract the possible keywords from the query blog together with the extra information. The similar blogs are treated as the extra information.

Step1: Calculate the TFIDF score for the original Blog and the similar blogs
From Part I, we have found two similar blogs for us to enrich the information, the similar blog 1 (SB1) and similar blog 2 (SB2). In Part II, we first calculate the TFIDF score for query blog and similar blogs for every keyword.

Step2: Adjust the score and combine it in the similar blogs
Two similar blogs are selected in Part I. For the similar blogs, each blog has its TFIDF table. According to VSM, we know which blog is more similar to the query blog. We then group the keywords in these two blogs together for further calculation. We add bias information to the similar blogs. Since we trust similar blog 1 more than similar blog 2, we combine their TFIDF score together to form a new mixed score.

\[
\text{NewScore} = \text{TFIDF}_{in} \times 60\% + \text{TFIDF}_{in} \times 40\%
\]

Step3: Create the matrix of mutual information
After calculating the score for both query blog and similar blogs, we use these information to create another heuristic for adjust the final score of the keywords. In our experiment, we use pointwise mutual information for addition information to the heuristic.

The pointwise mutual information using data collected by information retrieval was suggested by Turney [13] as an unsupervised measure for the evaluation of the semantic similarity of words. It is based on word co-occurrence using counts collected over very large corpora (e.g. the Web). Our query collects web count from Yahoo! search API.

\[
MI(x, y) = \log \frac{\text{hits}(x \text{ AND } y) \times \text{WebSize}}{\text{hits}(x) \times \text{hits}(y)}
\]

The WebSize value is needed for the overall calculation of the text-text similarity metric. We approximate the value of WebSize to \(7 \times 10^{11}\), which is the value used by Chklovski [3] in co-occurrence experiments involving Web counts.

We create the matrix of mutual information. We use Yahoo! search API to calculate the mutual information. For the query blog, we calculate the keywords corresponding with its content, finding out the information of that word affecting the whole content. For the similar blog, we find the mutual information of the word compare to the query content.

**Step4: Average the mutual information**

After the MI matrix has been formed, we calculate the average MI value for each keyword. Each row of the MI matrix represents one keyword respecting to other content.

\[
\text{Avg}, i = \frac{\sum \text{score}_i}{n}
\]

The progress is executed on the query MI matrix and the similar blog MI matrix. And we will get two vectors, one contains the query average MI vector, another is the similar blog average MI vector. The dimension of these vectors is the same as the total number of keywords \(n\).

**Step5: Combine the final score using TFIDF and mutual information**

At this stage, we have two different data for us to calculate heuristic value to the final score.

- TFIDF of the query and similar blog
- Mutual information of query and similar blog.

Since the extra information may not be 100% trusted, therefore, a confidence value that will affecting the impact for the similar blog information. The final score is calculated and sorted by:

\[
\text{Score}_{i} = \text{SourceTFIDF} \times \text{SourceMI} \times 0.8 + \text{SimBlogTFIDF} \times \text{SimBlogMI} \times 0.3
\]

**Step6: Result keywords suggested**

The final score is sorted and formed with a final score list. We choose the highest final score keywords as tag suggestion to the user.

4 Results and Comparisons

In our experiments, the user directly input the query blog for tag suggestion. We first use a common blog search engine to search for English blog contents to act as testing data. From the blog search engine results, we extract the blog content and use as the query to generate suggestion tags. Each testing data is pre-processed to reduce noise. As a result, we split the blog content into a series of keywords. This is used as query to the Technorati API, which returns a set of matching blogs. The top 15 blogs and their contents are then analyzed. We then use these results to find two similar blogs by simple vector space model.

We further use these similar blog to calculate extra information. In the stage of forming the mutual information matrix in both query blog and similar blog, we only use the keywords in the 20 highest TFIDF score. It is because the Yahoo! Search API limits 50,000 query 24 hours per IP. To calculate the mutual information, by the formula:

\[
MI(x, y) = \log \frac{\text{hits}(x \text{ AND } y) \times \text{WebSize}}{\text{hits}(x) \times \text{hits}(y)}
\]

we need to access Yahoo! 3 times. Therefore, if we use all the keywords to form a mutual information matrix, the access limit will be too restrictive. Therefore, we only choose 20 keywords for calculating the mutual information matrix. For each tag suggestion, we will have at least \(20 \times 20 \times 3 \times 2 = 2400\) queries to the Yahoo! Search API. Thus, we limit the mutual information matrix to make our experiments more manageable. After that, we use the information of TFIDF score and the mutual information of keywords to generate the tags suggestion to the user.
We believe our approach produce quality tags because it makes use of collective intelligence provided in Web 2.0 collaborative tagging. Tags suggested are learned from this collective intelligence which comes from the Internet users. Since our approach dynamically harnesses collective intelligence from the web in real-time, the tags suggested by our system will change as Internet users’ writing habit and vocabulary changes.

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 [6] 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.

Furthermore, Yahoo! Inc [14] also has a different approach for collaborative tag suggestions. In their method, they are using greedy heuristic approach while we are using extra information for the tag suggestion. Their algorithm emphasizes correlation within tags and the reputation of the user.

5 Conclusions
This paper presented a new and powerful feature for the Web 2.0 blogosphere – automatic tag suggestion generation. We believe these technologies can also be used in other situations where tags or metadata need to be generated. For example, these technologies can also be used to automatically generate metadata for HTML files by analysis the semantic content of a webpage.

References:
5. Emmanuele Quintarelli, “Folksonomies: power to people.”, paper presented at ISKO Italy-UniMIB Meeting, Mi, June 2005