

Blog Profile Extraction using Text Categorization

Nurcan Durak

nurcan.durak@louisville.edu

Knowledge Discovery & Web Mining Lab

University of Louisville,

Louisville, KY, USA

ABSTRACT

Our study aims at extracting dominant subjects from a weblog automatically. To fulfill this aim, we first develop a blog entry categorization system that assigns categories to an individual blog entry. Then, using the blog entry categorization system, we extracted weighted weblog profiles from a variety of weblogs, that were shown to accurately reflect the dominant subjects.

Keywords

Blog Entry Categorization, Blog Profile Extraction, Blog Recommendation System.

1. INTRODUCTION

Bloggers are generally free to write what they want on their weblogs. There is no limitation on subject, language, or time. However, most bloggers tend to write entries about certain topics more often than others. When we look at an ordinary weblog, we can grasp certain dominant topics that characterize of that weblog. Thus, readers can describe a weblog page more or less by using its dominant topics. For example, "Blogger X may be writing mostly on music, literature, and movies." Thus readers tend to extract a rough profile of the weblog, and if their interests intersect with the profile of the weblog, they can become loyal readers of the weblog. Moreover, if weblog profiles can be extracted based on *standard* categories, then blog recommendation systems and blog search systems could bring many more accurate results to the interested readers.

In this study, our motivation is to extract the weighted weblog profiles from a weblog automatically, for later use in weblog recommendation systems. Profiles can also be useful to provide a quick list of subjects of a weblog before spending effort to read through many entries in the weblog.

2. Blog Entry Categorization System

One weblog page consists of many blog entries in chronological order. In a weblog, there can be one or more bloggers. A blog entry has the following attributes: permalink, date, title, content, comments, tags, images, and out links. Well defined data structures are given below. In the definition, (*) means 0 or more times and (+) means one or more times.

Weblog = {*Weblog URL*, *RSS URL*, *Weblog Title*, *Blogger*⁺, *(Blog Entries)*⁺, *Weblog Profile*}

Blog Entry = {*Permalink*, *Date*, *Title*, *Blogger*⁺, *Content*, *Out-links*^{*}, *Comments*^{*}, *Tags*^{*}, *Categories*^{*}}

Most blog entries reflect personal opinions in a rather informal way. Therefore the language of blog entries tends to contain many pronouns, daily language patterns, phrasal verbs,

adjectives, idioms or abbreviated subject verb combinations such as "you're", "we've", etc. Most of those words are not helpful in representing the categories of the entry. So we added those words into the stop word list. Our stop word list contains around 600 words that include common verbs such as "make, get, do", conjunctions such as "and, or, because", pronouns such as "I, you, we", etc. The stop word list is applied on the content and title of the blog entry in both training and testing the categorization system. After getting rid of the stop words, we applied Snowball stemming techniques on the blog content and title.

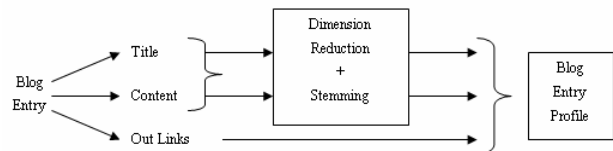


Figure 1. Blog Entry Profile Extraction

After applying stemming, we computed the term frequency values of each term in the content, and then sorted all content terms according to their term frequency in descending order. We also extracted out-links from the content. Figure 1 depicts the preprocessing of a blog entry and acquiring blog entry profile. We define our blog entry profile formally as follows.

Blog Entry Profile = {
[Sorted terms from the content and their frequency values],
[Sorted terms from the title and their frequency values],
[Out-links from the content and their number of occurrence] }

We used a training data set with pre-defined categories and with enough blog entries under each category. Using this training data set, we extracted term frequency (tf) and inverse document frequency (idf) values from the blog entry collection set in each category. Using all of the blog entries under a category, we extract a category profile which is a vector consisting of weighted terms. For constructing a category file, we use content information, title information and out-links seen in the blog entry contents. The formal definition of a *category profile* is given below.

Category Profile = {
[Top 200 terms and their tfidf values from Content],
[Top 100 terms and their tfidf values from Title],
[Top 50 out-links in content and their tfidf values]}

After extracting category profiles for pre-defined categories, we can predict the categories of *new* blog entries. When a new blog entry comes, the blog entry profile of the new entry is compared to existing category profiles to get the category

similarity values (CSV) of new blog entry to existing categories [1]. Categories which have higher similarity values than given threshold are assigned to new blog entry. It is worth noting that one blog entry may be categorized under *more than one category* in our system, and if a blog entry profile is not similar enough to any profiles, then it may not be categorized under “Others” category.

3. Blog Entry Categorization Experiments

We collected blog entries from www.engadget.com which is one of the top 5 weblogs according to www.blogpulse.com statistics. Entries in engadget.com are mostly technology related and written by different bloggers. All of the entries are filed under pre-defined categories by their authors. Some pre-defined categories are “Cell phones”, “Laptops”, “Gps”, “Robots”, etc. Some entries were filed under more than one category.

From the training data, we built weighted category profiles for each category. One example of the category profile is given below. The first part is the weighted *content* vector, while the second part is the weighted *title* vector, and the third part is the weighted *out-links* vector.

Cellphones = { [Content: {iphon 0.361}, {handset 0.313}, {phone 0.286}, {mobil 0.216}...], [Title: {iphon 0.478}, {handset 0.290}, {android 0.223}, {sprint 0.167}.....], [Out-links: {store.apple.com 0.006}, {www.nxp.com 0.004}, {www.nokia.com 0.001}...] }

Separately from the training data, we kept 30 blog entries from each category to show the effectiveness of the results more understandably. We obtained 79% for precision value which means 79% percent of predicted categories reflected true categories. and 65% for recall value. To explain why we get lower entry recall values than entry precision values, we should note that if one blog entry has three or four categories, our system assigns the most similar categories to an entry. We use high thresholds to avoid predicting too many categories for an entry. If we assigned more categories to an entry, obviously the entry recall value would be high, but at the expense of lower precision.

4. Weblog Profile Extraction System

The main goal of this study is extracting weighted profiles from weblogs. For this purpose, every blog entry should first be categorized under pre-defined categories. After finding the categories of these entries, we can construct a weighted profile for their weblog.

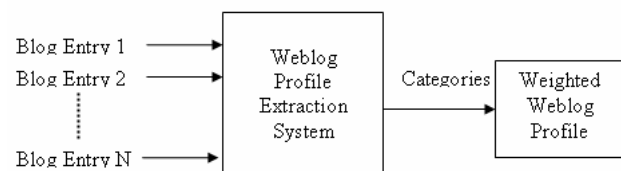


Figure 2. Profile Extraction System from a Weblog

Figure 2 shows the structure of our weblog profile extraction system. Each blog is categorized under pre-defined categories

using our Blog Entry Categorization system. A blog entry is categorized or not depending on the threshold values and other parameter weights in the CSV based categorization. We keep a counter for each pre-defined category and counter of an category is incremented each time that another blog entry is categorized under that category. After computing the counter values for each pre-defined category, then we sort them in descending order. Then the weight for each category is computed by dividing the counter value by the number of entries. If the weight of a category c_i is lower than a given threshold value, then we do not include it in our profile list, since we aim to extract *dominant* categories from the weblog. Adjusting the threshold value to decide which categories should be placed in the profile list is done by experimentally using cross validation experiments.

We collected blog entries from other technology related weblogs to build weblog profiles. Table 1 shows the weighted profiles of two weblogs. Weights show percentage distribution of the predicted categories for all blog entries in that blog. Since a threshold value of 5% was chosen, categories weighted under 5% were not shown in the weblog profiles. By looking at the weighted profiles of the weblogs, we can determine which weblogs write about which subjects dominantly.

Table 1. Sample Weighted Weblogs Profiles

Blog1: www.gizmodo.com		Blog2: www.cybernet.com	
Category	Weight	Category	Weight
Cellphones	14	Cellphones	19
Entertainment	12	Gaming	8
Gaming	11	Handhelds	8
Digital Cameras	8	Gps	6
Laptops	6	Hdtv	6

5. Conclusion and Future Prospects

We have presented a system for blog entry categorization and weblog profile extraction, obtaining 79% in blog entry precision for the current predefined categories. Our weighted profile extraction system is also successful in discovering the dominantly written subjects on a weblog. We plan to use these weblog profiles as part of a recommendation strategy. For the time being, the profile extraction system is limited by the number of pre-defined categories. It would also be interesting to generalize our categorization and profiling system to work for all kinds of weblogs, and to extend the blog categorization system with semantic category trees that arrange categories according to a predefined taxonomy.

6. References

[1] Sebastiani F. 1999. A Tutorial on Automated Text Categorisation, Pisa (Italy)