Wikis, Blogs, Bookmarking Tools
Mining the Web 2.0
(WBBTMine'08)

International Workshop at
the European Conference on Machine Learning and
Principles and Practice of Knowledge Discovery in Databases
in Antwerp, Belgium, 15 September 2008.

Preface

A host of Web 2.0 applications have rapidly emerged in recent years, resulting in grass-root creation of knowledge spaces on the Web. Cooperative Web publishing tools, like wikis and blogs, and resource sharing services, like social bookmark systems and photo sharing systems, have thrived thanks to their ease of use and low threshold to entry.

Web 2.0 applications yield rich data sets for data mining, with interwoven dimensions of data: text, multimedia, hyperlinks, tags and social interactions. Unlike domains over which data mining has historically been applied, the data coming from Web 2.0 applications emerges bottom-up from potentially from millions of different heterogeneous and independent sources.

One early form of data mining over large data sets of this kind was collaborative filtering, an algorithmic approach to building recommender systems from user ratings and actions. New data mining techniques extend to arbitrarily complex data and data mining tasks. Work in this new and exciting application area include: data analysis and exploration, application and adaptation of well-known data mining and machine learning algorithms, and the development of entirely new algorithms. This workshop was put together to support research advances in the analysis of Wikis, blogs, social bookmarking systems in particular, and Web 2.0 in general. Its particular focus is the interactions between knowledge discovery and Web 2.0.

All submissions were reviewed by at least three reviewers. This led to the selection of five high-quality papers. Three of them deal with *tagging and folksonomies*. The fourth focuses on *blogs* and the fifth analyzes *Wikipedia*.

In *BaggTaming - Learning from Wild and Tame Data*, Kamishima, Hamasaki and Akaho address a new machine learning problem, "taming," that involves two types of training sets: wild data and tame data. A wild data set (e.g., folksonomy tags) is less consistent, but is much larger in size than a tame set. Conversely, a tame set (e.g., labels from a controlled vocabulary) has consistency but not as many data. They develop methods for learning more accurate classifiers by exploiting the strong points of each training set.

In *Topical Structure Discovery in Folksonomies*, Subasic and Berendt combine information retrieval, Semantic Web and social web approaches to searching the Web by including general topic categories as a part of tagging systems and learning a hierarchical network of tag associations as an ontology.

In *Identifying Ideological Perspectives of Web Videos using Patterns Emerging from Folksonomies*, Lin and Hauptmann develop a classifier that can automatically identify a web video's ideological perspective on a political or social issue based on the pattern of tags emerging from folksonomies.

Ishikawa, Tsuchida and Takekawa deal with another important data type on Web 2.0: blogs. They propose *Clustering blog entries based on the hybrid document model enhanced by the extended anchor texts and co-referencing links*, employing a document vector space model where weights of noun terms vary depending on positions within the texts of blog entries as search results. In particular, link structures of blogs and anchor texts are exploited for weighting.

Another contribution focuses specifically on one of the best-known "products" of Web 2.0 and its usefulness for generating language-processing resources: Wikipedia. Kliegr, Svátek, Chandramouli, Nemrava and Izquierdo propose *Wikipedia as the Premiere Source for Targeted Hypernym Discovery.* They investigate the reasons that make Wikipedia articles good targets for lexico-syntactic patterns, and they suggest that the main reason is the adherence of its contributors to Wikipedia's Manual of Style, regardless of whether the article covers a popular topic or not.

This year's ECML/PKDD Discovery Challenge addresses similar topics to our workshop. As a result, the two events have been combined into a full-day event covering advances and research in data mining over Web 2.0 data. The two topics of the challenge were spam detection and recommendation of tags in folksonomies. These challenges represent key application areas for data mining and machine learning.

We thank all participants of the workshop for their contributions, our reviewers for their invaluable help in shaping this workshop and the organizers of the ECML/PKDD 2008 conference for their support.

August 2008

Bettina Berendt, Natalie Glance, and Andreas Hotho

Conference Organization

Programme Chairs

Bettina Berendt
Natalie Glance
Andreas Hotho

Programme Committee

Sarabjot Singh Anand
Mathias Bauer
Janez Brank
Michelangelo Ceci
Ed H. Chi
Brian Davison
Miha Grcar
Marko Grobelnik
Kristina Lerman
Pasquale Lops
Ernestina Menasalvas
Dunja Mladenic
Ion Muslea
Giovanni Semeraro
Barry Smyth
Ian Soboroff
Myra Spiliopoulou
Gerd Stumme
Michael Wurst
Marco de Gemmis
Maarten van Someren

External Reviewers

Fabio Calefato
Miha Grcar
Liangjie Hong
Jian Wang

Table of Contents

BaggTaming — Learning from Wild and Tame Data

Toshihiro Kamishima, Masahiro Hamasaki, and Shotaro Akaho

National Institute of Advanced Industrial Science and Technology (AIST),
AIST Tsukuba Central 2, Umezono 1–1–1, Tsukuba, Ibaraki, 305–8568 Japan,
mail@kamishima.net⟨http://www.kamishima.net/⟩,
hamasaki@ni.aist.go.jp, s.akaho@aist.go.jp

Abstract. We address a new machine learning problem, *taming*, that involves two types of training sets: wild data and tame data. These two types of data sets are mutually complementary. A wild data set is less consistent, but is much larger in size than a tame set. Conversely, a tame set has consistency but not as many data. The goal of our taming task is to learn more accurate classifiers by exploiting the strong points of each training set. For this task, we develop a new algorithm, BaggTaming, which is a modified version of bagging. Such an algorithm would be useful for predicting tags in collaborative tagging services, such as del.icio.us. With such services, users can register Web documents, assign tags to them, and share these tags. Such tags are similar to the notion of class labels that have been adopted in supervised learning. The tags are poorer in consistency than well-managed labels, but more tags are available. If we consider tags as wild data, and labels as tame ones, Web pages could be more accurately classified with our taming technique.

1 Introduction

In this paper, we address a new learning problem, which we call *taming*, and develop a method for solving this problem. The learner for this taming requests two types of training data sets, *tame* and *wild*. The label information of tame data is carefully maintained based on a consistent target concept, which we actually want to learn. In contrast, wild data are not so well maintained; thus, some labels are consistent with the target concept, while some others are not. Additionally, we assume that wild data are much more abundant than tame data. This assumption is reasonable, because it is generally difficult to provide a large tame data set due to its high maintenance cost. The goal of the taming is to acquire more accurate classifiers by exploiting wild data than those learned only from tame data.

One example of such wild data is found in collaborative tagging systems. Collaborative tagging services such as del.icio.us[1] enable users to register their favorite Web pages. To these registered pages, users can assign tags, which are words that describe the contents, characteristics, or categories of the tagged pages. These tags are useful for searching or classifying their own registered pages. In addition, these registered pages and assigned tags can be shared among users of the service. With these shared

[1] http://del.icio.us/

tags, users can search the pages that other users have registered, and they can find like-minded users.

This social tagging process differs from a traditional classification scheme. In the case of a document repository or library collection, the materials are classified under well maintained and unified criteria. That is to say, the labeling scheme is highly restricted, and the semantics of the labels is strictly defined. In contrast, social tags are labeled based on each user's personal criterion. Users can freely choose their favorite tags; thus, the semantics of social tags can vary greatly. Golder and Huberman have pointed out such inconsistency among collaborative tags [1]. They discussed the causes of the variation in the semantics of the tags. One variation involves the degree of specificity. For example, the official page of the programming language python can be tagged by either the specific word, "python,", or the general one, "programming." Another cause is polysemous words and phrases, which have many related senses. For example, one may consider the term "data mining" as a statistical technique for marketing research. But another user may use this term to indicate techniques for analyzing massive data sets. Note that these polysemous words and phrases are different from homonymous words that have multiple unrelated meanings. As a consequence, the tags labeled by one user may be inappropriate to another user. When searching for documents with shared tags, users might find undesired documents or miss relevant pages.

Our taming technique can be applied to cope with this difficulty. We prepare a set of Web pages that are consistently labeled based on the target concept in which we are interested. These labeled pages are treated as tame data. Because making tame data is labor intensive, the size of the tame data set tends to be small. For the task whose labeled training samples are fully available, statistical techniques often fail to find the regularity in the tame data. In contrast, it is relatively easy to collect many tagged pages by using collaborative tagging services, but these tags may not consistent with the target concept as described above; thus, we treat these pages as wild data. These tame and wild data are mutually complementary; the tame data are well maintained, while the size of the wild data set is large. Therefore, it would be fruitful to use both data sets together. Our taming technique lets the user know whether any collaborative tags are consistent with his/her own concept. Further, it is also useful for finding more appropriate tags for new Web pages. In this paper, we will show a method for this taming, which we call BaggTaming. This is a modified version of bagging [2] so as to be applicable to taming. We employ BaggTaming in a tag-prediction task, and show the effectiveness of our method. We expect that this taming technique would be generally helpful for any situation where many unreliable observations are available together with a small amount of highly reliable information.

In section 2, we state a taming learning problem and describe our BaggTaming method. Section 3 show the experimental results for synthetic and real tagging data, respectively. Finally, after discussing the related work in section 4, we conclude in section 5.

2 Taming Task and Its Solution

In this section, we address the learning problem of taming and describe our BaggTaming algorithm, which is developed by modifying bagging so as to be applicable for taming.

2.1 What's Taming

We first address the learning problem of taming. Here, we are focused on classification, though taming techniques can be applied to other types of supervised learning tasks, such as regression. An object is represented by a feature vector, $\mathbf{x}$. The variable c denotes the class to which the object should be classified. The pair of a class and object, $(c_i, \mathbf{x}_i)$, is a training example. The goal of classification is to acquire a classifier that can predict an appropriate class for any input feature vector, $\mathbf{x}$, from a set of training examples.

A standard classifier is learned from only one homogeneous set of training examples. These training examples are assumed to be independently sampled from an identical distribution, $\mathrm{P}[c, \mathbf{x}]$, which expresses the target concept to be learned. On the other hand, in our taming case, two types of training examples, tame and wild, are adopted. Similar to a standard classification case, tame data are assumed to be independently sampled from an identical distribution, $\mathrm{P}[c, \mathbf{x}]$, that corresponds to the target concept. This set of tame data is denoted by $\mathcal{D}_T = \{(c_i, \mathbf{x}_i)\}_{i=1}^{N_T}$, where $N_T = |\mathcal{D}_T|$. A wild data set might include examples that are sampled from distributions that express irrelevant concepts, together with examples that are consistent with the target concept. We further assume that, in the wild data, the number of examples of the target concept is at least a few times N_T and that the learner does not know which examples are generated from the target distribution. This wild data set is denoted by $\mathcal{D}_W = \{(c_i, \mathbf{x}_i)\}_{i=1}^{N_W}$, where $N_W = |\mathcal{D}_W|$. Finally, we assume the size of the wild data set is much larger than that of the tame set, i.e., $N_W \gg N_T$. The goal of the learning problem of taming is to acquire a classifier that can more accurately predict classes by exploiting the information in wild data.

2.2 BaggTaming

To solve the above problem of taming, we developed the algorithm BaggTaming (Bootstrap AGGregated TAMING). Before showing our BaggTaming, we briefly describe the original bagging method, because this algorithm is a modified version of this bagging [2]. In the bagging algorithm, the following steps are iterated for $t = 1, \ldots, T$.

1. Obtain a training example set $\mathcal{D}_t$ by bootstrap sampling, that is, sampling with replacement, from a given training example set $\mathcal{D}$.
2. From this training set $\mathcal{D}_t$, learn a weak classifier $\hat{f}_t(\mathbf{x})$ by using a classification algorithm, such as naive Bayes.

By this procedure, T weak classifiers, $\hat{f}_1(\mathbf{x}), \ldots, \hat{f}_T(\mathbf{x})$, can be acquired. These weak classifiers are then aggregated, and any input feature vectors are finally classified. In the case of classification, this aggregation is done by majority voting. Formally, the class to which the feature vector $\mathbf{x}$ should belong is determined by

$$\hat{c} = \arg\max_{c \in \mathcal{C}} \sum_{t=1}^{T} \mathrm{I}[c{=}\hat{f}_t(\mathbf{x})], \tag{1}$$

where $\mathrm{I}[cond]$ is an indicator function, which takes 1 if the condition $cond$ is true; otherwise it takes 0, and $\mathcal{C}$ denotes the domain of classes.

The reason why this bagging improves the prediction accuracy is explained simply based on the bias-variance theory [2, 3]. According to the bias-variance theory, a generalization error is divided into three factors: the bias, which is derived from the choice of models, the variance, which is derived from the variation in sampling of training examples, and the intrinsic error, which cannot be avoided. If a low-bias model, which can approximate various forms of functions, is used for learning, the effect of the bias factor can be lessened while that of the variance factor more affects the generalization error. Inversely, in the case of a high-bias model, the degrees of effect caused by the bias and variance factors are exchanged. In the process of bagging, various training example sets are generated, weak classifiers are learned from each example set, and these classifiers are aggregated. Because this aggregated classifier has been trained by using various training sets, the variance factor can be lessened without increasing the error caused by the bias factor. Therefore, if a low-bias model is adopted, the bagging technique contributes to reducing the generalization error. However, if a high-bias model, such as Fisher's discriminant analysis, is used, bagging fails to reduce the prediction error, because the ratio of the error caused by the variance factor is originally small.

We developed our BaggTaming algorithm by modifying this bagging. In [3, section 7], an improvement technique for bagging is reported. In the case of standard bagging, training examples are bootstrap-sampled from a given example set, and the sampled examples are fed to a weak learner without modification. Instead, by adding a noise to the feature vectors in the sampled examples, the prediction accuracy can be improved. This is because more varied weak classifiers can be learned by this modification. Our BaggTaming algorithm is inspired by this improvement technique. In the process of BaggTaming, training examples are sampled from wild data instead from tame data. We expected that the variance part of the error can be more drastically reduced, because a wild data set contains more diverse examples than a tame set. Note that the idea of reducing variance by using the relevant data sets was also proposed in [4]. However, there is one difficulty. A wild data set contains examples of irrelevant concepts, which we want to ignore, and there is no information whether each example is of the target concept or not. To avoid this difficulty, we exploit a tame data, which is consistent with the target concept. Specifically, learning of our BaggTaming iterates the following two steps:

- Training examples are randomly sampled from the wild data set, and a weak classifier is learned from these training examples.
- The empirical accuracy of the weak classifier on the tame data is computed. If the accuracy is sufficiently high, the classifier is accepted; otherwise, it is rejected, and a new classifier is repeatedly learned.

Note that we assume that if most of examples that have been used for training the weak classifier are consistent with the target concept, the empirical accuracy of the classifier would be high. By iterating these procedures, the learner can obtain weak classifiers that are consistent with the target concept and that have wide diversity. In this way, T weak classifiers are acquired, and the final class is determined by majority voting.

```
 1:  $t = 1$
 2:  while $t \leq T$ do
 3:      $s = 1$
 4:      repeat
 5:          generate a training set $\mathcal{D}_t$ by sampling with replacement from $\mathcal{D}_W$
 6:          learn a weak classifier $\hat{f}_t(\mathbf{x})$ from training set $\mathcal{D}_t$
 7:          calculate the accuracy $p_t$ of the classifier $\hat{f}_t(\mathbf{x})$ on the tame set n$\mathcal{D}_T$
 8:          if $p \geq \mathrm{AcceptLimit}$ then $t = t + 1$, goto step 2
 9:          if $s \geq \mathrm{FailureLimit}$ then
                 let the most accurate classifier in this inner loop be $\hat{f}_t(\mathbf{x})$
                 $t = t + 1$, goto step 2
10:          $s = s + 1$, goto step 4
11:  output the weak classifiers $\hat{f}_1(\mathbf{x}), \ldots, \hat{f}_T(\mathbf{x})$,
     together with their corresponding accuracies $p_1, \ldots, p_T$.
```

Fig. 1. BaggTaming algorithm

Our BaggTaming algorithm is shown in Figure 1. Two types of training sets, wild and tame data, and a weak learner are given as inputs for this algorithm. T weak classifiers are acquired in the outer loop beginning from step 2. The counter s for the rejection of weak classifiers is initialized at step 3, and the inner loop starts at step 4. In steps 4 to 6, training examples are sampled from wild data, $\mathcal{D}_W$, a weak classifier, $\hat{f}_t(\mathbf{x})$, is acquired, and its accuracy, p_t, on the tame set $\mathcal{D}_T$ is calculated. At step 7, if the accuracy is greater than or equal to the threshold, AcceptLimit, the weak classifier is accepted, and then the algorithm turns to the next iteration of the outer loop. In the next step, if the number of rejections exceeds the threshold, FailureLimit, the most accurate weak classifier is set to $\hat{f}_t(\mathbf{x})$ as a backup, and then the next iteration of the outer loop begins. At step 9, after incrementing the rejection counter s, the acquisition of a weak classifier is re-tried. Finally, the algorithm outputs the weak classifiers $\hat{f}_1(\mathbf{x}), \ldots, \hat{f}_T(\mathbf{x})$, together with their corresponding accuracies, $p_1, \ldots, p_T$.

We describe an option for our BaggTaming algorithm. As the threshold, AcceptLimit, we adopted three options: three classifiers are leaned from all wild data, all tame data, and merged data, respectively. The accuracies of these classifiers on the tame set are calculated, and these accuracies are used as thresholds. Generally, the frequency of accepted classifiers rises in the order of tame, merged, and wild options.

We describe a few comments and options for our BaggTaming algorithm. If the ratio of examples consistent with the target concept to the total number of wild data is low, weak classifiers are too frequently rejected, and the algorithm slows. In such a condition, a weak classifier is learned from a union set of $\mathcal{D}_t$ and $\mathcal{D}_T$ at step 6. By adopting this option, weak classifiers are accepted more frequently, but the effect of reducing the variance in the prediction error is lessened.

The classification procedure of our BaggTaming is similar to that of standard bagging. Majority voting is used to predict the classes for a new vector, but each vote is weighted. As described before, the accuracy, p_t, would be high if the $\mathcal{D}_t$ contains many examples that are consistent with the target concept. Therefore, we weigh each weak classifier by p_t. Specifically, the class to which the feature vector $\mathbf{x}$ should belong is determined by

$$\hat{c} = \arg\max_{c \in \mathcal{C}} \sum_{t=1}^{T} p_t \mathrm{I}[c = \hat{f}_t(\mathbf{x})]. \tag{2}$$

2.3 Application of our BaggTaming Technique to a Tag Prediction Task

We then apply the above BaggTaming technique to the task to predict personalized tags in a social bookmark system. As described in the introduction, different users may assign different tags to the same Web page. For example, one may choose “python”, but another may do “programming”, according to his/her preference in the specificity level. Further, the synonymous words or the singular/plural forms give rise to another type of inconsistency. However, if we give attention to the specific target user, the choice of tags would be highly consistent, because any users would behave according to their own preference pattern, which would be highly self-consistent. We here perform a tag prediction task that is personalized to the target user as follows. First, for each candidate tag, we acquire a binary classifier to discriminate whether the target user will assign the candidate tag to Web pages. To be personalized this discrimination, this classifier is trained from the Web pages that are tagged by the target user, because such tags are considered to be reflecting the target user’s concept. Once such classifiers are learned, these can be used for judging whether each candidate tag should be assigned to a new Web page. For example, consider the case that the target user prefers the tag “python” to “programming” in terms of a given Web page. The classifiers for the tag “python” and “programming” would return positive and negative outputs, respectively, and tags that are appropriate for the target user can be selected.

However, we encounter a serious problem, which is often referred as a *cold-start problem* in a recommendation context [5]. In order that Web pages are tagged based on the concept of the target user, the binary classifiers are learned from the set of Web pages that have been tagged by the user before. The number of such Web pages are generally small, because it is difficult for one user to tag thousands of pages. In this case, the learned classifiers cannot make precise prediction, due to the insufficiency of training examples.

To cope with this difficulty, we adopt our BaggTaming technique. The Web pages that tagged by the target user are highly consistent with the user’s concept, while the number of pages are generally small. Additionally, we can exploit the enormous Web pages that have been tagged by the non-target users. These Web pages cannot be directly used for training classifiers, because the most of these are inconsistent with the target user’s concept. However, We can apply our BaggTaming by treating the target user’s and the non-target users' tagging information as tame and wild data, respectively. We expect that our BaggTaming enables to learn the more precise classifier, because these wild data partially contain useful information for learning the target user's concept. In particular, some users would surely select the same level of specificity as that of the target user, and some users would definitely use synonymous words in the same sense as the target user. In the next experimental section, we will show the more precise personalized classifiers can be learned by employing our BaggTaming in conjunction with both the target user's and non-target users' tagging data.

3 Experiments for Collaborative Tagging Data

In this section, we applied our BaggTaming to collaborative tagging data.

3.1 Data Sets and Experimental Settings

We first describe our collaborative tagging (CT) data set. We crawled the collaborative tagging site, del.icio.us, in July, 2007. The number of unique URLs, tags, and users were $762,454$, $172,817$, and $6,488$, respectively. We found $3,198,185$ bookmarks, which were the pairs of one registered URL and one tag assigned to the URL.

We counted the number of URLs to which each tag was assigned, and selected the 20 most-assigned tags. We focused on one of these tags, and call it a target tag. In this experiment, we dealt with a binary classification to predict whether the target tag would be assigned to a given URL or not, as described in section 2.3. For each target tag, we focused on the top user, who was the user who assigned the target tag to the most URLs among all users. We treated the URLs tagged by the top user as the tame data. We hereafter refer to this top user as a tame user. Among all URLs tagged by the tame user, the URLs to which the target tag was assigned were used as positive examples, and the rest of the URLs were used as negative ones. As the wild data set, we used the URLs tagged by the second to twentieth top users of the target tag. We refer to these nineteen users as wild users. The most of tags assigned by these wild users would be inconsistent, but a part of them might share the tame user's concept. Thus, the URLs tagged by these wild users were treated as wild data.

We next turned to the features that describe the URLs. As features, we adopted the most popular 100 tags except for the target tag. That is to say, the i-th element of the feature was the number of users who assigned the i-th most popular tag to the URL. We then abandoned the URLs to which none of the top 100 tags was assigned. Note that the Web pages' texts is frequently used as features, but we didn't employ them to avoid steps for cleaning irrelevant information, such as advertise messages. For each of the twenty target tags, we generated data sets. The sizes of tame and wild data are summarized in Table 1.

3.2 Results for Collaborative Tagging Data

We first compared our BaggTaming with a baseline method, which is standard bagging whose weak classifiers were trained by using the tame data. For both methods, we

Table 1. The sizes of collaborative tagging data sets

target tag	tame	wild	target tag	tame	wild
blog	2908	28214	web2.0	1784	13829
design	2511	26791	politics	1234	13709
reference	2355	22847	news	2473	13429
software	2658	22529	howto	1685	13407
music	2898	19725	imported	405	12862
programming	1697	18668	linux	1535	12231
web	2296	18503	blogs	1465	12217
tools	2365	18488	tutorial	1883	12001
video	2538	16734	games	2097	11291
art	2054	16521	free	1960	11258

NOTE: The “target tag” columns show the words used as target tags. The “tame” and “wild” columns show the sizes of the corresponding tame and wild data sets.

adopted again a naive Bayes learner with multinominal model [6] as weak classifiers, because this learner is computationally fast. The number of weak classifiers, T, was 30. The size of the sampled data was one-half of the tame data, i.e., $|\mathcal{D}_t| = N_T/2$. The FailureLimit threshold was set to 100, and the AcceptLimit threshold was the accuracy of the classifier learned from the merged data. We performed a 5-fold cross-validation test. Tame example set was divided into five blocks. One block was respectively picked, and the examples in these blocks were used for testing. The remaining four tame blocks and all the wild examples were used as the tame and wild training data, respectively.

The accuracies on the tame data set are shown in Table 2. We changed the tame data set without changing the wild data. For all ALL – 1/16 cases, our BaggTaming was clearly superior to bagging for the tame data. This result leads to the conclusion that our BaggTaming method can select the target information in the wild data and can use the information for learning classifiers. Further, the effectiveness of BaggTaming became more significant as the size of the tame sets decreased. This fact enhances the usefulness of BaggTaming, because a taming technique is more useful under a cold-start condition where less tame data are available. We also tested bagging classifiers whose weak classifiers were trained by using the wild set or merged data sets. We observed that almost all of these classifiers were inferior than bagging with weak classifiers trained from the tame set. Especially, for the tags, “web” or “web2.0”, the classifiers learned from the wild data are fairly worse. This fact indicates that these wild data sets don’t contain useful information for solving the target classification problem; thus, for these tags, we considered that our BaggTaming is inferior to the baseline as observed in Table 2.

We finally examined the effects of the parameters and options of BaggTaming. We applied BaggTaming under different settings to the tagging data with the ALL type tame set, and three statistics were calculated for each setting as shown in Table 3. As described in section 2.2, we prepared three options for the AcceptLimit, which is the threshold used for checking whether a candidate weak classifier is fully accurate or not. The columns “tame”, “merged”, and “wild” show results derived by BaggTaming with the tame, merged, and wild options. Adopting the tame option produces a set of highly

Table 2. Prediction Accuracies on The Tame Data Set

tag name	size of tame data sets									
	ALL		1/2		1/4		1/8		1/16	
	BT	bagg	BT	bagg	BT	bagg	BT	bagg	BT	bagg
blog	0.700	0.755	0.697	0.701	0.665	0.658	0.663	0.697	**0.713**	0.637
design	0.744	0.723	0.738	0.725	0.737	0.728	0.738	0.718	0.746	0.729
reference	0.825	0.813	**0.833**	0.808	**0.836**	0.812	**0.835**	0.816	**0.849**	0.815
software	0.752	0.771	0.764	0.767	0.784	0.775	**0.800**	0.753	**0.801**	0.775
music	0.956	0.953	**0.964**	0.954	**0.966**	0.956	**0.967**	0.958	**0.966**	0.954
programming	0.896	0.900	0.895	0.895	0.893	0.889	0.888	0.878	0.881	0.878
web	0.661	**0.775**	0.657	**0.763**	0.660	**0.760**	0.651	**0.731**	0.645	**0.702**
tools	0.772	0.749	**0.769**	0.728	**0.768**	0.700	**0.760**	0.667	**0.741**	0.681
video	0.884	0.887	0.892	0.891	0.891	0.897	0.896	0.901	0.883	0.901
art	**0.920**	0.899	**0.924**	0.895	**0.930**	0.898	**0.932**	0.908	**0.928**	0.915
web2.0	0.614	**0.708**	0.614	**0.696**	0.630	**0.706**	0.627	**0.716**	0.709	0.706
politics	0.649	0.666	0.633	**0.658**	0.625	0.645	0.628	0.638	0.622	0.631
news	**0.940**	0.806	**0.925**	0.687	**0.791**	0.522	0.657	0.478	**0.910**	0.403
howto	0.901	0.902	0.908	0.901	**0.917**	0.893	**0.920**	0.893	**0.926**	0.877
imported	1.000	0.971	1.000	0.971	0.994	0.965	**1.000**	0.907	0.936	0.930
linux	**0.783**	0.740	0.794	0.769	0.793	0.775	0.798	0.796	0.778	0.749
blogs	**0.937**	0.923	**0.941**	0.923	0.940	0.928	0.938	0.931	0.938	0.925
tutorial	**0.908**	0.889	**0.914**	0.884	**0.918**	0.880	**0.922**	0.877	**0.920**	0.889
games	0.961	0.963	0.964	0.962	0.964	0.960	**0.965**	0.953	**0.963**	0.945
free	0.810	0.802	**0.830**	0.783	**0.835**	0.801	**0.840**	0.753	**0.852**	0.756
win / lose	5 / 2		8 / 3		8 / 2		10 / 2		11 / 1	

NOTE: The column “tag name” shows the strings of the target tags. The column pair “ALL” shows the results when entire tame data were used, and the column pairs labeled “1/2” – ”1/16” show the results when the tame data were reduced to the 1/2 – 1/16 of the ALL case, respectively. The left “BT” and right “bagg” columns of each pair show the results derived by our BaggTaming and baseline bagging, respectively. Each row shows the accuracies for the corresponding target tag. Bold face indicates that the accuracy was larger than that derived by the other method, and the difference was statistically significant at the significance level of 1% using Z-test statistic. The last row “win/lose” shows the number of target tags for which our method won/lost baseline bagging.

Table 3. Statistics when AcceptLimit and the number of samples were changed

# samples	tame	merged	wild
100%	[0.477 52.55 3/3]	[0.060 15.33 3/3]	[0.004 4.82 4/4]
50%	[0.437 49.68 5/4]	[0.076 17.12 5/2]	[0.009 6.89 5/4]
20%	[0.404 46.40 8/1]	[0.124 24.00 8/2]	[0.032 13.32 7/3]

NOTE: The columns "tame", "merged", and "wild" show results when the tame, merged, and wild options for the AcceptLimit threshold (Section 2.2) were adopted. The "100%" – "20%"rows show the results when the training set size, $|\mathcal{D}_t|$, was set to the 100%, 50%, and 20% of N_T, respectively. Each entry contains three statistics. The left one is the failure ratio, which is the ratio of the failed weak learning in step 9 of BaggTaming to the total number of trials. The middle one is the mean number of trials until the resultant weak classifier is accepted. The right one is the number of wins and loses as shown in Table 2.

qualified weak classifiers, but makes the algorithm slow due to frequent rejections. On the other side, using the wild option accelerates BaggTaming, but weak classifiers can be poorer. The "100%," "50%," and "20%" rows show the results when the training set size, $|\mathcal{D}_t|$, was set to the 100%, 50%, and 20% of the size of the tame set, N_T, respectively. Each entry contains three statistics. The left one is the *failure ratio*, which is the ratio of the failed weak learning in step 9 of the BaggTaming algorithm to the total number of trials. The middle one is *mean trial*, which is the mean number of trials until the resultant weak classifier is accepted. That is to say, the mean number of iterations of the BaggTaming inner loop. The right one is the *win/lose*, which is number of wins and loses as shown in Table 2.

We first discuss the AcceptLimit option. In the case of the tame option, the failure ratio and mean trial are so large that the algorithm get slow. Inversely, inaccurate classifiers are more frequently selected under the wild option; thus, we employed the merged option. We next discuss the sample size, $|\mathcal{D}_t|$. If the number of training examples, $|\mathcal{D}_t|$, grows, the weak learner gets the more chance to acquire better classifiers. However, at the same time, this increases the probability that the non-target wild data are involved in the training set. This size should be determined according as how many samples in the wild data are tagged based on the target concept, but this number is unknown. By considering the balance of the efficiency and effectiveness, we set $|\mathcal{D}_t|$ to the 50% of N_T. We empirically confirmed that BaggTaming worked well for the wide ranges of data sets under this setting. Finally, we also changed the number of weak classifiers, T, from 10 to 100. We observed that the accuracies rises until $T = 30$, but further improvement was not observed. We expected that the reason for this is as follows. As described in section 2.2, bagging cannot reduce the bias factor of the total generalization error. We used naive Bayes, which is rather high-bias, as a weak learner. Therefore, we expected that the error derived by the variance factor would be almost diminished at $T = 30$. In future, we want to test lower-bias weak learners.

In summary, our BaggTaming successfully acquired the more accurate classifier than the classifier that were learned solely from the tame data. It can be concluded that BaggTaming succeed to learn a better classifier by exploiting useful information involved in the wild data.

4 Related Work

There are several machine learning schemes that adopt two types of training data sets. Semi-supervised learning [7] employs both labeled and unlabeled data sets. Because both tame and wild data are labeled, our taming approach clearly differs from semi-supervised learning.

Our taming problem can be considered as a sort of inductive transfers [8–15, 4]. Inductive transfer refers to the problem of retaining and applying the knowledge learned in one or more tasks to efficiently develop an effective hypothesis for a new task[2]. There are many variant tasks in this inductive transfer, such as multitask learning [8] or domain adaptation [11]. However, we may say that there is no strict consensus about the relations or definitions of these tasks in this research community. We consider that these problems are different in their motivation. Multitask learning [8] aims to simultaneously solve multiple tasks. For this aim, the learner have to find the common knowledge shared by these tasks in a form, such as hyper prior or similarity measure. In the case of domain adaptation [11], the tasks other than the target task don't need to be solved. Thus, it is important to find out useful information for solving the target task from the data of the non-target domains. According to this interpretation, we can say that our taming is more relevant to a domain adaptation task. However, constituents of training data sets for non-target tasks are different between taming and domain adaptation. In a case of domain adaptation, a non-target training data is labeled based on some single concept that is related to the target concept. Therefore, it is possible to fit a non-target model to such training data, as in [11]. On the other hand, in a case of taming, because the wild training set includes the examples of many kinds of tasks, it is difficult to fit a simple single model to them due to their diversity. We therefore designed our BaggTaming so as to ignore non-target data in the wild set. Indeed, we tried an approach fitting mixture models for solving the tag prediction task, but no improvement was observed.

Dai et al. recently proposed the problem more related to taming [9]. This problem is treated as inductive transfer, but is more close to domain adaptation according to the above interpretation. They prepared two types of data sets. One is highly similar to the target concept, while the other is less similar. Again, what is the similarity is not so formally defined as above problems, the relation to taming is not yet clear. We plan to investigate the relation between their method and ours.

In the covariate shift [16] approach, the data distributions of test and training data sets are assumed to differ, while the criteria of labeling are the same for both data sets. This problem differs from our taming because in the case of covariate shift one homogeneous training set is assumed.

We finally discuss tagging on Web pages or blog pages. The P-TAG [17] is the system that predicts appropriate tags for given Web pages. Because this prediction is based on the users' personal repository, and other users' tags are not referred to, there is no need for considering inconsistency among tagging criteria. The Autotag [18] is designed to predict proper tags for blog articles. Though other users' tags are used, inconsistency in tags is not taken into account.

[2] From the announcement of the "NIPS 2005 Workshop, Inductive Transfer: 10 Years Later"

5 Conclusion

We proposed a new machine problem, which we call *taming*. This exploits two types of data sets, tame and wild. The tame data are labeled based on a consistent concept, while the size of the data set is relatively small. The labels of the wild data can be inconsistent, but are more easily collected. We proposed a method, BaggTaming, for solving this problem. We applied BaggTaming to synthetic and real collaborative tagging data sets, and showed that labels are more accurately predicted by our method.

We plan to enhance the theoretical background, to improve the efficiency by using adaptive sampling, and to acquire more accurate classifier by using other probabilistic models.

Acknowledgments: We wish to thank Dr. Yutaka Matuo and Dr. Atsushi Fujii for their valuable advices. Thanks are due to Hottolink Inc. for assistance in crawling.

References

1. Golder, S.A., Huberman, B.A.: Usage patterns of collaborative tagging systems. J. of Information Science **32**(2) (2006) 198–208
2. Breiman, L.: Bagging predictors. Machine Learning **24** (1996) 123–140
3. Breiman, L.: Arcing classifiers. The Annals of Statistics **26**(3) (1998) 801–849
4. Wu, P., Dietterich, T.G.: Improving SVM accuracy by training on auxiliary data sources. In: Proc. of The 21st Int'l Conf. on Machine Learning. (2004) 871–878
5. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Trans. on Information Systems **22**(1) (2004) 5–53
6. McCallum, A., Nigam, K.: A comparison of event model for naive bayes text classification. In: AAAI-98 Workshop on Learning for Text Categorization. (1998) 41–48
7. Chapelle, O., Schölkopf, B., Zien, A., eds.: Semi-supervised Learning. MIT Press (2006)
8. Caruana, R.: Multitask learning. Machine Learning **28** (1997) 41–75
9. Dai, W., Yang, Q., Xue, G.R., Yu, Y.: Boosting for transfer learning. In: Proc. of The 24th Int'l Conf. on Machine Learning. (2007) 193–200
10. Daumé III, H.: Frustratingly easy domain adaptation. In: Proc. of the 45th Annual Meeting of the Association of Computational Linguistics. (2007) 256–263
11. Daumé III, H., Marcu, D.: Domain adaptation for statistical classifiers. Journal of Artificial Intelligence Research **26** (2006) 101–126
12. Do, C.B., Ng, A.Y.: Transfer learning for text classification. In: Advances in Neural Information Processing Systems 18. (2006) 299–306
13. Liao, X., Xue, Y., Carin, L.: Logistic regression with an auxiliary data streams. In: Proc. of The 22nd Int'l Conf. on Machine Learning. (2005) 505–512
14. Raina, R., Ng, A.Y., Koller, D.: Constructing informative priors using transfer learning. In: Proc. of The 23rd Int'l Conf. on Machine Learning. (2006) 713–720
15. Thrun, S.: Is learning the n-th thing any easier than learning the first? In: Advances in Neural Information Processing Systems 8. (1996) 640–646
16. Shimodaira, H.: Improving predictive inference under covariate shift by weighting the log-likelihood function. J. of Statistical Planning and Inference **90** (2000) 227–244
17. Chirita, P.A., Costache, S., Handschuh, S., Nejdl, W.: P-TAG: Large scale automatic generation of personalized annotation TAGs for the Web. In: Proc. of The 16th Int'l Conf. on World Wide Web. (2007) 845–854
18. Mishne, G.: AutoTag: A collaborative approach to automated tag assignment for Weblog posts. In: Proc. of The 15th Int'l Conf. on World Wide Web. (2006) 953–954

Identifying Ideological Perspectives of Web Videos using Patterns Emerging from Folksonomies

Wei-Hao Lin and Alexander Hauptmann*

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213 USA
{whlin,alex}@cs.cmu.edu

Abstract. We are developing a classifier that can automatically identify a web video's ideological perspective on a political or social issue (e.g., pro-life or pro-choice on the abortion issue). The problem has received little attention, possibly due to inherent difficulties in content-based approaches. We propose to develop such a classifier based on the pattern of tags emerging from folksonomies. The experimental results are positive and encouraging.

1 Introduction

Video sharing websites such as YouTube, Metacafe, and Imeem have been extremely popular among Internet users. More than three quarters of Internet users in the United States have watched video online. In a single month in 2008, 78.5 million Internet users watch 3.25 billion videos on YouTube. On average, YouTube viewers spend more than one hundred minutes a month watching videos on YouTube [1].

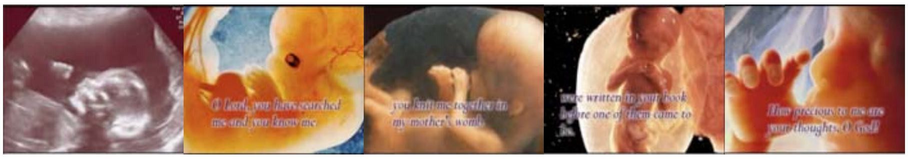
Video sharing websites have also become an important platform for expressing and communicating different views on various social and political issues. In 2008, CNN and YouTube held United States presidential debates in which presidential candidates answered questions that were asked and uploaded by YouTube users. In March 2008 YouTube launched YouChoose'08 [1] in which each presidential candidate has their own channel. The accumulative viewership for one presidential candidate as of June 2008 has exceeded 50 millions [2]. In addition to politics, many users have authored and uploaded videos expressing their views on social issues. For example, Figure 1 is an example of a "pro-life" web video on the abortion issue[2], while Figure 2 is an example of "pro-choice" web video[3].

* We would like to thank the anonymous reviewers for their valuable comments and suggestions. This work was supported in part by the National Science Foundation (NSF) under Grants No. IIS-0535056 and CNS-0751185.

[1] http://youtube.com/youchoose

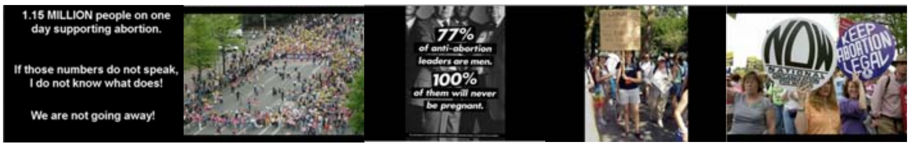
[2] http://www.youtube.com/watch?v=TddCILTWNr8

[3] http://www.youtube.com/watch?v=oWeX0jsv58c



O Lord, you have searched me and you know me
you knit me together in my mother's womb
were written in your book before one of them came to be.
How precious to me are your thoughts, O God!

Fig. 1. The key frames of a web video expressing a "pro-life" view on the abortion issue, which is tagged with `prayer`, `pro-life`, and `God`.



1.15 MILLION people on one day supporting abortion.
If those numbers do not speak, I do not know what does!
We are not going away!
77% of anti-abortion leaders are men.
100% of them will never be pregnant.
NOW
KEEP ABORTION LEGAL

Fig. 2. The key frames of a web video expressing a "pro-choice" view on the abortion issue, which is tagged with `pro`, `choice`, `feminism`, `abortion`, `women`, `rights`, `truth`, `Bush`.

We are developing a computer system that can automatically identify highly biased broadcast television news and web videos. Such a system may increase an audience's awareness of individual news broadcasters' or video authors' biases, and can encourage viewers to seek videos expressing contrasting viewpoints. Classifiers that can automatically identify a web video's ideological perspective will enable video sharing sites to organize videos on various social and political views according to their ideological perspectives, and allow users to subscribe to videos based on their personal views. Automatic perspective classifiers will also enable content control or web filtering software to filter out videos expressing extreme political, social or religious views that may not be suitable for children.

Although researchers have made great advances in automatically detecting "visual concepts" (e.g., car, outdoor, and people walking) [3], developing classifiers that can automatically identify whether a video is about *Catholic* or *abortion* is still a very long-term research goal. The difficulties inherent in content-based approaches may explain why the problem of automatically identifying a video's ideological perspective on an issue has received little attention.

- In this paper we propose to identify a web video's ideological perspective on political and social issues using associated tags. Videos on video sharing sites such as YouTube allow users to attach tags to categorize and organize videos. The practice of collaboratively organizing content by tags is called folksonomy, or collaborative tagging. In Section 3.3 we show that a unique pattern of tags emerges from videos expressing opinions on political and social issues.
- In Section 2 we apply a statistical model to capture the pattern of tags from a collection of web videos and associated tags. The statistical model

simultaneously captures two factors that account for the frequency of a tag associated with a web video: what is the subject matter of a web video? and what ideological perspective does the video's author take on an issue?

– We evaluate the idea of using associated tags to classify a web video's ideological perspective on an issue in Section 3. The experimental results in Section 3.2 are very encouraging, suggesting that Internet users holding similar ideological beliefs upload, share, and tag web videos similarly.

2 Joint Topic and Perspective Model

We apply a statistical model to capture how web videos expressing strongly a particular ideological perspective are tagged. The statistical model, called the Joint Topic and Perspective Model [4], is designed to capture an *emphatic* pattern empirically observed in many ideological texts (editorials, debate transcripts) and videos (broadcast news videos). We hypothesize that the tags associated with web videos on various political and social issues also follow the same emphatic pattern.

The emphatic pattern consists of two factors that govern the content of ideological discourse: *topical* and *ideological*. For example, in the videos on the abortion issue, tags such as `abortion` and `pregnancy` are expected to occur frequently no matter what ideological perspective a web video's author takes on the abortion issue. These tags are called *topical*, capturing what an issue is about. In contrast, the occurrences of tags such as `pro-life` and `pro-choice` vary much depend on a video author's view on the abortion issue. These tags are emphasized (i.e., tagged more frequently) on one side and de-emphasized (i.e., tagged less frequently) on the other side. These tags are called *ideological*.

The Joint Topic and Perspective Model assigns *topical* and *ideological* weights to each tag. The topical weight of a tag captures how frequently the tag is chosen because of an issue. The *ideological* weight of a tag represents to what degree the tag is emphasized by a video author's ideology on an issue. The Joint Topic and Perspective Model assumes that the observed frequency of a tag is governed by these two sets of weights combined.

We illustrate the main idea of the Joint Topic and Perspective Model in a three tag world in Figure 3. Any point in the three tag simplex represents the proportion of three tags (e.g., `abortion`, `life`, and `choice`) chosen in web videos about the abortion issue (also known as a multinomial distribution's parameter). T represents how likely we would be to see `abortion`, `life`, and `choice` in web videos about the abortion issue. Suppose a group of web video authors holding the "pro-life" perspective choose to produce and tag more `life` and fewer `choice`. The *ideological* weights associated with this "pro-life" group in effect move the proportion from T to V_1. When we sample tags from a multinomial distribution of a parameter at V_1, we would see more `life` and fewer `choice` tags. In contrast, suppose a group of web video authors holding the "pro-choice" perspective choose to make and tag more `choice` and fewer `life`. The *ideological* weights associated with this "pro-choice" group in effect move the proportion

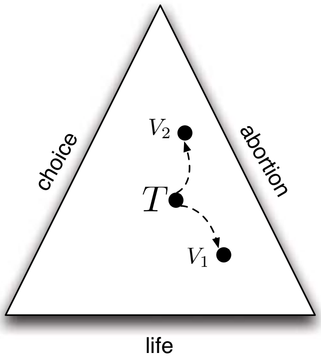


Fig. 3. A three tag simplex illustrates the main idea behind the Joint Topic and Perspective Model. T denotes the proportion of the three tags (i.e., topical weights) that are chosen for a particular issue (e.g., abortion). V_1 denotes the proportion of the three tags after the topical weights are modulated by video authors holding the "pro-life" view; V_2 denotes the proportion of the three tags modulated by video authors holding the contrasting "pro-choice" view.

from T to V_2. When we sample tags from a multinomial distribution of a parameter at V_2, we would see more `life` and fewer `choice` tags. The topical weights determine the position of T in a simplex, and each ideological perspective moves T to a biased position according to its ideological weights.

We can fit the Joint Topic and Perspective Model on data to simultaneously uncover topical and ideological weights. These weights succinctly summarize the emphatic patterns of tags associated with web videos about an issue. Moreover, we can apply the weights learned from training videos, and predict the ideological perspective of a new web video based on associated tags.

2.1 Model Specification and Predicting Ideological Perspectives

Formally, the Joint Topic and Perspective Model assumes the following generative process for the tags associated with web videos:

$$\begin{aligned} P_d &\sim \text{Bernoulli}(\pi), d = 1, \ldots, D \\ W_{d,n}|P_d = v &\sim \text{Multinomial}(\beta_v), n = 1, \ldots, N_d \\ \beta_v^w &= \frac{\exp(\tau^w \times \phi_v^w)}{\sum_{w'} \exp(\tau^{w'} \times \phi_v^{w'})}, v = 1, \ldots, V \\ \tau &\sim \text{N}(\mu_\tau, \Sigma_\tau) \\ \phi_v &\sim \text{N}(\mu_\phi, \Sigma_\phi). \end{aligned}$$

The ideological perspective P_d from which the d-th web video in a collection was produced (i.e., its author or uploader's ideological perspective) is assumed to be a Bernoulli variable with a parameter π. In this paper, we focus on bipolar ideological perspectives, that is, those political and social issues with only two perspectives of interest ($V = 2$). There are a total of D web videos in the collection. The n-th tag in the d-th web video $W_{d,n}$ is dependent on its author's ideological perspective P_d and assumed to be sampled from the multinomial distribution of a parameter β. There are a total of N_d tags associated with the d-th web video.

The tag multinomial's parameter, β_v^w, subscripted by an ideological perspective v and superscripted by the w-th tag in the vocabulary, consists of two parts: a topical weight τ^w and ideological weights $\{\phi_v^w\}$. Every tag is associated with one topical weight τ^w and two ideological weights ϕ_1^w and ϕ_2^w. β is an auxiliary variable, and is deterministically determined by (unobserved) topical and ideological weights.

τ represents the *topical* weights and is assumed to be sampled from a multivariate normal distribution of a mean vector μ_τ and a variance matrix Σ_τ. ϕ_v represents the *ideological* weights and is assumed to be sampled from a multivariate normal distribution of a mean vector μ_ϕ and a variance matrix Σ_τ. Every tag is associated with one topical weight τ^w and two ideological weights ϕ_1^w and ϕ_2^w. Topical weights are modulated by ideological weights through a multiplicative relationship, and all the weights are normalized through a logistic transformation. The graphical representation of the Joint Topic and Perspective Model is shown in Figure 4.

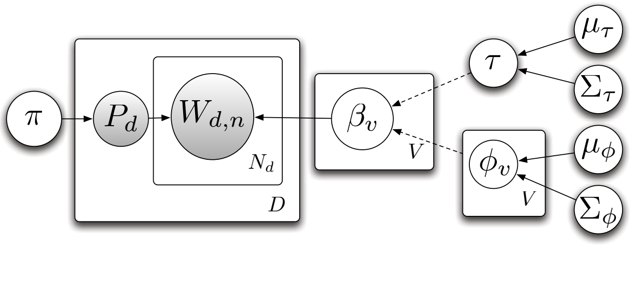


Fig. 4. A Joint Topic and Perspective model in a graphical model representation. A dashed line denotes a deterministic relation between parents and children nodes.

Given a set of D documents on a particular topic from differing ideological perspectives $\{P_d\}$, the joint posterior probability distribution of the topical and ideological weights under the Joint Topic and Perspective model is

$$
\begin{aligned}
&P(\tau, \{\phi_v\}|\{W_{d,n}\}, \{P_d\}; \Theta) \\
\propto &P(\tau|\mu_\tau, \Sigma_\tau) \prod_v P(\phi_v|\mu_\phi, \Sigma_\phi) \prod_{d=1}^{D} P(P_d|\pi) \prod_{n=1}^{N_d} P(W_{d,n}|P_d, \tau, \{\phi_v\}) \\
= &\mathrm{N}(\tau|\mu_\tau, \Sigma_\tau) \prod_v \mathrm{N}(\phi_v|\mu_\phi, \Sigma_\phi) \prod_d \mathrm{Bernoulli}(P_d|\pi) \prod_n \mathrm{Multinomial}(W_{d,n}|P_d, \beta),
\end{aligned}
$$

where $N(\cdot)$, Bernoulli$(\cdot)$ and Multinomial$(\cdot)$ are the probability density functions of multivariate normal, Bernoulli, and multinomial distributions, respectively.

The joint posterior probability distribution of τ and $\{\phi_v\}$, however, are computationally intractable because of the non-conjugacy of the logistic-normal prior. We have developed an approximate inference algorithm [4]. The approximate inference algorithm is based on variational methods, and parameters are estimated using variational Expectation Maximization [5].

To predict a web video's ideological perspective is to calculate the following conditional probability,

$$
\begin{aligned}
&P(\tilde{P}_d|\{P_d\}, \{W_{d,n}\}, \{\tilde{W}_n\}; \Theta) \\
= &\int \int P(\{\phi_v\}, \tau|\{P_d\}, \{W_{d,n}\}, \{\tilde{W}_n\}; \Theta) \\
&P(\tilde{P}_d|\{\tilde{W}_n\}, \tau, \{\phi_v\}; \Theta) d\tau d\phi_v \qquad (1)
\end{aligned}
$$

The predictive probability distribution in 1 is not computationally tractable, and we approximate it by plugging in the expected values of τ and $\{P_d\}$ obtained in variational inference.

3 Experiments

3.1 Data

We collected web videos expressing opinions on various political and social issues from YouTube[4]. To identify web videos expressing a particular ideological perspective on an issue, we selected "code words" for each ideological perspective, and submitted the code words as query to YouTube. All of the returned web videos are labeled as expressing the particular ideological perspective. For example, the query words for the "pro-life" perspective on the abortion issue are "pro-life" and "abortion."

We downloaded web videos and associated tags for 16 ideological views in May 2008 (two main ideological perspectives for eight issues), as listed in Table 1. Tags are keywords voluntarily added by authors or uploaders[5]. The total number of downloaded videos and associated tags are shown in Table 2. Note that the

[4] `http://www.youtube.com/`.

[5] `http://www.google.com/support/youtube/bin/answer.py?hl=en&answer=55769`

	Issue	View 1	View 2
1	Abortion	pro-life	pro-choice
2	Democratic party primary election in 2008	pro-Hillary	pro-Obama
3	Gay rights	pro-gay	anti-gay
4	Global warming	supporter	skeptic
5	Illegal immigrants to the United States	Legalization	Deportation
6	Iraq War	pro-war	anti-war
7	Israeli-Palestinian conflict	pro-Israeli	pro-Palestinian
8	United States politics	pro-Democratic	pro-Republican

Table 1. Eight political and social issues and their two main ideological perspectives

	total videos	total tags	vocabulary
1	2850	30525	4982
2	1063	13215	2315
3	1729	18301	4620
4	2408	27999	4949
5	2445	25820	4693
6	2145	25766	4634
7	1975	22794	4435
8	2849	34222	6999

Table 2. The total number of downloaded web videos, the total number of tags, and the vocabulary size (the number of unique tags) for each issue

number of downloaded videos is equal to less than the total number of videos returned by YouTube due of the limit on the maximum number of search results in YouTube APIs.

We assume that web videos containing the “code words” of an ideological perspective in tags or descriptions convey the particular view, but this assumption may not be true. YouTube and many web video search engines are so far not designed to retrieve videos expressing opinions on an issue, let along to retrieve videos expressing a particular ideological view using keywords. Moreover, a web video may mention the code words of an ideological perspective in titles, descriptions, or tags but without expressing any opinions on an issue. For example, a news clip tagged with “pro-choice” may simply report a group of pro-choice activists in a protest and do not express strongly a so-called pro-choice point of view on the abortion issue.

3.2 Identifying Videos’ Ideological Perspectives

We evaluated how well a web video’s ideological perspective can be identified based on associated tags in a classification task. For each issue, we trained a binary classifier based on the Joint Topic and Perspective model in Section 2, and applied the classifier on a held-out set. We reported the average accuracy of the 10-fold cross-validation. We compared the classification accuracy using the Joint Topic and Perspective Model with a baseline that randomly guesses one of two ideological perspectives. The accuracy of a random baseline is close but not necessarily equal to 50% because the number of videos in each ideological perspective on an issue are not necessarily equivalent.

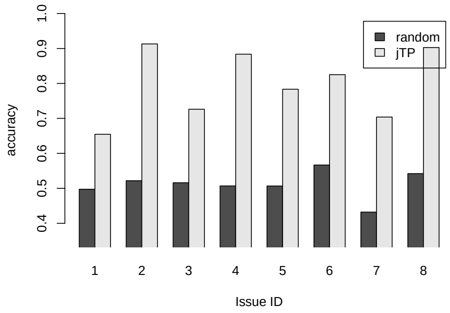


Fig. 5. The accuracies of classifying a web video's ideological perspective on eight issues

The experimental results in Figure 5 are very encouraging. The classifiers based on the Joint Topic and Perspective Model (labeled as jTP in Figure 5) outperform the random baselines for all eight political and social issues. The positive results suggest that the ideological perspectives of web videos can be identified using associated tags. Note that because the labels of our data are noisy, the results should be considered as a lower bound. The actual performance may be further improved if less noisy labels are available.

The positive classification results also suggest that Internet users sharing similar ideological beliefs on an issue appear to author, upload, and share similar videos, or at least, to tag similarly. Given that these web videos are uploaded and tagged at different times without coordination, it is surprising to see any pattern of tags emerging from folksonomies of web videos on political and social issues. Although the theory of ideology has argued that people sharing similar ideological beliefs use similar rhetorical devices for expressing their opinions in the mass media [6], we are the first to observe this pattern of tags in user-generated videos.

The non-trivial classification accuracy achieved by the Joint Topic and Perspectives Model suggests that the statistical model seem to closely match the real data. Although the Joint Topic and Perspective Model makes several modeling assumptions, including a strong assumption on the independence between tags (through a multinomial distribution), the high classification accuracy supports that these assumptions are not violated by the real data too much.

3.3 Patterns of Tags Emerging from Folksonomies

We illustrate the patterns of tags uncovered by the Joint Topic and Perspective Model in Figure 6 and Figure 7. We show only tags that occur more than 50 times in the collection. Recall that the Joint Topic and Perspective Model simultaneously learns the topical weights τ (how frequently a word is tagged in web videos on an issue) and ideological weights ϕ (how frequently a tag is emphasized by a particular ideological perspective). We summarize these weights and tags in a color text cloud, where a word's size is correlated with the tag's topical weight, and a word's color is correlated with the tag's ideological weight. Tags not particularly emphasized by either ideological perspectives are painted light gray.

The tags with large topical weights appear to represent the subject matter of an issue. The tags with large topical weights on the abortion issue in Figure 6 include `abortion`, `pro life`, and `pro choice`, which are the main topic and two main ideologies. The tags with large topical weights on the global warming issue in Figure 7 include `global warming`, `Al Gore` and `climate change`. Interestingly, tags with large topical weights are usually not particularly emphasized by either of the ideological views on an issue.

The tags with large ideological weights appear to closely represent each ideological perspective. Users holding the pro-life beliefs on the abortion issue (red in Figure 6) upload and tag more videos about `unborn baby` and religion (`Catholic`, `Jesus`, `Christian`, `God`). In contrast, users holding the pro-choice beliefs on the abortion issue (blue in Figure 6) upload more videos about women's rights (`women`, `rights`, `freedom`) and atheism (`atheist`). Users who acknowledge the crisis of global warming (red in Figure 7) uploads more videos about energy (`renewable energy`, `oil`, `alternative`), recycling (`recycle`, `sustainable`), and pollution (`pollution`, `coal`, `emissions`). In contrast, users skeptical about global warming upload more videos that criticize global warming (`hoax`, `scam`, `swindle`) and suspect it is a conspiracy (`NWO`, `New World Order`).

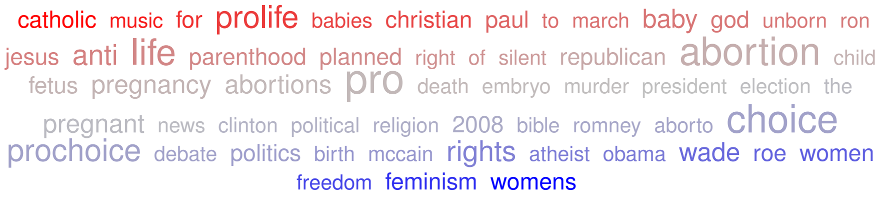


Fig. 6. The color text cloud summarizes the topical and ideological weights learned in the web videos expressing contrasting ideological perspectives on the abortion issue. The larger a word's size, the larger its topical weight. The darker a word's color shade, the more extreme its ideological weight. Red represents the pro-life ideology, and blue represents the pro-choice ideology. The words are ordered by ideological weights, from strongly pro-life (red) to strongly pro-choice (blue).

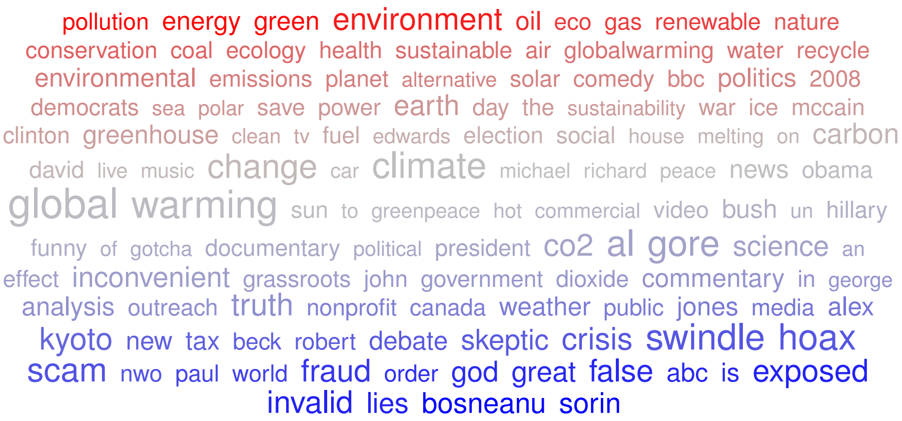


Fig. 7. The color text cloud summarizes the topical and ideological weights learned in the web videos expressing contrasting ideological perspectives on the global warming issue. The larger a word's size, the larger its topical weight. The darker a word's color shade, the more extreme its ideological weight. Red represents the ideology of global warming supporters, and blue represents the ideology of global warming skeptics. The words are ordered by ideological weights, from strongly supporting global warming (red) to strongly skeptical about global warming (blue).

We do not intend to give a full analysis of why each ideology chooses and emphasizes these tags, but to stress that folksonomies of the ideological videos on the Internet are a rich resource to be tapped. Our experimental results in Section 3.2 and the analysis in this section show that by learning patterns of tags associated with web videos, we can identify web videos' ideological perspectives on various political and social issues with high accuracy.

Folksonomies mined from video sharing sites such as YouTube contain up-to-date information that other resources may lack. Due to the data collection time coinciding with the United States presidential election, many videos are related to presidential candidates and their views on various issues. The names of presidential candidates occur often in tags, and their views on various social and political issues become discriminative features (e.g., `Ron Paul`'s pro-life position on the abortion issue in Figure 6). Ideological perspective classifiers should build on folksonomies of web videos to take advantage of these discriminative features. Classifiers built on static resources may fail to recognize these current, but very discriminative, tags.

4 Related Work

We borrow statistically modeling and inference techniques heavily from research on topic modeling (e.g., [7], [8] and [9]). They focus mostly on modeling text collections that containing many *different* (latent) topics (e.g., academic conference papers, news articles, etc). In contrast, we are interested in modeling ideology texts that are mostly on the *same* topic but mainly differs in their ideological perspectives. There have been studies going beyond topics (e.g., modeling authors [10]). In this paper we are interested in modeling lexical variation collectively from multiple authors sharing similar beliefs, not lexical variations due to individual authors' writing styles and topic preference.

5 Conclusion

We propose to identify the ideological perspective of a web video on an issue using associated tags. We show that the statistical patterns of tags emerging from folksonomies can be successfully learned by a Joint Topic and Perspective Model, and the ideological perspectives of web videos on various political and social issues can be automatically identified with high accuracy. Web search engines and many Web 2.0 applications can incorporate our method to organize and retrieve web videos based on their ideological perspectives on an issue.

References

1. comScore: YouTube.com accounted for 1 out of every 3 u.s. online videso viewed in january. `http://www.comscore.com/press/release.asp?press=2111Please` (March 2008)
2. techPresident: YouTube stats. `http://www.techpresident.com/youtube` (June 2008)
3. Naphade, M.R., Smith, J.R.: On the detection of semantic concepts at TRECVID. In: Proceedings of the Twelfth ACM International Conference on Multimedia. (2004)
4. Lin, W.H., Xing, E., Hauptmann, A.: A joint topic and perspective model for ideological discourse. In: Proceedings of the 2008 European Conference on Machine Learning and Principles (ECML) and Practice of Knowledge Discovery in Databases (PKDD). (2008)
5. Attias, H.: A variational bayesian framework for graphical models. In: Advances in Neural Information Processing Systems 12. (2000)
6. Van Dijk, T.A.: Ideology: A Multidisciplinary Approach. Sage Publications (1998)
7. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. (1999) 50–57
8. Blei, D.M., Ng, A.Y., Jordan, M.: Latent Dirichlet allocation. Journal of Machine Learning Research **3** (January 2003) 993–1022
9. Griffiths, T.L., Steyvers, M.: Finding scientific topics. Proceedings of the National Academy of Sciences **101** (2004) 5228–5235
10. Rosen-Zvi, M., Griffths, T., Steyvers, M., Smyth, P.: The author-topic model for authors and documents. In: Proceedings of the 20th Conference on Unvertainty in Artificial Intelligence. (2004)

Topical Structure Discovery in Folksonomies

Ilija Subasic and Bettina Berendt

Katholieke Universiteit Leuven, Dept. of Computer Science,
Celestijnenlaan 200A, B-3001 Heverlee, Belgium
{Ilija.Subasic, Bettina.Berendt}@cs.kuleuven.be

Abstract. In recent years social bookmarking systems (tagging systems) became one of the highly popular applications on the Internet. The main idea of social bookmarking is to organize content in a loose fashion by allowing users to completely freely annotate content. This work presents a way of combining the information retrieval (IR), Semantic Web and social web approaches of searching the Web by including general topic categories as a part of tagging systems. In this way semantic and social web are presented in a unified framework of search and indexing content. The work also shows a way of ontology learning by creating a hierarchical network of tag associations. This network is created using association rules discovery. In order to enhance these networks, IR search engine results are used to evaluate relevance of resources to a given topic. Networks of association, created by application of a modified Apriori algorithm, are evaluated with topic networks from the Open Directory Project (www.dmoz.org).

1 Introduction

Much of the recent growth of digital content is an effect of an increased number of internet users and their interest in on-line publishing. However, search for appropriate content remains one of the main problems. Early approaches to search that include information retrieval based search engines and web directories have almost completely been replaced by a second generation of search agents that beside the content itself take into account the interaction of users through, for example, PageRank or tagging. Social bookmarking tools/systems (SBSs) are one of the ways this second generation of agents tackle the search problem. SBSs enable users to express their reflection upon some content that is available on the web. The main idea of the social web through the use of social bookmarking is that a form of content organization emerges when users are allowed to completely freely annotate content. Databases of such systems consist of millions of users, tags and resources. A combination of these 3 objects (users, tags and resources) constitutes a folksonomy [20]. SBSs rely on the principle of collective intelligence. This principle, older than the web itself, is based on the idea that every decision made by a group of people with diverse expertise is better than the decision made by a single domain expert. Based on a different organization of SBSs they could be divided into self-tagging, which allow only content creator to tag, and non-self-tagging systems, which enable any user to tag any content [8]. In this paper's case study, we consider a non-self-tagging system. Nevertheless, the approach could be applied to self-tagging SBSs as well.

The problem faced by most SBSs is how to utilize the huge metadata repository created by users. Most applications just use tag clouds as a representation of their metadata. A tag cloud is a statistical overview of tagging behaviour of users. This way of summarizing folksonomy only offers users simple search based on the frequency of keywords (tags), without taking into consideration content or context of a resource. The analysis of tag distributions has shown that they follow some structure [16]. Nevertheless, most of the applications that enable users to tag provide little or no structured way of presenting tags to the users.

The need for structure discovery in such large repositories must involve some level of intelligent data analysis. This mix of user-generated (meta)data and intelligent ways of organizing and presenting it to users could be regarded as the next generation of web search. The main idea of this approach is that users' contributions are combined with some 'ground truth'. As ground truth, we consider content organization by predefined categories done by some agent (human expert or machine learning algorithm).

The key motivation of the present paper is to show a possible way of using data mining methods to benefit both from social and Semantic Web. The paper shows the use of data mining for discovering structure inside folksonomies by applying an algorithm for topic specific association rules discovery. Its goals are answers to three main questions:

1. Are resources tagged with some tag similar to those that one can find using IR-based search engine using the tag as query?

2. Can networks of tags and associations between them approximate existing topic hierarchies?

3. Does topic relevance inclusion in analysis enhance this approximation?

The first goal is thus to find underlying topics in folksonomies, and the second is to compare the topic structure. Building on this, the third goal investigates the inclusion of resource-to-topic relevance into structure discovery. The idea of topic relevance is that user input has to satisfy some conditions to be included into the structure discovery process. An algorithm for topical structure discovery in folksonomies is developed to meet this goal.

The paper continues by giving a related work overview in section 2 and then describes the topical structure discovery algorithm in section 3. Section 4 describes a case study and evaluation results, and the last section gives a brief conclusion of the presented work.

2 Related work

To be able to apply data mining methods to folksonomies, one must define them formally. We adopt a model described in [13], which models a folksonomy F by its representing tripartite hypergraph:

$$H(F) = < V, E > \qquad (1)$$

where *V* is a set of nodes that is the union of the sets *A*,*C*,*I*, which stand for users (identities), tags (concepts) and resources (items), and *E* is a set of edges denoting which user tagged a certain resource with a certain tag: $E = \{\{a,i,c\} \mid (a,i,c) \in F\}$.

2.1 Structure discovery in folksonomies

In [9], the authors examine the way tag clouds are created and evaluate resource overlap using three different measures which express the weight of the tag inside a tag cloud. The authors compared different weighting schemes using the Jaccard index to calculate the relative co-occurrence of tags in two different resource sets. Although very popular and used by most social bookmarking systems, tag clouds are just ways of summarizing basic statistical properties of a tag set. Therefore, different data mining methods have been applied in order to describe folksonomy structure in a better way.

From the graph described in equation (1), the author [13] derives different graphs. For tag clustering, he extracts an IC (item-concept) graph, a bipartite graph of resources and tags, on which a graph clustering algorithm is applied. Alternative clustering approaches [4, 20] are modularity and EM clustering. The application of EM clustering showed that a small number of concepts (40 for del.icio.us) can be used to group tags. However, clustering just shows that there is a relation between tags (in the similarity sense), but not the nature of this relationship.

Another way of discovering links between tags is the application of association rules discovery algorithms [14]. The goal is to learn association rules from a folksonomy. The quality of a rule is measured by support and confidence values of tags describing an item. Although folksonomies should in a way present an alternative to creating taxonomies, the hierarchical nature of concept relations should not be discarded in explorations of hidden structures inside folksonomies. A way of creating taxonomic structure from a set of association rules is described in [15]. It consists of the creation of a graph whose vertices are a set of tags that form the association rules, whose edges are connections between tags which appear in the same rule (directed like the association), with edge weights given by the confidence of a rule. To get a hierarchical organization, only the edges with maximum weight are kept. Hierarchical relationships between tags were also mined from tagged resources based on the similarity of resources in tag vector space in [10].

The disadvantage of all of these methods is that they do not incorporate the content and context of tagging. By content, we mean the content of the tagged resources; by context, we denote the combination of tags applied by a given user to an item.

2.2 Tagging behavior

To understand which kind of structure can be mined from SBSs, the behaviour of users in such systems must be understood. An overview of SBSs and the incentives for their use is given in [8]. The investigation of user behaviour in tagging system done in [11] reports that the number of new unique tags decreases with the increase of the number of users. This suggests that users are using the same tags, which opens the possibility of pattern discovery through discovering how these tags are connected to each other. The same study [11] makes a categorization of tags into 9 categories: *identifying what (or who), identifying what it is, identifying who owns it, refining categories, identifying qualities, self reference* and *task organizing*. The first one includes the tags that are general in the sense that they represent general concepts known to most users, and the fourth one is used to better explain the tags that describe the high level objective category of the tagged resource. From such tag categories, some hierarchical structure between tags can be inferred. Therefore, some tags can be organized into hierarchical networks of association.

Another study [16] reports on the influences on the tagging behaviour of users. The authors studied an SBS by creating 4 experimental groups that were presented with no tags, all tags, popular tags and recommended tags, respectively. The authors found three major influences on tagging behaviour: user habits, community tagging, and the algorithm for selecting the tags that are presented to the user. Furthermore, the authors present a second categorization of tags into tag types: factual, subjective and personal. The study examines the utilization of tags and states that the major usage areas for tags are: self-expression, organizing, learning, finding and decision support. A major finding on the relationship between types and tasks is that while personal tags are most popular for organizing, factual tags are preferred for finding. From this brief survey of user behaviour in tagging systems, some conclusions for the task of a structure discovery algorithm can be drawn. First of all, the algorithm should show users only those tags that are general enough for everyone to understand their relation to the content of the resources "behind" the tag. This means that a content analysis of resources must be applied at some level. The other problem that should be solved is how the algorithm presents the tag structure to the user. This paper proposes to organize tags into hierarchical association networks.

2.3 Semantic and social web

Folksonomies depict users' view of content and a different way of conceptualizing knowledge as opposed to expert-created taxonomies. Although some works [17, 10] suggest that folksonomies are by themselves enough to create an organization of knowledge, this does not mean that they are an equivalent of ontologies. Ontology as a model of knowledge is not the same as the taxonomic way of organizing concepts into hierarchical structure. In fact, the social web does not reject the hierarchical structure of knowledge as such, but rather the way it is built in traditional settings. If ontology is understood as "the specification of conceptualization" [5] then it should not be mistaken for one of the ways of its expression and design – centrally controlled categorization. Folksonomies reject centrally controlled knowledge structures, but centrally controlled categorization does have advantages when it comes to consistency, comprehensiveness, etc. Therefore, in order to create a system of collective intelligence it is necessary to extend SBSs by some other knowledge than the one which comes only from the users and their interaction with the system. In [6], the author suggests a model of tagging that incorporates a source into it, and represents it with 4 arguments as: *Tagging (object, tag, tagger, source).*

In this schema, s*ource* represents “objects universe quantification” [5] that corresponds to a namespace. This means that *sources* can represent different communities (applications) or different categories (resource *i* has been tagged with tag *c* by the user *a* for category *o*). This is a way of conceptualizing the ontology of folksonomy [6]. To make this connection between the subjective views of users and objective reality of resource content, in [7] the author suggests 3 different groups of methods: standard-based, information extraction and knowledge discovery.

The idea of semantic and social web combination through an SBS is investigated in a number of works. In [2], the authors combine tags into hierarchies using Word Net. The authors create a hierarchical network of tags enriched by the different type of relations between them. A combination of clustering and semantic enrichment is presented in [18], which uses Wikipedia relations between tags previously clustered into several groups. The authors first create a co-occurrence matrix of tags and then apply a clustering scheme to create groups of similar tags. After clustering, the Wikipedia topic hierarchy is used to discover the nature of relationships between tags in a cluster. These relationships could be simple IS-A relationships between tags or more complex relationships. An LSA (Latent Semantic Analysis) was applied to folksonomy data in [20] to find the latent connections between tags and then regard the latent factors as high-level ontological concept.

3 Topical structure discoveries

In this paper, we adopt the structure discovery approach to semantic and social web combination by association rules discovery. The standard Apriori algorithm [1] is modified to take into account the relevance of a resource to a specific topic. A topic *o* represents a tag that can be regarded as a concept with a wide-enough scope to describe an ontological category that is generally understandable in a domain in which a system functions.

First of all, it is necessary to identify topics. As shown in [20], the number of underlying topics in tagging systems is small.

3.1 Topic selection

In order to define topics we searched for tags that satisfy 3 conditions:

1. high frequency,
2. good descriptors of content,
3. existence at a high level of an expert-made hierarchy.

The motivation for setting the first condition has to do with topic definition. If we consider topics as terms that are well understood terms that identify some category that means that it used often by different users. In that way the subjective and personal tags are discarded and only factual are used as topic candidates. The second condition has the task of filtering factual tags into relevant and not relevant tags. For example, if a resource (a web-page) is tagged with tag *europe*, but is in fact a page describing Asia, that even if *europe* can be generally considered as a factual tag in this particular case it is irrelevant, so it has to be discarded. Finally, the topic has to understandable both to the users who tag the resources and to the users who use SBSs for searching. Therefore, in order to find the topics which all users can relate to we introduced the third condition. This condition has to check if experts have identified some concepts as general concepts which could be comprehended by all users of a system.

Condition 1 is easily checked with a simple count of occurrence of tags inside a folksonomy (as high, we consider the 20 most frequently used tags). The second condition means that a tag can be used to clearly identify content that is tagged with it. To test this clarity of identification, a classifier that considers tag as a positive class is built. The examples used to train the model are some resources that are not a part of a system with the same annotation. In our evaluation, we took the 200 first Google hits (using the tag as query) as positive examples, and 100 hits of several[1] semantically clearly different query words as negative examples. If this classifier had a high recall when applied to the resources tagged with the tag, we considered the tag as a candidate topic tag. For the third condition, we considered whether the tag (or its lemma or an obvious synonym) appears in one of the first five levels in an expert-created hierarchy. Concretely, we used Open Directory Project (dmoz.org) hierarchy. If a tag *c* satisfies all the conditions, then it is considered as representing the topic *o*.

3.2 Structure discovery

Once topics are identified, it is necessary to find the relevance of resources to the selected topics. The use of relevance or topic specificity is necessary to avoid the inclusion of subjective or deliberately false tagging. For example if we look at the set of three resources (song lyrics) which are tagged by users with three tags:

- *i1* tags: lyrics, love
- *i2* tags: lyrics, love
- *i3* tags: music, fun

If user searches for love song lyrics, it should be enough for him to query the system with tags *lyrics* and *love*. If we discover association rule from these tags, one of them will be *lyrics*➔*love (sp=0.66; cf=1).* If confidence (*cf*) and support (*sp*) are calculated in this way (based on tags only), the results do not reflect the resource itself. However, the users do not search for tags, but for resources. “Love” can be used by some users to tag lyrics that are the lyrics of love song, and in that “mindset”, the tagging user uses the tag as a factual tag [16], but it could also be the personal tag of a user who reflected on lyrics he loves. In order to distinguish between these two, the relevance of a resource to a topic must be calculated. For example, let the relevance of i1 to a topic *love song* be 1, and relevance of resource i2 to the same topic be 0. Then the *lyrics*➔*love* association rule has different *sp* and *cf* values: 0.33 and 0.5 respectively. This means that for the query (*love* and *lyrics*), the probability of

[1] For our evaluation negative example query words were manually selected. This could be automated by using WordNet in the following way. If we sample a WordNet for a subgraph which contains term that is used as a positive class label, we can find semantically different terms by choosing the term that has the highest distance from the positive class label term.

a relevant resource being included in the answer set is 50%, which is more precise than the previous. For a given set *O* of topics *o*, the relevance of a resource $i \in I$ can be expressed for any topic *o* as *T(o,i)*. The value of *T(o,i)* can be determined in number of ways, by expert decision or ontology learning. In our algorithm, we consider only binary relevance obtained as a result of a classification model built to check for the second topic selection condition.

To find how the tags relate to each other we extend the association rule $x \rightarrow y$, where *x* and *y* are tags, to include topics. We express topic-specific association rule for a topic *o* and tags *x* and *y* as:

$$x \rightarrow_o y \qquad (2)$$

To discover association rules ($x \rightarrow_o y$), we extend the well-known formalism that defines association rules by their support and confidence.[2] We introduce two new measures: relevant support (3) and relevant confidence for a chosen topic *o* and its representing tag *c* (4):

$$suppR(o,c,x,y) = \frac{\sum_{k=1}^{d} T(o,i_k) \times |\{a|a \in A \wedge (a,i_k,c) \in F \wedge (a,i_k,x) \in F \wedge (a,i_k,y) \in F\}|}{\sum_{k=1}^{d} |\{a|\exists z \in C:(a,i_k,z) \in F\}|} \qquad (3)$$

$$confR(o,c,x,y) = \frac{\sum_{k=1}^{d} T(o,i_k) \times |\{a|a \in A \wedge (a,i_k,c) \in F \wedge (a,i_k,x) \in F \wedge (a,i_k,y) \in F\}|}{\sum_{k=1}^{d} T(o,i_k) \times |\{a|a \in A \wedge (a,i_k,c) \in F \wedge (a,i_k,x) \in F\}|} \qquad (4)$$

Using these two measures, association rules are discovered, and then based on these rules a network of association is created using the approach described in [14] and [15]

4 Case study and evaluation

In this section, we describe the results of a first evaluation of the method. This case study investigated tags from bibsonomy and compared them to the category structure from the Open Directory dmoz.org. All tests were run on a set of data from bibsonomy (www.bibsonomy.org) from 30/06/2007. The data used was only from the bookmarking part of the system (the system also allows for scientific publication tagging). It consists of 78544 resources with 163298 tag applications of 19978 distinct tags by 901 users.

The method consists of 4 parts:

- choice of topic(s);
- association rule discovery;
- creation of the network of association;
- evaluation;

[2] In traditional association rule mining, the support of $x \rightarrow y$ is the number of transactions containing x and y divided by the number of all transactions. The confidence of $x \rightarrow y$ is the number of transactions containing x and y divided by the number of transactions containing x [1]. Here, "tagging actions" (user-item relations) are the transactions, a transaction contains a tag if this tag was used by this user for this item, and filters ensure that the rule is relevant in the context of the topic and its representing tag. These adaptations are expressed in (3), (4).

As stated in section 3, we set three requirements that a given tag must meet in order to be considered as a topic. Based on the first one (tag frequency), we chose 4 topic candidates to run our tests. For the case study, we chose the following 4 tags: *web2.0*, *linux*, *programming* and *semanticweb*. The first tag (*web2.0*) was chosen because of its highest frequency, and the others were randomly selected from the tags that were in English. Table 1 shows the top 20 tags and their frequency in the used data set.

Table 1. Frequency of top 20 tags in the dataset (highlighted tags are considered topic candidates in the case study)

	Frequency	tag		Frequency	Tag
1	1904	web2.0	11	955	java
2	1675	allgemein	12	930	tagging
3	1646	blog	13	907	search
4	1593	software	14	764	research
5	1563	tools	15	764	semanticweb
6	1396	linux	16	763	news
7	1265	web	17	750	opensource
8	1206	reference	18	744	design
9	1174	programming	19	740	webdesign
1	1144	internet	20	722	wiki

The second condition called for tags that are good descriptors of content. To address this, a classification model was trained. In lack of a generally accepted truth we relied on Google for providing us with examples. For each of candidate tag (*web2.0*, *programming, semanticweb, Linux)*, the positive examples were the 200 (+/- due to the fact that some pages could not be retrieved) web pages on English language that are returned by Google when one of the tags is used as a query. As negative examples, we used 100 web pages (5 * the first twenty hits) which were given by Google as results for the following 5 queries: gardening, usability, cartoon, graphic, design. Using these inputs (200 positive and 100 negative examples), an SVM document classifier is trained for each candidate tag, using the Weka 3.5 machine learning tool. Kernels and parameters for each classifier were chosen to minimize the error, using 10-fold cross validation as the evaluation setting. Recall and precision for best performance classifiers for each of 4 tags is shown in table 2.

Table 2. Precision and recall of the classifiers learned from Google hits.

	web2.0	*Semanticweb*	*programming*	*linux*
precision	0.68	0.89	0.78	0.95
recall	0.75	0.82	0.85	0.93

After this, we used the trained models to assess the description power of a tag by measuring recall of the resources tagged with a candidate tag and 100 randomly selected resources that are not tagged with the candidate date set. In Table 3, we show the recall values from that result set. Since the classifiers model the ‘ground truth’, the recall shows how well users used a tag as a concept name for the ‘ground truth’ model. It also shows whether the resources can be classified using the tag name as a discriminative function, and it indicates a level of user ‘agreement’ on the concept that could be expressed by a keyword (tag). Table 3 shows the results for the 4 candidate tags.

Table 3. Recall results for built models with the respect to the ‘ground truth’ models

	web2.0	*semanticweb*	*programming*	*Linux*
recall	0.65	0.91	0.96	0.97

For one tag to be considered as good descriptor of content its recall must be over a predefined threshold. We set this to 85%, a value that should be evaluated against human judgments in future work. Due to its low recall, the tag *web2.0* is discarded as a candidate topic set. The low score could be explained as a result of very broad meaning of a term Web 2.0 and a number of themes ranging from business, entertainment, IT and other areas that can be described by this tag.

To satisfy the third condition we searched the Open Directory hierarchy to find top level entries that correspond to the candidate tags. Minor spelling differences were disregarded. This could be automated in a straightforward way. For the tag *programming,* an entry was found on the second (not considering the top level named TOP) hierarchical level (http://www.dmoz.org/Computers/Programming/), for the tag *linux* an entry was found on the fourth level (http://www.dmoz.org/Computers/Software/Operating_Systems/Linux/), and for the tag *semanticweb* an entry was also found on the fourth level (http://www.dmoz.org/Reference/Knowledge_Management/Knowledge_Representation/Semantic_Web/). Although all candidates have a corresponding tag in the directory, the last one (*semanticweb*) does not have any child entries. However, the method for finding hierarchical structure presented above finds hierarchical structure below the topic. Therefore, the method would produce a network which is incomparable with the structure of this keyword in dmoz.org (no nodes below it).

The next step in our method was to create association rules and to use them to derive hierarchical networks. This was done in order to address goals 2 and 3. For each candidate tag, we define two data sets: the *whole corpus (w)* dataset that represents the topical subset for a candidate tag (as defined in equation (4) above) and the *relevant corpus (r)* dataset which includes only those resources that have been classified by the classification model as positive classes (e.g., for *programming,* there are 1174 resources in *w* and *959* in r, for *linux* 1396 in *w* and 1354 in *r*, for *web2.0* 1904 in *w* and 1275 in *r*, for *semanticweb* 764 in *w* and 695 in *r*). After this, for both corpora association rules are learned, and an association network is created using the method described in [15], see Section 2.1.

In order to compare these networks with an expert-created network, a network was derived from the Open Directory Project in a following way. For each tag that appears in the generated association network, we searched for the corresponding concepts inside the dmoz.org category network and included a complete path (all the nodes) from the category to the concept that was found. For example, if we are searching for the *javascript* in a *programming* data set we add it to the network only if it exists in dmoz.org under the *programming* category, and also add all the categories from programming to javascript (Programing:Languages:JavaScript). The derivation of such network and not the use of the entire dmoz.org category network are necessary to overcome differences in the number of concepts (tags), since bibsonomy.org is a research project with a limited number of tags. Figure 1 shows the created network of associations for the whole *programming* corpus. Figure 2 shows the created network of associations for the relevant corpus of tag *programming.* Figure 3 shows the network derived from dmoz.org for *programming.*

Once all 3 networks were created, we calculated the similarity between them using the taxonomic overlap measure described in [12]. This measure calculates the similarity of taxonomic networks by calculating the overlap of two nodes' respective set of parents and children. Table 4 shows the taxonomic overlap between *programming* and *linux* whole and relevant networks and dmoz.org derived.

Table 4. Taxonomic overlap for 3 derived networks for the tags *programming* and *linux*

	Programming			Linux		
	whole	relevant	dmoz.org	whole	relevant	dmoz.org
whol		0.5982	0.3664		1	0.7923
relev	0.8059		0.7033	1		0.7923

The results in table 3 (for *programming*) show that the similarity between dmoz.org structure and structure derived using the approach described in the paper. The results in table 3 (for *linux*) show no improvement with the topic inclusion which can be explained by the high precision of tag *linux* – out of 1122 documents which were tagged with linux that could be retrieved, 1105 were classified as relevant, and both *w* and *r* corpora produced the same networks of association.

The high TO similarity between the dmoz.org structure and relevant corpus structure for both tags implies that there is a high similarity between expert-generated structure and structure that emerged from user interaction with the system.

This evaluation is only a first step in a thorough assessment of our method. Its main aims were to validate the basic approach by testing whether a random test set would produce "good-enough" quantitative results, and to gain some qualitative insights into the compared structures. For instance, the results showed that a pre-selection of domains may be needed to focus on topics that are "basic-level concepts" worthy of further subdivision both for experts and for taggers (unlike the example *semanticweb*). They also showed that while the expert-generated hierarchies are often more semantically constrained than those created by taggers (for example, *ajax* is a specialization of *javascript* in Fig. 3, whereas in Fig. 2 its frequent occurrence and central importance to users active in the bookmarking platform lift it to the level directly below *programming*, and make *web2.0* a descendant – essentially showing that tag co-occurrence not always indicates an IS-A relationship, but may indicate some other strong association). In future work, this will be extended by a more comprehensive quantitative evaluation that also compares this approach to other ways of deriving tag hierarchies such as the ones described in [10, 14].

5 Conclusion

Research presented in this paper showed an approach to structure discovery in folksonomies by combination of tag structure and content analysis. The goal was to show and evaluate a possible way of interaction between social and semantic views of web content organization.

The motivation of the paper was to show the use of data mining in an attempt to combine social and semantic web into a single framework. By evaluating three goals, we showed that users can specify categories using certain tags as well as state-of-the-art search engine, and that based on a combination of user inputs, we can artificially create such structures that are similar to structures created by human experts. We also showed that these structures could be enhanced by applying content analysis.

Future work should go more into direction of usability and try to measure how usable different structures are that could be discovered inside folksonomies, and whether topic enrichment can produce better results. Besides not tackling the usability side of the approach, our algorithm has some limitations. First, by setting the first condition of topic choice, we disregard those tags that are not frequent, and this could result in a failure to recognize such topics that are well structured in folksonomy by a limited number of users and resources and therefore a smaller number of tag applications. In such a way, small communities would be omitted from topic discovery. A possible way to overcome this is to use a different selection criterion for the first condition such as weights similar to tf.idf weight (used in information retrieval) applied to tags. Since our goal was not to build a better text classifier, we used a simple pre-processing methods (stemming) and a well-known text classifier method. The effects of choosing different pre-processing and mining methods could be investigated in further work. Also the algorithm applies only to the textual resources (documents, HTML pages), but many social bookmarking engines allow users to tag multimedia documents, and this is not considered in this work, although a similar approach could be applied to multimedia using a different kind of 'ground truth' (for example, given by image annotations). In order to completely evaluate our algorithm and the results, a study on a larger and more commercially oriented social bookmarking engine would be useful.

In sum, many of the approaches that emphasize collective intelligence as a fundamental element of Web 2.0 disregard the statistical (data mining) element that is in fact the underlying element of collective intelligence. As Web 2.0 brought a social revolution to the internet, we believe that a rise of Web 3.0 applications (www.twine.com, www.opencalais.com) and research results will bring some order into the chaos that came with Web 2.0.

References

1. Agrawal, R.; Srikant, R. (1994). Fast Algorithms for Mining Association Rules. In Proc. 20th Int. Conf. Very Large Data Bases, VLDB; Bocca, J. B.; Jarke, M.; Zaniolo, C., Eds.; Morgan Kaufmann: 1994.
2. Angeletou, S., Sabou, M., Specia, L., Motta, E., (2007) Bridging the Gap Between Folksonomies and the Semantic Web: An Experience Report. Workshop: Bridging the Gap between Semantic Web and Web 2.0, European Semantic Web Conference.
3. Atlee, T., and Pór, G. (2007) Collective Intelligence as a Field of Multi-disciplinary Study and Practic,, http://www.community-intelligence.com/blogs/public/2007/01/a_source_document_for_collecti.htmlm, retrieved on 18/10/2007
4. Begelman, G., Keller, P., and Smadja, F. (2006) 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.
5. Gruber, T. (1995). Toward principles for the design of ontologies used for knowledge sharing. Int. J. Hum.-Comput. Stud., 43(5-6):907-928.
6. Gruber, T. (2007). Ontology of Folksonomy: A Mash-up of Apples and Oranges, Published in Int’l Journal on Semantic Web & Information Systems, 3(2), 2007. (Originally published to the web in 2005)
7. Gruber, T. (2008). Collective knowledge systems: Where the social web meets the semantic web. Web Semantics: Science, Services and Agents on the World Wide Web, 6(1):4-13.
8. Hammond, T., Hannay, T., Lind, B., and Scott, J. (2005), Social Bookmarking Tools (I), D-Lib magazine, vol. 11, no. 4, 2005, p. 1082
9. Hassan-Montero, Y., and Herrero-Solana, V. (2006) 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.
10. Heymann, P., and Garcia-Molina, H., (2006), Collaborative Creation of Communal Hierarchical Taxonomies in Social Tagging Systems. Stanford InfoLab Technical Report 2006-10.
11. Golder, S. A. and Huberman, B. A. (2006). Usage patterns of collaborative tagging systems. J. Inf. Sci., 32(2):198-208.
12. Maedche, A., and Staab, S.: Comparing Ontologies - Similarity Measures and a Comparison Study. Internal Report 408, Institute AIFB, University of Karlsruhe, 2001.
13. Mika, P. (2005). Ontologies Are Us: A Unified Model of Social Networks and Semantics, Web Semantics Science Services and Agents on the World Wide Web (ISSN: 1570-8268), vol. 5, no. 1, 2007, p. 5.
14. Schmitz, C., Hotho, A., Jaschke, R., and Stumme, G. (2006).Mining association rules in folksonomies. In V. Batagelj, H.-H. Bock, A. Ferligoj, and A. iberna, editors, Data Science and Classifcation, Studies in Classifcation, Data Analysis, and Knowledge Organization, Berlin, Heidelberg, Springer, pages 261-270.
15. Schwarzkopf, E., Heckmann, D., Dengler, D., and Krner, A. (2007). Mining the structure of tag spaces for user modeling. In Complete On-Line Proceedings of the

Workshop on Data Mining for User Modeling at the 11th International Conference on User Modeling, pages 63-75, Corfu, Greece.

16. Sen, S., S. K. Lam, A. M. Rashid, D. Cosley, D. Frankowski, J. Osterhouse, M. F. Harper, and J. Riedl (2006). Tagging, communities, vocabulary, evolution. In CSCW '06: Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work, New York, NY, USA, pp. 181-190. ACM Pres
17. Shirky, C. (2006), Ontology is Overrated: Categories, Links, and Tags, http://www.shirky.com/writings/ontology_overrated.html, retrieved on 16/11/2007
18. Specia, L. and Motta, E., (2007) Integrating Folksonomies with the Semantic Web. In Proc. of ESWC'07, 2007.
19. Van der Wal T. (2004), Would We Create Hierarchies in a Computing Age?, December, 17. 2004 http://vanderwal.net/random/entrysel.php?blog=1598, retrieved on 16/06/2008
20. Wu, H., Zubair, M. , and Maly, K. (2006), Harvesting social knowledge from folksonomies, Proceedings of the seventeenth conference on Hypertext and hypermedia, August 22-25, 2006, Odense, Denmark
21. Wu, X., Zhang, L., Yu, Y.(2006). Exploring social annotations for the semantic web, Proceedings of the 15th international conference on World Wide Web, May 23-26, 2006, Edinburgh, Scotland

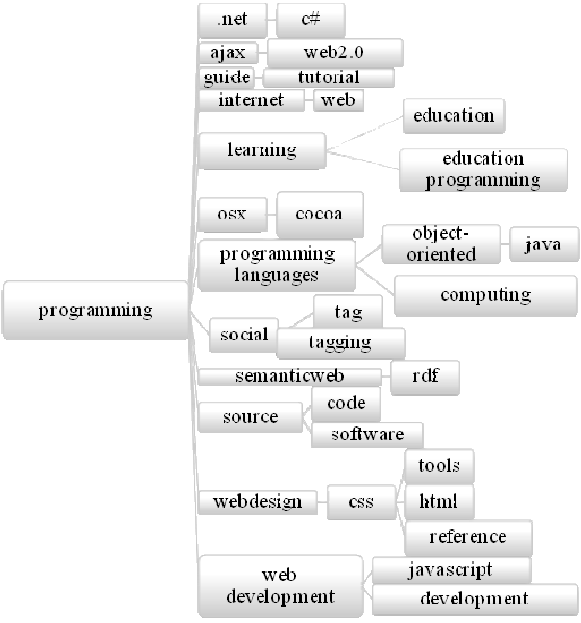


Fig. 1. Network of associations for the whole *programming* corpus.

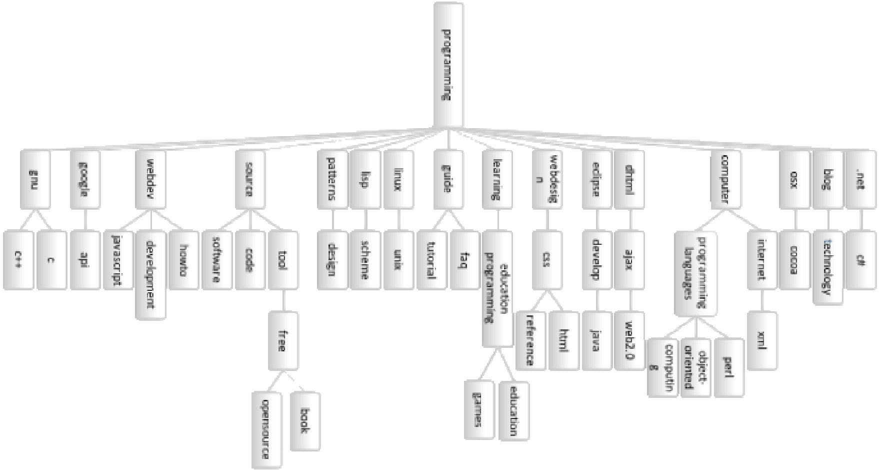


Fig. 2. Network of associations for the relevant *programming* corpus.

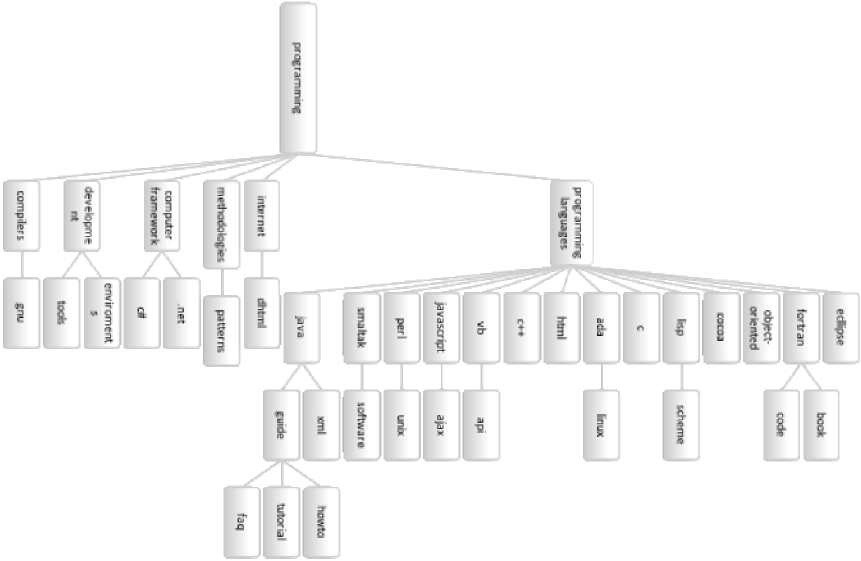


Fig. 3. Derived dmoz.org network of associations for *programming* corpus.

Wikipedia as the Premiere Source for Targeted Hypernym Discovery

Tomáš Kliegr[1], Vojtěch Svátek[1], Krishna Chandramouli[2], Jan Nemrava[1] and Ebroul Izquierdo[2]

[1] University of Economics, Prague, Department of Information and Knowledge Engineering, Winston Churchill sq. 4, Prague 3, 130 67, Czech Republic
[2] Queen Mary University of London, Multimedia and Vision Research Group, Mile End Road, London, E1 4NS, UK

Abstract. Targeted Hypernym Discovery (THD) applies lexico-syntactic (Hearst) patterns on a suitable corpus with the intent to extract one hypernym at a time. Using Wikipedia as the corpus in THD has recently yielded promising results in a number of tasks. We investigate the reasons that make Wikipedia articles such an easy target for lexicosyntactic patterns, and suggest that it is primarily the adherence of its contributors to Wikipedia's Manual of Style. We propose the hypothesis that extractable patterns are more likely to appear in articles covering popular topics, since these receive more attention including the adherence to the rules from the manual. However, two preliminary experiments carried out with 131 and 100 Wikipedia articles do not support this hypothesis.

1 Introduction

Most research in the field of hypernym discovery has been so far focused on non-statistical approaches, particularly on lexico-syntactic patterns (Hearst patterns) first introduced in [4]. Lexico-syntactic patterns were in the past primarily used on larger text corpora with the intent to discover all word-hypernym pairs in the collection. The extracted pairs were then used e.g. for taxonomy induction [10] or ontology learning [2]. This effort was, however, undermined by the low performance of lexico-syntactic patterns in the task of extracting *all* relations from a *generic* corpus. On this task, the state-of-the-art algorithm of Snow [9] achieves an F-measure of 36 %.

However, applying lexico-syntactic patterns on a *suitable document* with the intent to extract *one hypernym* at a time can achieve extraction accuracy close to 90% [6]. We refer to this approach as Targeted Hypernym Discovery (THD). Particularly, using THD in conjunction with Wikipedia has been found to bring promising results in a number of applications – ranging from named entity recognition [5] through query refinement [1] to image classification [6].

In this paper, we investigate the reasons why Wikipedia, particularly the first sentences of its articles, is such an easy target for lexicosyntactic patterns. Based on our prior experience with THD [1,6] we suggest the hypothesis that articles covering popular topics are more likely to contain an extractable hypernym (at the beginning of the text) than less popular ones.

Paper organization. Section 2 gives an overview of existing research; Section 3 discusses the role of Wikipedia in THD and proposes a hypothesis. The experimental setting and the experiments are described in Section 4. Conclusions presented in Section 5 summarize our findings and outline future work.

2 Related Research

Recently there has been a resurgence of interest in lexicosyntactic patterns [4], which can be in part attributed to the new possibilities given by the large amounts of textual content freely available on the web.

With respect to targeted hypernym discovery, [5] noticed that the Wikipedia encyclopedia can be used to improve the accuracy of Named Entity Recognition (NER) systems since it contains many articles defining named entities. For a given named entity extracted from text, the algorithm of [5] automatically found a Wikipedia entry for this entity and applied simple lexico-syntactic patterns to extract a hypernym from the first sentence of the article. The authors do not report on the number of correct and incorrect hypernyms and conclude that while they achieved an improvement on the NER task by using the extracted hypernyms as features of Conditional-Random-Fields (CRF) NER tagger, they believe that the hypernyms extracted from Wikipedia are too fine-grained for the classical NER task.[3]

In our recent work [1] we similarly tried to exploit hypernyms contained in Wikipedia for improving the performance of image retrieval through query refinement. In this research we tried to address some of the issues encountered in [5], particularly, we used a more sophisticated extraction grammar, relaxed the requirement for strict match between the article title and the named entity name by utilizing string similarity functions and increased the scope of extraction from the first sentence to the lead (introductory) section of the article. Preliminary results showed an improvement of image retrieval precision by 27%.

In [6] we performed additional experiments with an enhanced version of THD and introduced Semantic Concept Mapping, which exploits WordNet similarity measures in an effort to bridge the gap between the often too fine-grained hypernyms extracted from Wikipedia and a custom set of classes to which entities appearing in text need to be classified. The fact that the accuracy of hypernym discovery was 88% while the accuracy of mapping entities from text to 10 general Wordnet concepts was only 55% partly supports the conclusion of [5].

Apart from the free text, the (semi)structured information contained in Wikipedia articles—the infoboxes and the categories to which the article is assigned—are another prospective source of hypernyms. Further research will probably focus on fusing information retrieved from both the free text and the structured part of Wikipedia articles.

[3] I.e. classification to PERSON, LOCATION, ORGANIZATION and MISCELLANEOUS categories.

3 Wikipedia as the Source for Hypernym Discovery

WordNet is usually considered to be a gold-standard dataset for training and testing hypernym discovery algorithms [9]. Its structured nature and general coverage make it a good choice for general disambiguation tasks. However, we noticed in our previous work [8] that WordNet is less useful for analysis of text with high proportion of named entities.

Various free-text corpora have been used to overcome the problem of WordNet sparsity. State-of-the-art approaches based on mining patterns from text already achieve a higher F-measure than WordNet (with human judgment as the ground truth). Interestingly, Wikipedia as a free and comprehensive source of information plays a vital role in these efforts; one of the best results [9] were achieved by extending the TREC corpus with articles from Wikipedia.

Inspired by the promising results of [9], we used Wikipedia as the sole source of knowledge in our Targeted Hypernym Discovery tool. Unlike standard hypernym discovery, THD only discovers hypernyms for the given 'hypernym query'. It first uses the Wikipedia search API to find the most relevant articles for the query and then calls the NLP components available in the GATE framework [3] to extract the hypernym from the first 'is a' pattern that appears in each article. Empirical results suggest that this approach may be quite effective.

Experiments carried out in [6] hint that targeted hypernym discovery could successfully extract hypernyms from more than 90% of Wikipedia articles describing named entities[4], if the extraction grammar used in [6], which performed at 88% only using variations of the verb *to be*, were extended to cover some less frequent lexical constructions. Only about 6% of articles do not contain a hypernym extractable by a lexicosyntactic pattern.

Upon a manual inspection of the results we observed that, with a few exceptions, the first match of an ideal Hearst pattern in the article provided an informative and specific hypernym. Indeed, we found the vast majority of Wikipedia articles to open with clear definitions following the pattern "XYZ is a ... *detailed hypernym.*" Exceptions included cases such as "XYZ is a *cross* between a A and B". In this case, the word *cross* matches the lexicosyntactic pattern "XYZ is a ?" but cannot be accepted as a useful hypernym for XYZ.

We consider this rigidity in opening sentences surprising. Such a clearness and uniformity of articulation could be expected from an expert-created encyclopedia or thesaurus but not from a resource collaboratively created by unpaid volunteers whose only training comes, in general, from reading the Wikipedia guidelines. The Wikipedia's *Manual of Style*[5] has, indeed, a special section on first sentences, which instructs authors "to put the article title as the subject of the first sentence". A special article on the lead (introductory) section[6] states that the first paragraph "needs to unambiguously define the topic for the reader".

[4] In the assessment of 129 articles obtained through the Wikipedia 'random article' link, 102 articles (79%) describe named entities.

[5] `http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style#First_sentences`

[6] `http://en.wikipedia.org/wiki/Wikipedia:Lead_section`

Remarkably, Wikipedia contributors seem to follow these guidelines and even exceed them by refraining from the use of a more varied vocabulary when writing the opening definitions. For example, instead of "Diego Armando Maradona is a former Argentine football *player*"[7], the article title could start e.g. by "Diego Armando Maradona, a former Argentine football player, played in four World cups..." or even worse "D.A. Maradona was a *backbone* of Argentine football...".

However, when working on papers [1,6] we got the impression that hypernym discovery from shorter, less elaborate Wikipedia articles, which often describe uncommon entities, tends to be less successful than that from long articles on popular topics. If this empirical observation proved to be true, it would have implications on the predicted accuracy of extraction, depending on the kind of named entities expected.

In the rest of this paper, we try to empirically evaluate this hypothesis.

4 Experimental Setup

Two experiments were conducted in order to explore if there is a correlation between the popularity of Wikipedia article and the successfulness of hypernym discovery from this article. The experimental setup particularly included a) determining a measure of article popularity, b) determining the dataset and c) choosing a hypernym extraction algorithm. It should be noted that these choices were influenced by our previous work [1,6], which evaluated the usefulness of hypernyms discovered from Wikipedia for image retrieval.

Measure of Popularity. The popularity of Wikipedia articles can be measured in several ways. Experiment 1 takes the viewpoint of Wikipedia contributors and uses the number of inbound links from other Wikipidia articles. In turn, Experiment 2 takes the viewpoint of Wikipedia readers and uses the number of hits the article receives as the measure of popularity.

Datasets. The dataset and results[8] obtained in [1] were used as a basis of Experiment 1 presented in section 4.1. In [1] we used Wikipedia to discover possible hypernyms for a set of likely real-world queries from a specific domain. While multiple hypernyms from articles of varying popularity were extracted, in [1] we used background knowledge to filter out articles, which were not from the target domain. In contrast, Experiment 1 uses the full set of retrieved article-hypernym pairs.

The article-hypernym pairs extracted in our previous work [6] (which followed after [1]) form the dataset of Experiment 2. In [6] we established the name *Targeted Hypernym Discovery* for the task addressed by the algorithm introduced in [1] and used THD as a part of a larger framework, this time focused on entity classification. Nevertheless, we further evaluated our THD implementation on a

[7] Opening sentence of Wikipedia article on Maradona as of the time of writing.

[8] The results also included the relevancy of each article to the query, which is at the time of writing no longer available in Wikipedia's search results.

dataset consisting of 100 randomly selected Wikipedia articles. This experiment aimed at evaluating the hypernym extraction from the Wikipedia articles under the condition that an article defining the given entity is available. Article titles were used as queries for hypernym and the articles as the corpus. First sections (only these are processed by our THD algorithm) of the 100 articles contained approximately 100.000 words and 50.000 noun pairs.

Extraction Algorithm. The implementation of THD used in the experiments was built on top of the GATE NLP text engineering framework, which was used for shallow NLP parsing (particularly sentence splitting and part-of-speech tagging) of the document. The resulting tags were stored in annotations. These were further processed by a JAPE grammar engine. The extraction grammar matched several variation of the "is a" pattern, followed by a rather loosely defined sequence of unimportant words and finally by the desired hypernym. A detailed description of the THD implementation used in the experiments is presented in [1,6].

4.1 Experiment 1: Influence of Article Popularity (Links)

This experiment aimed to evaluate the influence of article popularity as measured by the number of inbound links from Wikipedia articles on the performance of THD. The underlying rationale is that the higher the number of linking articles, the higher the chance that other contributors would intervene if an article did not comply with the guidelines or its opening section was poorly/unclearly written.

The hypernym discovery implementation used [1] utilized Wikipedia's search interface to retrieve articles. Wikipedia MediaWiki search engine can use article popularity as measured by the number of articles that link to it as one of the ranking factors in addition to text-based relevance.[9] However, since we are only interested in article popularity, we try to mitigate the influence of text-based relevance by only involving articles whose title contains the entity for which the hypernym is sought, assuming that the part of relevance coming from the textual similarity between the article and the query is the same for all the relevant (returned) articles (up to a certain threshold). Manual inspection of the results showed that this technique was effective and the relevance measure generally reflected the relative popularity of the article subject in our dataset.

As test hypernym queries we used the surnames of ten top-rated NHL goalkeepers: Nabokov, Brodeur, Lundqvist, Luongo, Leclaire, Giguere, Miller, Bryzgalov, Turco and Kiprusoff. We already used this test set in our earlier work [1], where we found these words to provide enough ambiguity, as each represents a surname of several important persons from different fields, in addition to other meanings such as names of jobs, places or companies.

For each query, we downloaded the first section of all returned articles from Wikipedia up to a relevancy threshold of 50%. Redirects were followed but disambiguation articles and articles where the query term was not in the title were discarded. The resulting collection contained 131 documents (articles).

[9] http://www.mediawiki.org/wiki/Extension:Lucene-search

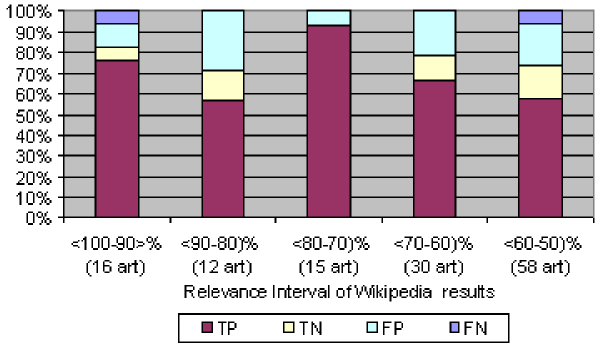


Fig. 1. Impact of article popularity on THD performance

Two human annotators annotated each of the documents. They were instructed to only mark the first hypernym per document (as does the used THD algorithm), regardless of how much more general this hypernym would be than the query. The annotation was not restricted to the context of ice hockey; hypernyms expressing all conceivable meanings of the original query were considered. The annotators agreed on 96% of annotations, which formed the ground truth.

With only five rules in our grammar we were able to discover 95 out of 124 hypernyms on which annotators agreed. Figure 1 shows the distribution of the THD outcome depending on article popularity. If the word marked as hypernym by the algorithm matched the ground truth then it was considered as a true positive (TP) if there was an annotation and true negative (TN) otherwise; false negative (FN) was the case when no hypernym was found but there was one according to the ground truth, and false positive (FP) in the opposite case. The case when both annotations were present but did not match was counted as two errors—FP as well as FN error—as this introduces a false hypernym and misses the true one. Due to latter case, some articles are counted twice on Figure 1.

We performed a statistical test to explore the significance of association between the article popularity (given by the respective relevance interval) and the successfulness of THD (1 for TP – correct hypernym extracted, 0 otherwise). The test used was one-sided Kendall's Tau B [7], which makes adjustment for ties and is also suitable for binary variables. A value of zero indicates the absence of association, while -1 or 1 mark perfect negative/positive association. Since the one-tailed Kendall's Tau B between the success of the extraction in these two groups is equal to 0.1281 (p-value of 0.055), we cannot reject the null hypothesis at a 5% significance level that there is not correspondence between article popularity (based on links) and the successfulness of hypernym extraction.

It should be noted that the experimental results may be skewed by the narrow character of the dataset and by the residual influence of article-query relevance.

4.2 Experiment 2: Influence of Article Popularity (Hits)

Another way of measuring the popularity of an article is the number of hits (views) it receives from the general public. Again, since anyone can edit Wikipedia, the higher the number of hits, the higher the chance that a random user would intervene if the article was poorly written.

This experiment was carried out with the sample of 100 articles describing named entities, which were randomly selected using the Wikipedia's random article link in [6]. The ground truth was established in a similar manner as in Experiment 1, but hypernyms were extracted for the topic of the article and not for a query. Additionally, articles were only annotated by one annotator[10]. The system failed to extract the correct hypernym from 14 articles: a Hearst-like pattern was not present in 8 articles.[11]. In 6 cases a Hearst-like pattern was present but was not matched by the grammar.

We evaluated whether the inability of the system to extract a hypernym is dependent on the number of hits each of the 100 articles obtained during a one-month period.[12] The range of hits was between 1 (for *Kielpino Kartuskie*) and 25.253 (for *Dead Space (video game)*), with the median value being 237. The result of extraction was marked as either 1 (success) or 0 (all other cases). The different kinds of error were alone too rare to be tested separately.

The test used was the same as in Experiment 1 – Kendall's Tau B. The value of the Kendall's Tau B in Experiment 2 was -0.037 (p-value of 0.320). This result hints that there is not a statistically significant correspondence between article popularity (based on hits) and the successfulness of hypernym extraction.

5 Conclusions and Further Work

In the paper we reviewed the recent work on Targeted Hypernym Discovery (THD) from Wikipedia and analyzed the reasons contributing to its success. Unlike existing approaches to hypernym discovery, THD selects a suitable document and extracts the most likely hypernym from it. The latter task is particularly interesting, since it seems that very good results are achieved by only considering the first hypernym matching the grammar from the article.

Our suggested explanation is that this can be partly attributed to the Wikipedia authoring guidelines. We conjectured that for articles covering less popular topics these guidelines are less rigidly applied, which may result in a worse performance of hypernym discovery algorithms.

[10] Exp. 1 showed that annotations by two humans do not significantly differ.

[11] Interestingly, the hypernym was a part of the article name in 6 cases, in 1 case there was no hypernym. The last case has interesting history. At the time of the preparation of the camera ready version of this paper, Wikipedia editors corrected the first sentence of this entry on R. E. Holz from "Richard E Holz, ... an American brass band composer,.." to now extractable "Richard E Holz was an American brass band composer..."

[12] during May 2008, using the http://stats.grok.se/en tool.

Our preliminary experimental results carried out altogether on 231 Wikipedia documents do not, however, support this hypothesis. Nevertheless, it should be noted that the value of the test in Experiment 1 was very close to the critical value for $\alpha = 5\%$. Since Experiment 1 was conducted on a sample from a specific domain and the article popularity was inferred from search relevance results, which might have introduced additional error, a larger scale experiment is thus indispensable to give the final answer.

6 Acknowledgements

The work has been partially supported by the European Commission under the IST research network of excellence K-SPACE of the 6th Framework programme, the work of Tomas Kliegr is also supported by grant GACR 201/08/0802 of the Czech Grant Agency.

References

1. Krishna Chandramouli, Tomáš Kliegr, Jan Nemrava, Vojtěch Svátek, and Ebroul Isquierdo. Query refinement and user relevance feedback for contextualized image retrieval. In *VIE 08: Proceedings of the 5th International Conference on Visual Information Engineering*. IET, 2008.
2. Philipp Cimiano and Johanna Voelker. Text2onto - a framework for ontology learning and data-driven change discovery. In *NLDB'05: Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems*, 2005.
3. Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. A framework and graphical development environment for robust NLP tools and applications. In *ACL 2002*, 2002.
4. Marti Hearst. Automatic acquisition of hyponyms from large text corpora. In *14th International Conference on Computational Linguistics*, pages 539–545, 1992.
5. Jun'ichi Kazama and Kentaro Torisawa. Exploiting Wikipedia as external knowledge for named entity recognition. In *EMNLP-CoNLL'07: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 698–707, 2007.
6. Tomáš Kliegr, Krishna Chandramouli, Jan Nemrava, Vojtěch Svátek, and Ebroul Izquierdo. Combining captions and visual analysis for image concept classification. In *MDM/KDD'08: Proceedings of the 9h International Workshop on Multimedia Data Mining*. ACM, 2008. To appear.
7. James E. De Muth. Basic statistics and pharmaceutical statistical applications, 2nd edn. *Journal Of The Royal Statistical Society Series A*, 1999.
8. Jan Nemrava. Refining search queries using wordnet glosses. In Helena Sofia Pinto and Martin Labsky, editors, *Poster and Demo Proceedings of EKAW 2006*, 2006.
9. Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. Learning syntactic patterns for automatic hypernym discovery. In *Advances in Neural Information Processing Systems*, number 17, pages 1297–1304, Cambridge, MA, 2005. MIT Press.
10. Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. Semantic taxonomy induction from heterogenous evidence. In *COLING/ACL 06*, pages 801–808, Sydney, Australia, 2006. ACM.

Clustering blog entries based on the hybrid document model enhanced by the extended anchor texts and co-referencing links[1]

Hiroshi Ishikawa, Masashi Tsuchida, Hajime Takekawa
Graduate School of Science and Technology, Shizuoka University
3-5-1 Johoku, Naka-ku, Hamamatsu, Japan

Abstract. In this paper, we propose a document vector space model where weights of noun terms vary depending on positions within the texts of blog entries as search results. We extend "extended anchor texts" (i.e., extra texts surrounding anchor texts) with the exponential potential such that the weight of a noun term decreases exponentially as the distance between the term and link increases. In order to cluster blog entries as search results, we use the hybrid vector space model which takes into account both texts including extended anchor texts and co-references of Web pages through links described in blog entries. We evaluate the effects of our scheme on clustering blog search results.

Keywords: Clustering, blogspace, extended anchor text, vector space model, link analysis

1 Introduction

As a phenomenon related to Web 2.0, a tremendous number of blogs have emerged each day. One of the salient features is the explosive growth of recent blog population. As blog entries are more frequently updated and more individual opinions can be read in real-time than general Web pages, blogs are increasingly important sources of information. The purposes of reading blogs range from individual interests to mining and "word of mouth" marketing.

Due to these situations, there is an urgent demand for technologies for gaining accurate information from a large scale of information sources rapidly. Most blog searches such as Technorati [Technorati 2008], display the search result set linearly by sorting the elements based on certain kinds of ranks such as relevance. As we live in the age of information explosion, such methods alone are insufficient to rapidly access necessary information.

As a solution to this problem, clustering a search result set based on similarities of the elements is considered as promising. Assuming that an appropriate label is automatically assigned to each cluster, the user is expected to more rapidly reach the necessary information by accessing only the clusters relevant to the purpose of the search.

Some Web (meta) search engines, such as Clusty [Clusty 2008] not only display the result set linearly but also display the folders of elements dynamically generated from the result of preceding clustering. Clusty allows the user to cluster the blog entries returned by dedicated blog search although the deployed technologies are not deeply described.

[1] This work was partially supported by KAKENHI (B) (19300026).

Generally, as the author of Web pages deliberately describes links to other Web pages, those pages are expected to be strongly related to each other. It is possible to increase the efficiency of clustering Web pages by analyzing link structures. It is possible to cluster elements based on their similarities, which inside use either texts or links, or both of them. Further, words in particular positions are often thought of more highly than those in other positions. Anchor texts associated with hyperlinks and title texts are such typical examples.

In this paper, in order to effectively cluster blogspace (i.e., a set of blog entries relevant to topics), we propose a document vector space model where the weights of terms vary depending on positions within the texts. We define “extended anchor texts” by extra texts surrounding anchor texts and extend them with the exponential potential such that the weight of a term decreases exponentially as the distance between the term and link increases. In reality, we use the hybrid vector space model which takes both texts including extended anchor texts and links within blog entries into account. In order to validate our approach as a preliminary work, we evaluate the effectiveness of our essential scheme.

We compare our work with related works. Shen et al. [Shen et al. 2006] proposed and utilized the extended anchor text model, which by their definitions consists of just a fixed number of words containing anchor texts, equally weighted. They used the models to classify Web pages, but not blog entries. Zhu et al. [Zhu et al. 2004] used a similar extended anchor model to automatically generate cluster labels in clustering generic Web pages.

2 Hybrid vector space model

We describe our hybrid vector space model deployed in clustering blog entries, which consists of texts and links. We have to take the following salient features of blog entries into account. Links contained by blog entries include hyperlinks, trackbacks, and comments. As the most common ways, hyperlinks link a page to another page by using the anchor tags as “<a href=”*url*”> *texts* </a>”. Texts surrounded by the corresponding anchor tags are called anchor texts. The anchor texts and the contents of hyperlink destinations are assumed to be highly similar. However, the origins of hyperlink jumps can be disclosed by analyzing Web access logs. For the reasons such as the purpose to conceal the existence of the hyperlink from the link destinations and the preference of the blog writers, urls are sometimes written directly in blog entries as a part of comments like “HP is *url*". We call such pseudo links direct links, where the sentences expressing the contents of the link destinations equivalent to anchor texts are not directly specified. We also consider texts surrounding such direct links as one kind of extended anchor texts deployed in clustering.

We use the hybrid vector space model consisting of link and document (text) vector space models in order to cluster blog entries. We describe the general flow of processes realizing our approach. First we make the link vector space model by focusing on co-references of Web pages through links. Next we make the document vector space model of blog entries with respect to the texts and weight it with our extended anchor text scheme. Then we hierarchically cluster the blog entries based on the similarities. When we merge two clusters, we use the furthest entries belonging to either of the clusters in order to measure the similarity between the two clusters. We use the hierarchical agglomerative clustering method with the prescribed cutoff threshold to avoid only one big cluster.

We collectively call hyperlinks and direct links explicit links. We describe the link vector space model based on such explicit links except image links because the images are rarely co-referenced and inappropriate for measuring similarities based on them.

First we extract all explicit links from each blog entry. The total number of distinct links throughout all blog entries is denoted as *n*. The counts of the url L_p appearing in the blog entry D_i is the component W_{ip} of the matrix corresponding to the link vector space. We cluster blog entries hierarchically based on the cosine similarity measure as follows:

$$sim(D_i, D_j) = \frac{W_{i1}W_{j1} + \ldots + W_{in}W_{jn}}{\sqrt{W_{i1}^2 + \ldots + W_{in}^2} \times \sqrt{W_{j1}^2 + \ldots + W_{jn}^2}} \quad (n \geq 1) \ldots (1)$$

This similarity measure also can apply to the document vector space model as described later. In order to construct the document vector space model, we extract only nouns except numerals and pronouns from blog entries and titles by using Japanese morphological analyzer. The varieties of verbs and adjectives, which we exclude, are not so rich; they are not so helpful to distinguish each entry. The total number of distinct nouns throughout all blog entries is denoted as *n*. The component W_{ik} of the document vector space model is calculated by combining tf_{ik} for frequencies of the noun term t_k appearing in the blog entry D_i and df_k for the counts of documents containing the noun term t_k as follows:

$$W_{ik} = tf_{ik} \times \left(\log \frac{n}{df_k} + 1 \right) \quad (1 \leq k \leq n) \ldots (2)$$

Blog titles are considered as the most concise abstracts of the blog entries. In general, the nouns contained by the blog titles are more highly weighted than those contained by the blog entries (bodies). The similarity can be measured by using the formula (1). In order to avoid the weighting of the insignificant words (i.e., stop words) such as “today”, “previously”, we don’t weight noun terms having tf*idf equal to or less than the average value of tf*idf even if they are included in extended anchor texts.

3 Extended anchor texts

We describe the general concept of our extended anchor text. We pay our attention to both texts surrounding so-called anchor texts and texts surrounding direct links. We don’t exclude image links for this time. For the blog writers, the information contained by the images is assumed to be important as well. Images are very important as sources of information as well. That is, terms surrounding image links are also more highly weighted.

Some blog writers describe explicit links around the beginning of the blog entries while others describe them in other positions such as the middle or end. We will explain how to determine the positions at the end of this section. We assume that the extended anchor texts follow the links at the beginning and precede the links at the other positions. We measure the distance forward and backward from the origin according to the positions, respectively.

We weight noun terms in the extended anchor texts. Prior to determining the weighting scheme, we measured how many noun terms averagely constitute one sentence. We randomly selected 250 sentences from a set of blog entries. The number of noun terms varies from 0 to 15. One sentence on the average contains 3.5 noun terms.

We assume that the origin is the position where the url is described. The unit of measuring the distance of the noun term is one noun term. The *ith* noun term from the link (the origin) has the distance *i-1(i>0)*. From the preparatory experiments, we assume that one sentence consists of three noun terms although a triplet of nouns does not always correspond to the same sentence. We weight noun terms sentence by sentence. Three noun terms within d_i corresponding to *ith* statement has the distances (3i-3,3i-2,3i-1). Weights g_i for noun terms within d_i are defined using the weight *g* for noun terms within d_1 as follows:

$$G = \{g_1, g_2, ..., g_M\} = \{g, \frac{g}{2}, ..., \frac{g}{2^{M-1}}\} \quad (M \geq 1)$$

How to determine *g* will be explained later. We use the exponentially decreasing potential in order to more strongly emphasize the effects of the distance than the linearly decreasing potential. On the other hand, traditional anchor texts and other extended anchor texts are regarded as constantly weighted independent of the distance from the origin.

We have analyzed the blog entries in order to determine the directions of weighting such as forward and backward from the origin (i.e., explicit link). We classify the origins of url into the beginning part, middle part, and ending part. After that, the anchor texts or explicit links are described. On the other hand, the blog writers describe sources of information and related Web pages collectively in the ending part. In this case, the titles and the concise descriptions of the Web pages are described in the anchor texts or followed by direct links. In the middle part, related noun terms precede anchor texts or explicit links. We assume the directions as follows: The origins precede extended anchor texts in the beginning part. The origins follow extended anchor texts in the middle or ending part.

We have done some experiments in order to determine the beginning part. We have counted 379913 noun terms in 9117 blog entries. One entry contains 41.67 noun terms on the average. As one sentence contains 3.5 noun terms on the average, one entry contains 11.9 statements on the average. By the prior analysis, urls appear around the second sentence on the average for the beginning part. Considering the length of individual blog entries, we decide the noun terms appearing until $C_i/5$ as the beginning part, C_i being the noun counts in the entry D_i. If urls are detected in the beginning part, the weighting direction is forward, otherwise backward.

4 Evaluation framework and hybrid vector space model

Here we describe the general framework for evaluating our approach through experiments. We have prepared a dataset of 37279 Japanese blog entries for the whole experiments, collected through crawling “Livedoor Blog” [Livedoor Blog 2008] and randomly selected blog entries co-referencing specific Web news articles. We use this dataset in order to construct the hybrid vector space model. Then we compare three clustering approaches including our proposed approach in order to evaluate the effectiveness of our approach.

As a preliminary work, our experiments focus on the following objectives: (1) Analysis of explicit links and construction of link vector space model, (2) Construction of the extended anchor text model with the exponential potential, (3) Validation of the relevance of the extended anchor texts to the feature terms of the entries, and (4) Validation of the effectiveness of the extended anchor texts and links in clustering

We have analyzed the dataset and have found that 4775 blog entries, about 13 % of the whole dataset, have explicit links (i.e., urls) and distinct 10790 urls appear for 23693 times in all. Around 80 % of distinct urls, 8035 urls are referenced only once. Only one entry references more than one Web page. The number of urls co-referenced by 2 to 10 entries is 2714 in all. That by 11 to 30 entries is 28; that by 31 to 147 (largest) entries is 13.

According to the analysis, most of urls are not co-referenced. Almost all urls co-referenced by more than and equal to 8 entries are either of shopping sites, auction sites, or application services embedded in the blog entries. The blog entries referencing “8 or more times co-referenced” urls can be categorized as advertisement- or affiliate-oriented blog entries. Those referencing “2 to 7 times co-referenced” urls are expected as the most promising as valuable information sources. This criterion is just the first approximation requiring elaboration; the threshold 7 or 8 here will change depending on the dataset.

We construct the link vector space model based on the urls extracted through the above analysis and cluster 4775 blog entries containing urls by using the link vector space model. If two clusters have the positive value as the similarity with respect to the urls, those two clusters are merged into one cluster. As the result, 3346 clusters are made. Among them, 390 clusters have co-references of urls and the remaining 2956 clusters have no co-references. Only one cluster has 147 blog entries, which co-reference advertizing sites and can be categorized as blog entries with commercial objects.

In order to construct the document vector space model, we also used the same dataset of consisting of 37279 blog entries as we used when we constructed the link vector space model. By focusing on entries containing a particular word, we evaluate the effectiveness of clustering based on our approach. As for the extended anchor texts, we consider the two nearest statements from the origin as the range of the extended anchor texts and we merge two clusters only if the similarity between them is equal to or larger than 0.4. We use the value 4.0, calculated by the following formula (3), as the weight *g* for the extended anchor texts. *N* is the total number of the entries and D_m denotes the entry *m (1<=m<=N).*

$$g = \frac{\sum_{m}^{N}\{Max_{D_m}(tf \times idf) - Avg_{D_m}(tf \times idf)\}}{N} \quad ...(3)$$

5 Evaluation of relevance of extended anchor texts

We will validate the relevance of our extended anchor texts with respect to the topics by checking whether noun terms related to the topics are successfully included by the extended anchor texts. We actually used three topics for the experiments, but we describe only the topic ”natto” (fermented soybeans) due to the limited space. The dataset contains references to Web pages related to these topics. As a background story, the broadcaster#2 aired the TV program, the production of the broadcaster#1, insisting that just taking natto every morning is an effective diet, which influenced consumers but was later found to be a frame-up.

We extract only blog entries dealing with “natto”, “diet”, “frame-up” and then use 16 entries referencing information sources related to this topic. We expect to extract the noun terms deeply associated with natto and frame-up from the extended anchor texts in the entries. Half of the ten entries contain explicit links and half contain normal anchor texts.

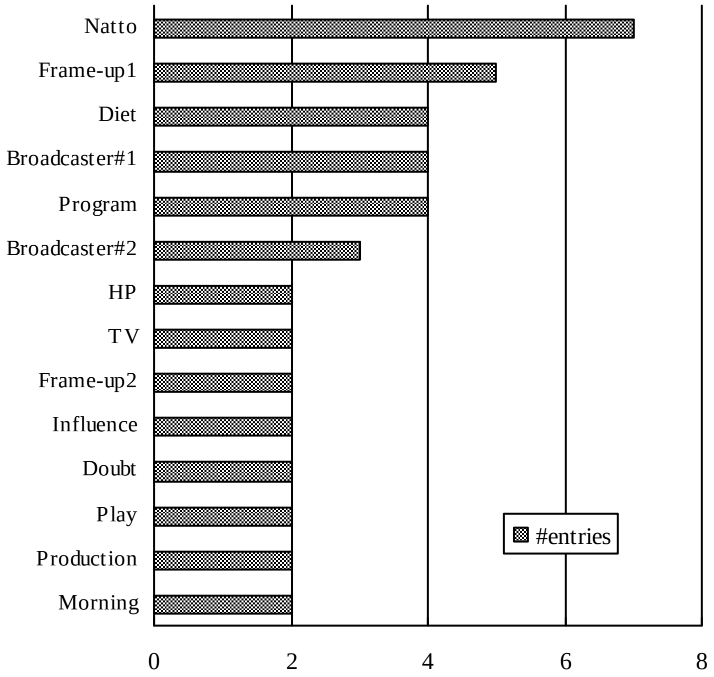


Figure 1. The number of entries with a specific term.

Among them, the noun terms related to this topic and those less related the topic appeared 63 and 103 times, respectively, in the extended anchor texts. Figure 1 illustrates the noun terms weighted in the extended anchor texts of at least two entries. We found that 63 (total, 166) noun terms could be extracted as related to the topic according to the definition of our extended anchor texts, especially from the beginning and ending parts.

In case of the explicit links described in the middle part where we only weight the extended anchor texts backward from the origin (explicit links), we sometimes failed to extract related noun terms although there are a lot of successful cases. So we have to refine or change our current rule of weighting the extended anchor texts. We think that it is possible to remedy this problem if we divide the whole blog entry into fragments by using some information such as line breaks and apply to such fragments our current rule of extracting noun terms from the extended anchor texts.

6 Comparison of clustering methods

In order to validate the effectiveness of our approach to clustering based on the hybrid vector space model enhanced by the extended anchor texts and co-referencing links, we compare the following three clustering methods: (Method1) Clustering based on document (text) vector space model, (Method2) Clustering based on both document- and link-vector space model, and (Method3, our approach) Clustering based on both document vector space model enhanced by our extended anchor text model and link vector space model.

We made experiments on clustering a set of blog entries containing specific noun terms

Table 1. The feature terms of clusters by the clustering method.

Method	Cluster	Feature terms
1	I	Natto Diet Broadcast Frame-up Dictionary
	II	Frame-up Cholesterol Natto Amino Broadcater1
2	III	Natto Frame-up Diet Broadcast Cholesterol
3	IV	Natto Broadcast Morning/Evening Dictionary Diet
	V	Natto Frame-up Cholesterol Diet Doubt

(i.e., search term) selected from the 37279 entry dataset. We actually did three sets of experiments by using three topics but we explain only about the ”natto” case as the others showed coherent results. We used “natto” as the search term for this clustering experiment deploying Method 1, Method 2, and Method 3. An article in a major news site reported that a frame-up of effective diet with natto by a certain broadcaster came to light. In advanced we found 5 entries co-referencing and discussing this topic. We name these “Co-A”, “Co-B”, “Co-C”, “Co-D”, and “Co-E”. There were 172 entries containing “natto”. We obtained 143, 137, and 138 clusters by Method 1, Method 2, and Method 3, respectively.

First, we focus on the cluster with 13 entries containing noun terms such as “diet” and “frame-up”, constructed by Method 1. We name this cluster “Cluster I”. We summed the tf*idf for each noun term and sorted them in descending order. We take the top 5 noun terms as feature terms for each cluster such as Cluster I (See Table 1). The entries Co-A, Co-B, Co-C belong to Cluster II as another cluster by Method 1. The entry Co-D belongs to Cluster I. Co-E belongs to another cluster. As a result of analysis of Cluster I, among 13 entries, only three entries were found related to the term “frame-up”. Two entries describe “natto diet”. Six entries describe “shortage of natto”. The rest is on the other topics.

As a result of clustering by Method 2, Cluster I and Cluster II were merged in Cluster III with 17 entries. “Frame-up” appeared again in the feature terms for the cluster. However, the topics are a mixture of “frame-up” and “shortage” (See Figure 2). As a result of clustering by Method 3, entries describing “shortage of natto” and ”natto diet” moved from Cluster I to Cluster IV. Entries about “shortage of natto” moved from other clusters than Cluster I to Cluster IV. The size of Cluster IV was 11. In addition, Co-A, Co-B, Co-C, and Co-D, entries about “frame-up” belong to Cluster V. The size of Cluster V is 6. Cluster V contains another entry about “frame-up”, originally belonging to Cluster I and Co-E, describing opinions about the news plainly. In summary, Method 3 more sharply separated topics and produced fewer mixtures of topics than Method 1 and Method 2.

In this experiment, our proposed method (Method 3) performed the best clustering in comparison to the other methods. We think that even if similarities based on co-references are not so large, the method by the extended anchor texts can compensate them.

The same author sometimes links the specific pages not related to the topic in question,

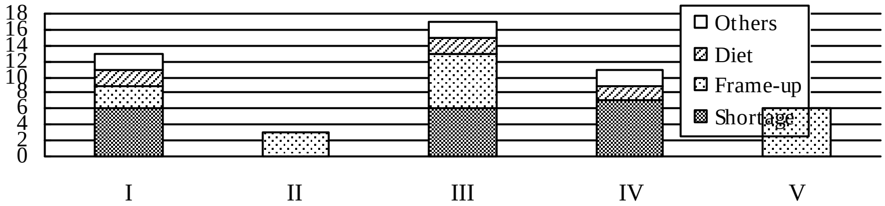


Figure 2. The number of entries with topics costituting clusters.

such as blog ranking and the author's own pages. In order to avoid the extracting of such "spam links" as co-references, we have to take special measures in dealing with explicit links in the blog entries written by the same author. While there may be errors, if we delete such spam blogs as commercial blogs and news blogs, the clustering method enhanced with the link vector space model is expected to allow clustering of entries describing more of the authors' opinions of the topics than of the description itself of the topics because such entries tend to omit the description by just linking Web pages as references.

In summary, the major advantage of deployment of co-referencing links in clustering is the possibility of clustering entries with explicit links, which describe fewer details about the topics and express more opinions about them. However, this approach may lead to mixtures of topics as a result of wrong clustering caused by spam links to non-relevant pages such as ranking pages and the authors' own pages. Especially, our extended anchor texts model is effective in labeling clusters with the extracted feature terms after clustering.

7 Conclusions

In this paper, we proposed the extended anchor texts model enhanced by exponential weighting. We also proposed blogspace clustering based on hybrid of document vector space model with extended anchor texts and link vector space model. Through the experiments, we have validated that it is possible to extract feature terms more relevant to the topic by using our extended anchor texts. We have found that use of co-referencing links is effective to blogspace clustering. The experiments have verified that the proposed scheme is at least effective for the dataset. Although some specific constants, such as three as the average number of nouns for one sentence, may depend on the dataset or the Japanese language, our essential schemes are neutral with respect to the dataset and the language.

One of the remaining issues as a preliminary work is to improve the rule of weighting the extended anchor texts from explicit links described in the middle of the blog entries. As the size of the dataset we used in the experiments was not so large and the sizes of constructed clusters were relatively small, the scalability of our approach with respect to large-scale blogspace is another big issue. As a third issue, the explicit links include lot of affiliate links to advertizing and commercial sites. We have to explore how to delete the affiliate links in order to more effectively cluster blog entries. Cleansing noun terms in constructing the document vector space model is another issue.

References

[Clusty 2008] Clusty http://clusty.com/ Accessed 2008.

[Livedoor Blog 2008] Livedoor Blog http://blog.livedoor.com/ Accessed 2008.

[Shen et al. 2006] D. Shen, et al.: A Comparison of Implicit and Explicit Links for Web Page Classification, Proc. Intl. Conf. WWW, 643-650, 2006.

[Technorati 2008] Technorati http://www.technorati.com/ Accessed 2008.

[Zhu et al. 2004] J. Zhu, et al.: PageCluster: Mining Conceptual Link Hierarchies from Web Log Files for Adaptive Web Site Navigation, ACM Trans. Internet Technology,185-208, 2004.

Author Index