Discriminative Clustering for Content-Based Tag Recommendation in Social Bookmarking Systems

Malik Tahir Hassan[1], Asim Karim[1], Suresh Manandhar[2], and James Cussens[2]

[1]Dept. of Computer Science
LUMS School of Science and Engineering
Lahore, Pakistan
[2]Dept. of Computer Science
The University of York
York, UK
{mhassan, akarim}@lums.edu.pk; {suresh, jc}@cs.york.ac.uk

Abstract. We describe and evaluate a discriminative clustering approach for content-based tag recommendation in social bookmarking systems. Our approach uses a novel and efficient discriminative clustering method that groups posts based on the textual contents of the posts. The method also generates a ranked list of discriminating terms for each cluster. We apply the clustering method to build two clustering models – one based on the tags assigned to posts and the other based on the content terms of posts. Given a new posting, a ranked list of tags and content terms is determined from the clustering models. The final tag recommendation is based on these ranked lists. If the poster's tagging history is available then this is also utilized in the final tag recommendation. The approach is evaluated on data from BibSonomy, a social bookmarking system. Prediction results show that the tag-based clustering model is more accurate than the term-based clustering model. Combining the predictions from both models is better than either model's predictions. Significant improvement in recommendation is obtained over the baseline method of recommending the most frequent tags for all posts.

1 Introduction

Social bookmarking systems have become popular in recent years for organizing and sharing resources on the Web. Such systems allow users to build a database of resources, typically Web pages and publications, by adding basic information (such as URLs and titles) about them and by assigning one or more keywords or tags describing them. The tags serve to organize the resources and help improve recall in searches. Individual users' databases are shared among all users of the system enabling the development of an information repository which is commonly referred to as a folksonomy [1]. A folksonomy is a collection of users, resources, and tags assigned by a user to a resource posted by him or her. Tag recommendation for new posts by users is desirable for two reasons. First, it ensures uniformity of tagging enabling better searches, and second, it eases the task of users in selecting the most descriptive keywords for tagging the resource.

Tag recommendation can have one of two goals: (1) to suggest tags tailored to individual users' preferences (the 'local' goal). and (2) to suggest tags that promote uniformity in tagging of resources (the 'global' goal). Tag recommendation can benefit from the tagging history of users and resources. However, when a user posts for the first time and/or the posted resource is new this historical information is less useful. In such cases, content-based tag recommendation is necessary, in which the contents of the resource are relied upon for tag recommendation.

This paper addresses task 1 of the ECML PKDD Discovery Challenge 2009 [2]. This task deals with content-based tag recommendation in BibSonomy, a social bookmarking system. The goal of tag recommendation is 'local', that is, to suggest tags tailored to individual users' preferences. Historical data of users, resources, and tags is available; however, the tag recommendation system must be able to provide good recommendations for unseen users and/or resources. Thus, the contents of resources must be utilized for tag recommendation.

Our solution to task 1 of the ECML PKDD Discovery Challenge 2009 relies on a novel discriminative clustering and term ranking method for textual data. We cluster the historical data of posted resources and develop a ranked list of discriminating tags and content terms for each cluster. Given a new posting, based on its contents, we find the best 3 clusters and develop a weighted list of tags and terms appropriate for tagging the post. If the poster's tagging history is available, then this provides a third ranked list of tags appropriate for the post. The final tag recommendation for the post is done by rules that select terms from the weighted lists. These rules also decide on the number of tags to recommend for each known poster. Extensive performance results are presented for the post-core training data provided by the challenge organizers.

The rest of the paper is organized as follows. We present the related work and motivation in Section 2. Section 3 presents details of our content-based tag recommendation approach, including description of the discriminative clustering and method. Data preprocessing and analysis is discussed in Section 4. The results of our approach are presented and discussed in Section 5. We conclude in Section 6.

2 Related Work and Motivation

Tagging resources with one or more words or terms is a common way of organizing, sharing, and indexing information. Tagging has been popularized by Web applications like image (e.g flickr), video (e.g. YouTube), bookmark (e.g. dec.icio.us), and publication (e.g. BibSonomy) sharing/organizing systems. Automatic tag recommendation for these applications can improve the organization of the information through 'purposeful' tag recommendations. Moreover, automatic tag recommendations ease the task of users while posting new resources.

The majority of the approaches proposed for tag recommendation assume that either the user posting the resource and/or the resource itself has been seen in the historical data available to the system [3–6]. If this is not the case, then only the contents of the posted resource can be relied upon. For social bookmarking systems, contents of resources are textual in nature requiring appropriate text and natural language processing techniques.

Content-based tag recommenders for social bookmarking systems have been proposed by [7, 8]. Lipczak's method extracts the terms in the title of a post, expands this set by using a tag co-occurrence database, and then filters the result by the poster's tagging history [7]. He reports significant improvements in performance after each step of this three step process. Tatu et al.'s method utilizes terms from several fields including URL and title to build post and user based models [8] . It relies on natural language processing to normalize terms from various sets before recommending them. We use terms from several fields of the posts including URL and title. We also study the impact of filling in missing and augmenting fields from information crawled from the Web.

A key challenge in tag recommendation is dealing with sparsity of information. In a typical collaborative tagging system, the vast majority of tags are used very infrequently making learning tagging behavior very difficult. This issue is often sidestepped in evaluation of tag recommenders when they are evaluated on post-core data with a high level of duplication (e.g. in [4, 6] post-core at level 5 is used). Our evaluation is done on post-core at level 2 data provided by the ECML PKDD Discovery Challenge 2009 [2].

Document clustering has been used extensively for organizing and summarizing large document collections [9, 10]. A useful characteristic of clustering is that it can handle sparse document spaces by identifying cohesive groups. However, clustering is generally computationally expensive. In the domain of collaborative tagging systems, clustering has been explored for information retrieval and post recommendation [11, 12]. In this paper, we explore the use of clustering for content-based tag recommendation. We use an efficient method that is practical for large data sets.

3 Discriminative Clustering for Content Based Tag Recommendation

Our approach for content-based tag recommendation in social bookmarking systems is based on discriminative clustering, content terms and tags rankings, and rules for final recommendations. We use a novel and efficient discriminative clustering method to group posts based on the tags assigned to them and based on their contents' terms. This method maximizes the sum of the discrimination information provided by posts and outputs a weighted list of discriminating tags and terms for each cluster. We also maintain a ranked list of tags for seen users. Tags are suggested from these three rankings by intuitive rules that fuse the information from the lists. The rest of this section presents our approach in detail.

3.1 Problem Definition and Notation

A social bookmarking system, such as BibSonomy [13], allows users to post and tag two kind of resources: Web bookmarks and publications. Each resource type is described by a fixed set of textual fields. A bookmark is described by fields like URL, title, and description, while a publication is described by fields in the standard bibtex record. Some of these fields (like title for bookmarks) are mandatory while others are optional. This

textual information forms the content of the resource. Each user who posts a resource must also assign one or more tags for describing the resource.

Let $p_i = \{u_i, \mathbf{x}_i, \mathbf{t}_i\}$ denotes the ith post, where u_i is the unique user/poster ID, and $\mathbf{x}_i$ and $\mathbf{t}_i$ are the vector space representations of the post's contents and tags, respectively. If T is the size of the vocabulary then the ith post's contents and tags can be written as $\mathbf{x}_i = \{x_{i1}, x_{i2}, \ldots, x_{iT}\}$ and $\mathbf{t}_i = \{t_{i1}, t_{i2}, \ldots, t_{iT}\}$, respectively, where x_{ij} (t_{ij}) denotes the frequency of term j (tag j) in post i. Note that an identical vector space model is used to represent both content terms and tags, $t_{ij} \in \{0, 1\}, \forall i, j$, and $x_{ij} \geq 0, \forall i, j$. The historical data contain N posts. The tag recommender suggests tags for a new post i described by u_i and $\mathbf{x}_i$. The user u_i and resource described by content $\mathbf{x}_i$ may or may not appear in the historical data.

Let $TG(i)$, $TM(i)$, and $TU(i)$ be the ranked list of tags from clustering, terms from clustering, and user tags, respectively, corresponding to the ith post. The actual tags recommended for post i, denoted by $TR(i)$, are determined from these ranked lists by intuitive rules.

Given a test data containing M posts, the performance of the tag recommender is evaluated by averaging F1-score of each prediction over the entire test data.

3.2 Discriminative Clustering for Tag and Term Ranking

The historical data of N posts is clustered into $K \ll N$ groups using a novel discriminative clustering method. This method is motivated from the recently proposed DTWC algorithm for text classification [14]. It is an iterative partitioning method that maximizes the sum of discrimination information provided by each textual content (a post, in our setting) between its assigned cluster and the remaining clusters. The key ideas include discriminative term weighting, discrimination information pooling, and discriminative assignment. Unlike other partitioning clustering methods, this method does not require the explicit definition of a similarity measure and a cluster representative. Furthermore, it builds a ranked list of discriminating terms for each cluster implicitly. The method is computationally more efficient than popular methods like the k-means clustering algorithm. We perform two clusterings of the historical data – one based on the content terms $\mathbf{x}$ and the other based on the tags $\mathbf{t}$ of the posts in the data. In the following description, we develop the method for content terms only; the method as applied to tags will be similar.

First, an initial clustering of the data is done. This can be done randomly or, less efficiently especially for large collections, by a single iteration of the k-means algorithm with the cosine similarity measure. Given this clustering, a discriminative term weight w_j^k is computed for each term j in the vocabulary and for each cluster k as [14]

$$w_j^k = \begin{cases} p(x_j|k)/p(x_j|\neg k) & \text{when } p(x_j|k) > p(x_j|\neg k) \\ p(x_j|\neg k)/p(x_j|k) & \text{otherwise} \end{cases}$$

where $p(x_j|k)$ and $p(x_j|\neg k)$ are the probabilities that term j belongs to cluster k and the remaining clusters ($\neg k$), respectively. The discriminative term weight quantifies the discrimination information that term j provides for cluster k over the remaining clusters. Note that this weight is expressed as a probability ratio and is always greater

than or equal to 1. The probabilities are computed by maximum likelihood estimation from the historical data.

Having computed the discriminative term weights for the current clustering, two discrimination scores can be computed for each post i. One score, denoted as $Score^k(\mathbf{x}_i)$, expresses the discrimination information provided by post i for cluster k, whereas the other score, denoted as $Score^{\neg k}(\mathbf{x}_i)$, expresses the discrimination information provided by post i for clusters $\neg k$. These scores are computed by linearly pooling the discrimination information provided by each term x_j in post i as [14]

$$Score^k(\mathbf{x}_i) = \frac{\sum_{j \in Z^k} x_j w_j^k}{\sum_j x_j} \text{ and}$$

$$Score^{\neg k}(\mathbf{x}_i) = \frac{\sum_{j \in Z^{\neg k}} x_j w_j^k}{\sum_j x_j}$$

In these equations, $Z^k = \{j | p(x_j|k) > p(x_j|\neg k)\}$ and $Z^{\neg k} = \{j | p(x_j|\neg k) > p(x_j|k)\}$ are sets of term indices that vouch for clusters k and $\neg k$, respectively. Each post, described by its contents $\mathbf{x}$, is then reassigned to the cluster k for which the cluster score $f^k = Score^k(\mathbf{x}) - Score^{\neg k}(\mathbf{x})$ is maximum. This is the cluster that makes each post most discriminating among all the clusters.

The overall clustering objective is to maximize the sum of discrimination information, or cluster scores, of all posts. Mathematically, this is written as

$$\text{Maximize } J = \sum_{i=1}^{N} \sum_{k=1}^{K} I^k(\mathbf{x}_i) \cdot f^k$$

where $I^k(\mathbf{x}_i) = 1$ if post i is assigned to cluster k and zero otherwise. Iterative reassignment is continued until the change in the clustering objective becomes less than a specified small value. Typically, the method converges satisfactorily in fewer than 15 iterations.

The discriminative term weights for the terms in the index set Z^k are ranked to obtain the weighted and ranked list of terms for cluster k. As mentioned earlier, clustering is also performed based on the tags assigned to posts. This clustering yields another weighted and ranked list of tags for each cluster.

It is worthwhile to point out that the term-based clustering is done on both the training and testing data sets. This approach allows the terms that exist only in the test data to be included in the vocabulary space, and for such terms to be available for recommendation as tags.

Given a new post i described by $\mathbf{x}_i$, the best cluster for it is the cluster k for which the cluster score f^k is a maximum. The corresponding ranked list of terms and tags for post i are denoted by $TM(i)$ and $TG(i)$, respectively. These ranked lists contain the most discriminating tags for post i based on its contents.

3.3 Final Tag Recommendation

Given a new post, and based on the contents $\mathbf{x}$ of the post, two ranked lists of terms appropriate for tagging are generated by the procedures described in the previous section.

If the user of the post appears in the historical data, then an additional list of potential tags can be generated. This is the ranked list of tags $TU(i)$ used by the user of post i The ranking is done based on frequency. Moreover, the average number of tags per user is computed and used while recommending tags for seen users.

The final list of tags for post i is made by simple and intuitive rules that combine information from all the lists. Let S be the number of tags to recommend for post i. Then, the final list of tags for the post is given by the following algorithm:

$$TR(i) = TG(i)[1:P] \cap TM(i)[1:Q]$$

$$\text{IF } TU(i)| \neq \oslash \text{ THEN } TR(i) = TR(i) \cap TU(i)[1:R]$$

$$\text{IF } |TR(i)| < S \text{ THEN add top terms from } TG(i), TM(i) in TR(i)$$

In the above algorithm, P, Q, and R are integer parameters that define how many top terms to include from each list. If after taking the set intersections $|TR(i)| < S$ then the remaining tags are obtained from the top tags and terms in $TG(i)$ and $TM(i)$, respectively. In general, as seen from our evaluations, $R \leq Q \leq P$, indicating that $TG(i)$ is the least noisy source and $TU(i)$ the most noisy source for tags.

4 Evaluation Setup

4.1 Data and their Characteristics

We evaluate our approach on data sets made available by the ECML PKDD Discovery Challenge 2009 [2]. These data sets are obtained from dumps of public bookmark and publication posts on BibSonomy [13]. The dumps are cleaned by removing spammers' posts and posts from the user dblp (a mirror of the DBLP Computer Science Bibliography). Furthermore, all characters from tags that are neither numbers nor letters are removed. UTF-8 encoding and unicode normalization to normal form KC are also performed.

The post-core at level 2 data is obtained from the cleaned dump (until 31 December 2008) and contain all posts whose user, resource, and tags appear in at least one more post in the post-core data. The post-core at level 2 contain 64,120 posts (41,268 bookmarks and 22,852 publications), 1,185 distinct users, and 13,276 distinct tags. We use the first 57,000 posts (in content ID order) for training and the remaining 7,120 posts for testing.

We also present results on the test data released as part of task 1 of the ECML PKDD Discovery Challenge 2009. This data is cleaned and processed as described above, but it contain only those posts whose user, resource, or tags do not appear in the post-core at level 2 data. This data contain 43,002 posts (16,898 bookmarks and 26,104 publications) and 1,591 distinct users. For this evaluation, we use the entire 64,120 posts in the post-core at level 2 for training and test on the 43,002 posts in the test data.

These data sets are available in the form of 3 tables – tas, bookmark, and bibtex – as described below. The content of a post is defined by the fields in the bookmark and bibtex tables, while the tags appear in the tas table.

tas fact table; who attached which tag to which post/content. Fields include: user (number; user names are anonymized), tag, content id (matches bookmark.content id or bibtex.content id), content type (1 = bookmark, 2 = bibtex), date

bookmark dimension table for bookmark data. Fields include: content id (matches tas.content id), url hash (the URL as md5 hash), url, description, extended description, date

bibtex dimension table for BibTeX data. Fields include: content id (matches tas.content id), journal, volume, chapter, edition, month, day, booktitle, howPublished, institution, organization, publisher, address, school, series, bibtexKey (the bibtex key (in the @... line)), url, type, description, annote, note, pages, bKey (the "key" field), number, crossref, misc, bibtexAbstract, simhash0 (hash for duplicate detection within a user – strict – (obsolete)), simhash1 (hash for duplicate detection among users – sloppy –), simhash2 (hash for duplicate detection within a user – strict –), entrytype, title, author, editor, year

A few tagging statistics from the post-core data are given in Table 1 and Figure 1. These statistics are used to fix the parameter S (number of recommended tags) for known users. For unseen users, S is set at 5.

Table 1. Post-core at level 2 data statistics

	Avg	**Min**	**Max**	**Std. Deviation**
No. of tags per post	4	1	81	3.3
No. of posts per user	54	2	2031	162.9
No. of tags per user	62	1	4711	214.5
Frequency of tags	19	2	4474	106.9

4.2 Data Preparation

We explore tag recommendation performance on original contents, contents that have been augmented by crawled information, and contents that have been augmented and lemmatized.

The vocabulary for the vector space representation is formed from the tags and content terms in the training and testing sets. Selected content fields are used for gathering the content terms. For bookmark posts, the selected fields are url, description, and extended. For publication posts, the selected bibtex fields are booktitle, journal, howpublished, publisher, series, bibtexkey, url, description, annote, note, bkey, crossref, misc, bibtexAbstract, entrytype, title, and author. As mentioned earlier, the tags, which appear in the tas table, are also included in the vocabulary.

We remove all the non-letter and non-digit characters, but retain umlauts and other non-Latin characters due to UTF-8 encoding. All processed terms of length greater than or equal to three are retained. The tags are processed similarly, but without considering the token length constraint.

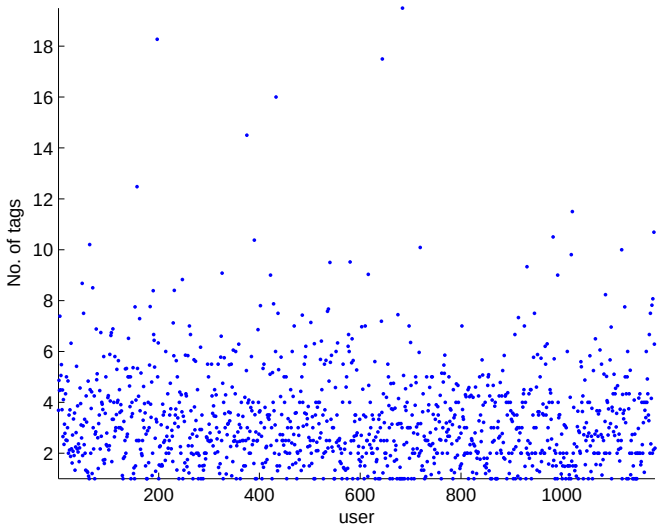

Fig. 1. Number of tags assigned to posts by users

Crawling Crawling is done to fill in and augment important fields. For bookmark posts, the extended description field is appended with textual information from <TITLE>, <H1> and <H2> HTML fields of the URL provided in the posts.

For publication posts, missing abstract field are filled using online search. We use the publication title to search for its abstract on CiteULike [15]. If the article is found, and its abstract is available on CiteULike, the bibtexAbstract field of the post is updated. CiteULike is selected because its structure is simpler and it does not have any restrictions on the number of queries (in a day for example).

Lemmatization We also explore lemmatization of the vocabulary while developing the vector space representation. Lemmatization is different from stemming as lemmatization returns the base form of a word rather than truncating it. We do lemmatization using TreeTagger [16]. TreeTagger is capable of handling multiple languages besides English. We lemmatize the vocabulary using English, French, German, Italian, Spanish and Dutch languages. The procedure, in brief, is as below:

1. TreeTagger is run on the vocabulary file once for each language: English, French, German, Italian, Spanish and Dutch.
2. TreeTagger returns the output file containing token, pos, lemma. The lemma is "<unknown>" if a token is not recognized in that language.
3. Using this "<unknown>" word, we combine the output of all six lemmatized files. If a term is not recognized by any language, the term itself is used as lemma.

4. If a word is lemmatized by more than one language, then lemmas are prioritized in the sequence: English, French, German, Italian, Spanish, Dutch. The first lemma for the word is selected.

4.3 Evaluation Criteria

The performance of tag recommendation systems is typically evaluated using precision, recall, and F1 score, where the F1 score is a single value obtained by combining both precision and recall. We report the precision, recall, and F1 score averaged over all the posts in the testing set.

5 Results

In this section, we present and discuss the results of our discriminative clustering approach for content based tag recommendation. We start off by evaluating the performance of the clustering method.

5.1 Clustering Performance

The performance of the discriminative clustering method is evaluated on the entire 64,120 posts of the post-core at level 2 data. We cluster these posts based on the tags assigned to them. After clustering and ranking of tags for each cluster, we recommend the top 5 tags from the ranked list for all posts in each cluster. The average precision, recall, and F1 score percentages obtained for different values of K (number of desired clusters) is shown in Table 2.

The top 5 tags become increasingly accurate recommendations as the number of clusters is increased, with the maximum recall of 48.7% and F1 score of 30.6% obtained when $K = 300$. These results simulate the scenario when the entire tag space (containing 13,276 tags) is known. Furthermore, there is no separation between training and testing data. Nonetheless, the results do highlight the worth of clustering in grouping related posts that can be tagged similarly.

Table 2. Performance of discriminative clustering of posts using the tags assigned to them (post-core at level 2 data)

K	10	50	100	200	300
Act. Clusters	10	48	95	189	274
Av. Precision (%)	12.5	19.2	22.3	25.2	26.9
Av. Recall (%)	21.0	32.8	38.6	45.9	48.7
Av. F1-score (%)	13.7	21.4	25.0	28.7	30.6

Table 3 shows the top ranked tags for selected clusters. It is seen that the discriminative clustering method is capable of grouping posts and identifying descriptive tags for

Table 3. Top tags for selected clusters (K = 200)

/ No.	Top Discriminating Tags
1	svm, ki2007webmining, mining, kernels, textmining, dm, textclassification
2	windows, freeware, utility, download, utilities, win, shareware
3	fun, flash, games, game, microfiction, flashfiction, sudden
4	tag, cloud, tagcloud, tags, folksonomia, tagging, vortragmnchen2008
5	library, books, archive, bibliothek, catalog, digital, opac
6	voip, mobile, skype, phone, im, messaging, hones
7	rss, feeds, aggregator, feed, atom, syndication, opml
8	bookmarks, bookmark, tags, bookmarking, delicious, diigo, socialbookmarking

each group of posts. Noisy tags are not ranked high in the lists. It is even able to discriminate and group posts of different languages (not shown in this table), especially when clustering is based on content terms. Two valuable characteristics of the discriminative clustering method are its stability and efficiency. The method converges smoothly (Figure 2) usually within 15 iteration. More importantly, especially considering the large post by vocabulary sizes involved, is the efficiency of the method. Each iteration of the method completes within 3 minutes, even for the large $107,122 \times 317,283$ data for the content-based clustering of the post-core plus task 1 test data.

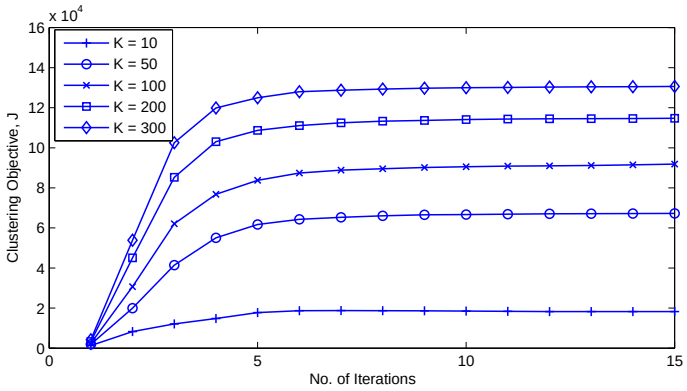

Fig. 2. Discriminative clustering convergence curves (clustering posts based on tags)

5.2 Tag Recommendation Using *TG* and *TM* Only

In this section, we discuss the performance of recommending the top 5 tags from the $TG(i)$ or $TM(i)$ list of each post i. This evaluation is done on the testing data of 7,120 posts held out from the post-core at level 2 data. The clustering model is based on the first 57,000 posts (in content ID order) from the data. In this evaluation, the original

data, without augmentation with crawled information, is used for creating the vector space representation.

The recommendation results for different K values are given in Table 4. Results are shown for the case when only the top cluster for each post is considered, and for the case when the top three clusters of each post are merged in a weighted manner (using cluster score and discriminative term weights). It is observed that merging the lists of the top three clusters always gives better performance. Moreover, recommendations based on $TG(i)$ are always better than those based on $TM(i)$ indicating that the term-based clustering is more noisy than that based on tags. We also find out that $K = 200$ yields the highest recommendation performances.

Table 4. Tag recommendation performance (average F1-score percentages) using *TG* or *TM* only for original data

K	10	50	100	200	300
***TG* Only (Best Cluster)**	6.6	7.4	8.7	8.7	7.2
***TG* Only (Top 3 Clusters)**	7.3	8.2	9.5	10.6	9.1
***TM* Only (Best Cluster)**				6.3	
***TM* Only (Top 3 Clusters)**				7.8	

5.3 Tag Recommendation Using All Lists

In this section, we evaluate the performance of our approach when utilizing information from all lists. We also evaluate performance on original, crawled, and crawled plus lemmatized data. These results are shown in Table 5. For this evaluation, we fix $K = 200$ and use the top three clusters for building $TG(i)$ and $TM(i)$.

The first column (identified by the heading *TF*) shows the baseline result of recommending the top 5 most frequent tags in the training data (57,000 posts from post-core data). It is seen that our clustering based recommendation improves performance beyond the baseline performance. The second and third columns show the performance of recommending the top 5 terms from $TG(i)$ and $TM(i)$, respectively. The predictions of the tag-based clustering always outperform the predictions of the term-based clustering. In the fourth column, we report results for the case when the top 5 recommended tags are obtained by combining $TG(i)$ and $TM(i)$, as described in Section 3.3. These results are significantly better than those produced by each list independently.

The fifth column shows the results of combining all lists, including the user list $TU(i)$ when known. This strategy produces the best F1 score of 15.5% for the crawled data. This is a significant improvement over the baseline F1 score of 7.0%.

Table 5 also shows that filling in missing fields and augmenting the fields with crawled information improves performance. Lemmatization does not help, probably because users do not necessarily assign base forms of words as tags.

Table 5. Tag recommendation performance (average F1-score percentages) for processed data (K = 200; prediction based on top 3 clusters). The bottom line shows performance on task 1 test data

Data / Lists)	***TF***	***TG***	***TM***	***TG, TM***	***TG, TM, TU***
Original Contents	7.0	10.6	7.8	11.5	12.8
Crawled Contents	7.0	12.3	10.4	14.3	15.5
Crawled+Lemmatized Contents	7.0	11.7	9.7	13.3	14.6
Task 1 Test Data (Crawled)	1.1	4.9	3.2	5.2	5.4

5.4 Tag Recommendation for Task 1 Test Data

We report the performance of our approach on task 1 test data released by the challenge organizers on the bottom line of Table 5. We filled in missing and augmented other fields by crawled information. No lemmatization is done. The final vocabulary size is equal to 317,283 terms making the tag recommendation problem very sparse. The baseline performance of using the 5 most frequent tags from the post-core at level 2 (the training data for this evaluation) is the F1 score of 1.1% only. By using our discriminative clustering approach, the average F1 score reaches up to 5.4%. This low value is attributable to the sparseness of the data, and it is unlikely that other methods can cope better without extensive semantic normalization and micro modeling of the tagging process.

6 Conclusion

In this paper, we explore a discriminative clustering approach for content-based tag recommendation in social bookmarking systems. We perform two clusterings of the posts: one based on the tags assigned to the posts and the second based on the content terms of the posts. The clustering method produces ranked lists of tags and terms for each cluster. The final recommendation is done by using both lists, together with the user's tagging history if available. Our approach produces significantly better recommendations than the baseline recommendation of most frequent tags.

In the future, we would like to explore language specific models, incorporation of a tag extractor method, and semantic relatedness and normalization.

References

1. Golder, S., Huberman, B.A.: The structure of collaborative tagging systems. http://arxiv.org/abs/cs.DL/0508082 (Aug 2005)
2. Eisterlehner, F., Hotho, A., Jäschke, R.: Ecml pkdd discovery challenge 2009. http://www.kde.cs.uni-kassel.de/ws/dc09 (2009)
3. Sigurbjörnsson, B., van Zwol, R.: Flickr tag recommendation based on collective knowledge. In: WWW '08: Proceeding of the 17th international conference on World Wide Web, New York, NY, USA, ACM (2008) 327–336
4. Jäschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L., Stumme, G.: Tag recommendations in folksonomies. (2007) 506–514

5. Jschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L., Stumme, G.: Tag recommendations in social bookmarking systems. AI Communications **21**(4) (2008) 231–247
6. Symeonidis, P., Nanopoulos, A., Manolopoulos, Y.: Tag recommendations based on tensor dimensionality reduction. In: RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems, New York, NY, USA, ACM (2008) 43–50
7. Lipczak, M.: Tag recommendation for folksonomies oriented towards individual users. In: ECML PKDD Discovery Challenge 2008. (2008) 84–95
8. Tatu, M., Srikanth, M., D'Silva, T.: Rsdc 08: Tag recommendation using bookmark content. In: ECML PKDD Discovery Challenge 2008. (2008) 96–107
9. Andrews, N.O., Fox, E.A.: Recent developments in document clustering. Technical report, Computer Science, Virginia Tech (2007)
10. Kogan, J., Nicholas, C., Teboulle, M.: A Survey of Clustering Data Mining Techniques. In: Grouping Multidimensional Data. Springer (2006) 25–71
11. Begelman, G.: Automated tag clustering: Improving search and exploration in the tag space. In: In Proc. of the Collaborative Web Tagging Workshop at WWW06. (2006)
12. Shepitsen, A., Gemmell, J., Mobasher, B., Burke, R.: Personalized recommendation in social tagging systems using hierarchical clustering. In: RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems, New York, NY, USA, ACM (2008) 259–266
13. BibSonomy: Bibsonomy: A blue social bookmark and publication sharing system. http://www.bibsonomy.org/ (2009)
14. Junejo, K., Karim, A.: A robust discriminative term weighting based linear discriminant method for text classification. In: Data Mining, 2008. ICDM '08. Eighth IEEE International Conference on. (Dec. 2008) 323–332
15. CiteULike: Citeulike website. http://www.citeulike.org/ (2009)
16. TreeTagger: Treetagger – a language independent part-of-speech tagger. http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/ (2009)