A k-Nearest-Neighbour Method for Classifying Web Search Results with Data in Folksonomies

Ching-man Au Yeung Nicholas Gibbins Nigel Shadbolt
Intelligence, Agents, Multimedia Group
School of Electronics and Computer Science
University of Southampton
Southampton, SO17 1BJ, UK
{cmay06r,nmg,nrs}@ecs.soton.ac.uk

Abstract

Traditional Web search engines mostly adopt a keyword-based approach. When the keyword submitted by the user is ambiguous, search result usually consists of documents related to various meanings of the keyword, while the user is probably interested in only one of them. In this paper we attempt to provide a solution to this problem using a k-nearest-neighbour approach to classify documents returned by a search engine, by building classifiers using data collected from collaborative tagging systems. Experiments on search results returned by Google show that our method is able to classify the documents returned with high precision.

1 Introduction

In this age of information explosion, traditional search engines such as Google have greatly facilitated retrieval of information on the Web. However, results returned by these search engines are very often not as useful as the users have expected, usually because the semantics of the keywords are not taken into consideration. When a keyword with multiple meanings is used as a query, documents relevant to any of these meanings are all likely to be retrieved [18]. However, very often the user is probably interested only in one of the meanings or one of the contexts in which the keyword is used. The user will have to scan through the list of returned documents and single out those which are relevant. A method to tackle this problem is to identify words associated with the keyword in different contexts and classify the returned documents based on these contexts [3].

In recent years, collaborative tagging systems such as Delicious have become very popular among Web users.[1] In these systems, users are allowed to choose any keywords they like as tags to describe their favourite Web resources, resulting in a user-generated classification scheme now commonly known as a *folksonomy* [15]. Not only does a folksonomy provide metadata of Web resources in the form of tags, it also provides a lot of information about the associations between different tags when they are used together. We have shown in [1] that by performing clustering on documents tagged in a folksonomy, it is possible to extract the sets of tags related to the different contexts in which an ambiguous tag is used.

This paper discusses how such implicit semantics extracted from a folksonomy can be utilised to provide a possible solution to keyword ambiguity in Web search. We first perform cluster analysis on folksonomies based on the collective behaviour of the users, and generate labels for clusters based on tag co-occurrence. We then build k-nearest-neighbour classifiers to classify documents returned by a search engine. We evaluate our proposal by applying the method on search results returned by Google.

In the next section we discuss the motivation of our work. We present our proposed method in Section 3 and 4. Experimental results are presented in Section 5. Finally, we mention related work in Section 6 and give conclusions and future research directions in Section 7.

2 Problems of Web Search

Search engines are designed to present resources which satisfy information needs. However, the information needs of the users become less clear once they are translated into queries composed of individual keywords. This becomes a problem particularly when the keywords can be used to represent different concepts.

Consider the following example of searching for information about ‘bridge’ as a card game. If we submit a query

[1] http://delicious.com/

1	Contract bridge - Wikipedia, the free encyclopedia http://en.wikipedia.org/wiki/Contract_bridge
2	Bridge - Wikipedia, the free encyclopedia http://en.wikipedia.org/wiki/Bridge
3	Play bridge card game online http://www.bridgeclublive.com/
4	Bridge Travel http://direct.bridge-travel.co.uk/
5	River Kwai Bridge Travel http://www.riverkwaibridge.com/
6	Golden Gate Bridge Guide — Attraction Travel Guide http://www.worldtouristattractions.travel-guides.com/attraction/170/attraction_guide/North-America/Golden-Gate-Bridge.html
7	Bridge - Mainstreaming Gender Equality http://www.bridge.ids.ac.uk/
8	Bridge to Reuters http://www.bridge.com/
9	The Bridge SE1 - London venue for parties, gigs, films, conference http://www.thebridgese1.co.uk/
10	BRIDGE (Building Radio Frequency IDentification for the Global Environment) http:// www.bridge-project.eu/

Table 1. The top ten pages from Google when *bridge* is used as a query term.

with the keyword *bridge* to the search engine Google, a very diverse list of Web resources is returned (Table 1). While the first and the third pages returned are about the card game, it also contains pages about other meanings of the word *bridge*. For example, the second item is a page from Wikipedia describing bridges as architectural structures, and the sixth item is a page which contains information about the Golden Gate Bridge. There are also pages (e.g. 7th and 10th) which involve organisations or projects with the name 'Bridge' but are by no means related to any commonly used meanings of the word.

Two major problems can be observed. Firstly, extra effort is required from the user to go through the list and select those items which are relevant. Secondly, the presence of irrelevant pages reduces the number of relevant items that can be presented to the user at one time, especially when users tend to inspect only the first set of items returned [5, 13].

In addition, while users may add additional keywords to narrow down the search result, single-term queries are very common, representing $20 - 35\%$ of all queries according to several Web search studies [6]. Even though some search engines would provide suggestions of refining the search results, it will definitely be more beneficial to the user more if the search results are first classified into different categories before they are presented to the users.

3 Building Classifier from Folksonomies

The key idea in this paper is that popular folksonomies such as Delicious in which associations between different tags are embedded represent a valuable source of information for understand the different meanings of a term, which can be applied to solve the very problem mentioned in the previous section.[2] We have shown that [1] users in Delicious are usually consistent in using a certain tag to represent the same concept, and thus documents which are relevant to the same meaning of the tag are closely associated with each other by the same group of users. This suggests that, for a particular tag, clustering algorithms can be applied to extract groups of documents which correspond to different meanings of the tag. Our target is to build a k-nearest-neighbour classifier for an ambiguous tag based on the clusters of documents, with sets of related tags extracted as labels of the classes. This classifier can then be used to classify results returned by search engines.

A folksonomy generally consist of three sets of elements [9, 16], namely users, tags, and documents. Formally, we define a folksonomy as a tuple $\mathbf{F} = (U, T, D, A)$. U is a set of users, T is a set of tags, and D is a set of Web documents. $A \subseteq U \times T \times D$ is a set of annotations, each of which represents a tagging act in which a user $u \in U$ has assigned a tag $t \in T$ to a document $d \in D$. In this paper as we are interested in the semantics of a particular tag $t \in T$, we extract a bipartite graph B_t by restricting $\mathbf{F}$ to t: $B_t = \langle V, E \rangle$, where $V = U \cup D$ and $E = \{(u, d) | (u, t, d) \in A\}$.

This graph can be represented in matrix form: $\mathbf{A} = \{a_{ij}\}$, $a_{ij} = 1$ if there is an edge connecting user u_i and document d_j, and $a_{ij} = 0$ otherwise. We further fold this bipartite graph into a one-mode network of documents, represented by a similarity matrix, by performing matrix multiplication: $\mathbf{M} = \mathbf{A}^{\mathrm{T}}\mathbf{A}$. In this one mode network, an edge is weighed by the number of users who have assigned tag t to the documents represented by the vertices on the two ends of the edge.

From this network of documents, we should be able to identify groups of documents which correspond to the different meanings of the tag t. This can be done by applying clustering algorithms to the network represented by $\mathbf{M}$. We adopt the fast greedy algorithm for community discovery in networks proposed in [10], which optimises modularity [11] by connecting the two vertices at each step which result in the largest increase (or smallest decrease) of modularity. The algorithm is chosen because of its efficiency and good performance on many network clustering problems, and we would investigate the performance of other algorithms on this task in our future work.

If D_t is the set of documents which are assigned the tag t, the result of the clustering process is a set of clusters of documents: $\mathbf{X}_t = \{X_{t,1}, X_{t,2}, ..., X_{t,m}\}$ where $X_{t,1} \cup X_{t,2} \cup \cdots \cup X_{t,m} = D_t$. Finally, for each cluster $X_{t,i}$, we obtain a set $T_{t,i}$ of the tags used by the users on the documents in the cluster as its class label.

While each of these clusters should correspond to a sin-

[2] Since a word is referred to as a tag, a keyword or a term depending on the context in which it is being mentioned, we will use these terms interchangeably in the rest of this paper.

```
Algorithm 1: Building Classifier from Folksonomy
Input: Adjacency matrix $\mathbf{M}$ of the network of documents
Output: A set $\mathbf{C}$ of classes with a set of labels $\mathbf{T}$
begin
    // Document clustering;
    $\mathbf{C} \leftarrow FastGreedyCommunityDiscovery(\mathbf{M})$;
    $\mathbf{T} \leftarrow \{\}$;
    // Extract frequent tags;
    for $C_i \in \mathbf{C}$ do
        $T_i \leftarrow ExtractFrequentTags(C_i)$;
        $\mathbf{T} \leftarrow \mathbf{T} \cup \{T_i\}$;
    end
    // Merge similar clusters;
    $merged \leftarrow 1$;
    while $merged = 1$ do
        $merged \leftarrow 0$;
        for $T_i, T_j \in \mathbf{T}$ and $i \neq j$ do
            if $overlap(T_i, T_j) \geq \alpha$ then
                $C_{new} \leftarrow C_i \cup C_j$;
                $\mathbf{C} \leftarrow \mathbf{C} - \{C_i, C_j\}$;
                $\mathbf{C} \leftarrow \mathbf{C} \cup \{C_{new}\}$;
                $T_{new} \leftarrow ExtractFrequentTags(C_{new})$;
                $\mathbf{T} \leftarrow \mathbf{T} - \{T_i, T_j\}$;
                $\mathbf{T} \leftarrow \mathbf{T} \cup \{T_{new}\}$;
                $merged \leftarrow 1$;
            end
        end
    end
    return $\mathbf{C}$,$\mathbf{T}$;
end
```

gle meaning of the ambiguous tag t, it is possible that two or more of these sets are related to the same meaning, as it is normal to have more than one group of users referring to the same context. To eliminate such redundancy we combine two clusters if there is significant overlap between the two class labels with the help of the following function:

$$overlap(T_{t,i}, T_{t,j}) = \frac{|T_{t,i} \cap T_{t,j}|}{|T_{t,i} \cup T_{t,j}|} \qquad (1)$$

We introduce a threshold α, and merge the two sets of documents $X_{t,i}$ and $X_{t,j}$ when $overlap(T_{t,i}, T_{t,j}) \geq \alpha$. A new class label is generated for the new cluster by extracting the frequently used tags among the documents. Hence, the final result of this process is a set of classes of documents $\mathbf{C}_t = \{C_{t,1}, C_{t,2}, ..., C_{t,n}\}$ with class labels $\{T_{t,1}, T_{t,2}, ..., T_{t,n}\}$, where $n \leq m$. The whole process is summarised in Algorithm 1.

4 Web Search Result Classification

Given the set of classes, we build a k-nearest-neighbour classifier to classify search results returned by search engines. We assume that a set S_t of documents will be returned by a search engine when queried with a keyword t. Our target is to put the documents into the different classes obtained from the folksonomy clustering process, and we adopt a k-nearest-neighbour approach for this purpose.

We assume that each document $s_{t,j} \in S_t$ is characterised by a set $K_{t,j}$ of keywords, which could be keywords extracted from the document by removing stop-words, or tags assigned to the document in a collaborative tagging system. We can then calculate the similarity between such document and a document $d_{t,i}$ in D_t, the set of documents which have been assigned the tag t in a folksonomy and have been classified to one of the classes in $\mathbf{C}_t$. Let $J_{t,i}$ be the set of tags assigned to $d_{t,i}$. The similarity measure is given by:

$$Sim(K_{t,j}, J_{t,i}) = \frac{2 \times |K_{t,j} \cap J_{t,i}|}{|K_{t,j}| + |J_{t,i}|} \qquad (2)$$

Based on this similarity measure, a classification process can be summarised as follows. For each $s_{t,j} \in S_t$, we obtain the k most similar documents from the set D_t. The class of $s_{t,j}$ is decided by a majority vote of the classes of these k nearest neighbours. Documents belonging to the same class can then be grouped together before the search result is presented to the user.

It should be noted that while the classes correspond to different contexts in which t is used, the classes cannot be considered as exhaustive of all possible contexts. It is possible that a meaning of t is never referred to by the users in the system, and is therefore not identified by the clustering algorithm. It is possible that a document in the search result cannot be classified into any of the classes in $\mathbf{C}_t$. Therefore we introduce a threshold β here, where $0 \leq \beta \leq 1$. For a particular document $s_{t,j}$, if half of the k nearest neighbours have a similarity value less than β, the document will be assigned the class $C_{t,0}$ which represents unclassified documents. We represent this classification process as a function which maps a document to a class with respect to a tag: $F_A : S_t \times T \rightarrow \mathbf{C}_t$.

One may wonder whether such a computationally expensive method as the k-nearest-neighbour approach is necessarily when we have already identified related tags associated with different contexts. In fact, the use of k-nearest-neighbour approach does offer some advantages. In particular, we increase the chance of being able to classifying a document when we compare its keywords with a number of classified items instead of the limited number of extracted tags. This is because keywords appearing in documents returned by search engines can be very diverse, but frequently used tags in a collaborative tagging system can be limited. In summary, the use of k-nearest-neighbour classifier should offer a more robust solution.

5 Evaluation

To evaluate our proposed method, we apply the method to results returned by Google for eight terms which are ob-

Tag	Number of Documents	Number of Users
architecture	333	2,303
bridge	758	1,671
opera	253	3,031
sf	427	1,873
soap	326	4,971
tube	492	2,313
wine	431	9,652
xp	226	7,297

Table 2. Data collected from Delicious.

served to be ambiguous. These are *sf*, *tube*, *bridge*, *wine*, *architecture*, *xp*, *soap* and *opera*. These terms are selected because they are observed to have been used to represent multiple concepts in Delicious, and that search results returned by Google when using these terms in the query also consist of documents related to rather diverse topics.

5.1 Data Preparation

To build the k-nearest-neighbour classifiers, we collect data involving the eight tags from Delicious by using a crawler program. The dataset includes documents which have been assigned the tags and users who have used the tags on the documents. In the data collection process, we skip documents which are only tagged by one user because these will only become isolated nodes in the resultant networks. Table 2 summarises the statistics of the dataset.

For the testing dataset, we submit queries using each of the eight terms to Google and obtain the top 50 pages returned. We denote the set of documents retrieved for the term t by S_t. Each of the documents is characterised by a set of keywords, which consists of terms extracted from the texts of the documents after stop-words have been removed, and tags, if any, which are assigned to the documents by some users on Delicious.

5.2 Experiments

We first apply the clustering process to obtain a set of classes for each of the tags, with $\alpha = 0.2$. We extract 10 most frequently used tags from each resultant class of documents, which are treated as class labels. The result is shown in Table 3. The contexts are added by us for better understanding of the classes. It can be observed that the proposed algorithm performs well in revealing the different contexts in which the tags are used. The tags extracted are also closely related to the contexts they represent.

Next, we apply our classification method to search results we obtained from Google. In order to evaluate the performance of our proposed method, we have to establish a ground truth against which our result can be compared. Hence, we first manually classify the returned documents into the classes discovered in the clustering process. For example, for the tag *sf*, we have two classes: $C_{\text{sf},1}$ corresponds to 'San Francisco' and $C_{\text{sf},2}$ corresponds to 'science fiction'. We manually assign each of the documents returned by Google to one of these two classes. If a document cannot be classified to any of the available classes, we assign it the class $C_{t,0}$, which is reserved for unclassified documents. We represent this manual classification as a function which maps a document to a class with respect to a certain tag: $F_M : S_t \times T \rightarrow \mathbf{C}_t$.

Tag	Context	Tags Extracted
architecture	Software design	architecture, design, programming, toread, development, software, article, work, reference, web
	Buildings design	architecture, design, art, inspiration, cool, home, blog, house, culture, arquitectura
bridge	Design pattern	bridge, programming, development, library, code, ruby, tools, software, adobe, dev
	Card game	bridge, games, cards, game, imported, howto, conventions, card, bidding, online
	Computer networking	bridge, networking, linux, network, howto, software, sysadmin, firewall, virtualization, security
	Architecture	bridge, bridges, structures, engineering, science, physics, school, education, building, reference
opera	Web browser	opera, browser, web, software, javascript, tools, tips, internet , browsers, firefox
	Musical performance	opera, music, musique, classical, culture, art, travel ,nyc, musica, classic
sf	San Francisco	sf, sanfrancisco, bayarea, san, francisco, california, travel, events, art, san_francisco
	Science fiction	sf, scifi, fiction, books, sci-fi, literature, writing, sciencefiction, science, fantasy
soap	Cleaning agent	soap, soapmaking, diy, recipes, crafts, shopping, making, beauty, howto, craft
	Web services	soap, webservices, webservice, programming, web, xml, soa, development, wsdl, java
tube	YouTube videos	tube, youtube, video, funny, videos, fun, cool, music, feel.good, flash
	Vacuum tubes	tube, audio, electronics, diy, amplifier, amp, tubes, music, elect, guitar
	London underground	tube, london, underground, travel, transport, maps, map, uk, subway, reference
wine	Software application	wine, linux, ubuntu, howto, windows, software, tutorial, emulation, reference, games
	Beverage	wine, food, shopping, drink, reference, vino, cooking, alcohol, blog, news
xp	Windows XP	xp, windows, software, tools, pc, computer, tech, winxp, microsoft, windowsxp
	Extreme programming	xp, software, programming, process, methodology, development, agile, tech, extremeprogramming, extreme_programming

Table 3. Results of the clustering process.

Given the classifications F_A and F_M, it becomes possible to investigate the performance of our proposed method. We employ three different performance measures here, namely *precision*, *recall* and *coverage*. ***Precision*** measures the extent to which the documents are classified correctly. It is calculated by dividing the number of correctly classified documents by the total number of classified documents:

$$P = \frac{|\{d \in S_t | F_M(d,t) = F_A(d,t) \wedge F_A(d,t) \neq C_{t,0}\}|}{|\{d \in S_t | F_A(d,t) \neq C_{t,0}\}|} \quad (3)$$

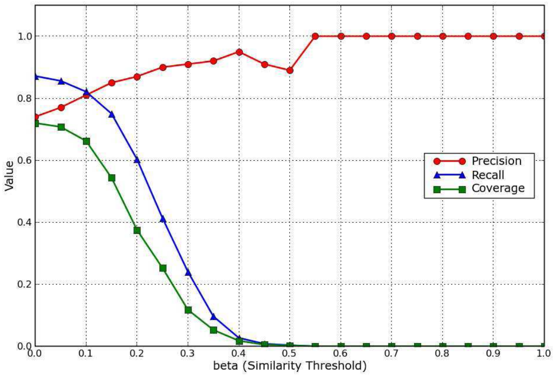

Figure 1. Precision, recall and coverage against different values of β.

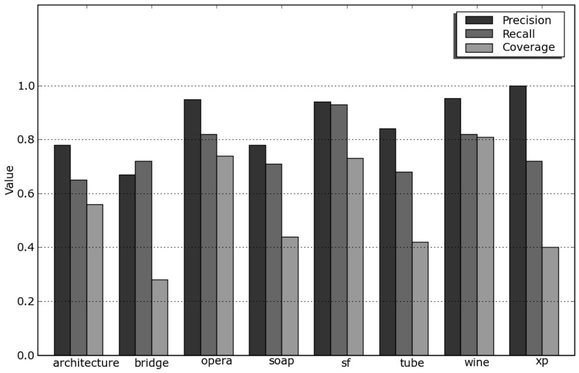

Figure 2. Precision, recall and coverage for different tags ($k = 11$, $\beta = 0.15$).

Note that we define precision to be one when no documents are classified. ***Recall*** measures the fraction of classifiable documents which the method is able to classify:

$$R = \frac{|\{d \in S_t | F_M(d,t) = F_A(d,t) \wedge F_M(d,t) \neq C_{t,0}\}|}{|\{d \in S_t | F_M(d,t) \neq C_{t,0}\}|} \tag{4}$$

By classifiable documents we refer to documents which should fall into any one of the contexts discovered in the tag meaning disambiguation phase. Finally, ***coverage*** measures how many documents can be classified given the total number of documents returned:

$$C = \frac{|\{d \in S_t | F_M(d,t) = F_A(d,t) \wedge F_M(d,t) \neq C_{t,0}\}|}{|S_t|} \tag{5}$$

We run our experiment using different values of k (number of nearest neighbour) and β (similarity threshold). We find that when $k \geq 5$, the value of k does not have significant effect on the performance of the classification, hence we choose $k = 11$ for the rest of our experiments. Figure 1 shows the performance of our method for different values of β. In addition, we also take a closer look at the performance of our proposed algorithm on different tags at $\beta = 0.15$ (Figure 2), a value chosen in the range where recall and coverage are high enough for the measure of precision to be meaningful.

5.3 Discussions

Figure 1 shows that as β becomes larger precision first increases, then decreases, and finally becomes flat at the value of 1.0. The increase of the precision in the first part is reasonable, because misclassified documents are eventually excluded due to their low similarity with the training data as β increases. However, as recall approaches zero, precision is greatly affected when one or two documents are misclassified, thus resulting in the fluctuation. When β becomes so large that no documents are classified and precision becomes one towards the end as we have defined. On the other hand, our experiment shows that recall and coverage decrease as β increases. This is because more documents will be considered as unclassified when β increases. It should be note that the calculations of recall and coverage are only different from each other in the denominators which are constants in both cases. Hence it is not surprising to see that they have similar declining curves. Their significance is better revealed when we take a closer look at the performance of our method for each of the tags.

Figure 2 shows the performance measures for different tags at $\beta = 0.15$. Our proposed method gives satisfying results as judged from the precision of the classifications, ranging from 67% (*bridge*) to 100% (*xp*). This suggests that the clustering process performed on the folksonomy is able to place documents into meaningful clusters, such that these documents provide an accurate basis for the k-nearest-neighbour classification process. Our investigation into cases with relatively low precision (e.g. *architecture*, *bridge* and *soap*) reveals that while misclassified documents contain keywords which provide enough information about their contexts, they are not always the same as the tags assigned to documents corresponding to the same contexts in Delicious. For example, documents about bridges as an architectural structure returned by Google contain keywords such as *river* and *stream*, however these keywords do not appear as tags in Delicious on related documents. This suggests that users in Delicious do assign tags they think suitable but do not appear in the content of the documents.

Recall and coverage are relatively lower than precision in all cases. Low recall means that the algorithm is unable to classify many documents which are actually related to one of the contexts discovered in the clustering process. This is probably due to the same reason mentioned above which causes low precision in some cases. Documents in Delicious and those returned by Google are characterised by different keywords of the same context in some cases, rendering a certain amount of documents unclassifiable.

The measure of coverage has the greatest range among the three, with values between 28% and 81%. While low coverage is partly due to low recall in some cases, the result also suggests that the clustering process do not always return all the contexts in which the tag in question is used. For example, the common usage of *tube* to refer to a hollow, long and circular structure is not found in the list of contexts discovered, and because of this Websites selling different kinds of tubes become unclassifiable. However, further study of the documents reveal that the major reason of low recall is that the documents are actually not related to any commonly known meanings of the words. For example, search result for *bridge* contains information pages about entertainment venues or organisations whose names contain the word *bridge*, and such items account for more than half of the 50 documents we have examined. Judging from this observation, we believe that a low coverage is not as undesirable as it first seems, because our proposed method actually helps to filter out documents which are not semantically related to the query term.

In summary, our proposed method for Web search result classification is able to classify documents with high precision based on the implicit semantics extracted from a collaborative tagging system. Clearly, from the discussions about the experimental results we believe there are several ways in which the proposed method can be improved. In particular, how we can build a more comprehensive classifier – both in terms of the keywords characterising the documents and of the contexts in which the ambiguous terms are used – is a major issue which requires further investigation.

6 Related Works

To the best of our knowledge, there have been no studies which make use of user-contributed annotations to classify Web search results, although in the literature different methods have been used to discriminate word meanings. These include the use of dictionaries or thesauri (e.g. [8]). Our work is similar in part to studies which employ lexical co-occurrence to discover different senses of an ambiguous word. For example, Schütze and Pedersen [12] construct a term vector for each word representing word similarity derived from lexical co-occurrence. The vectors are then combined to form context vectors which are clustered to represent different senses of ambiguous words.

Our work is also similar in principle to studies which apply machine learning approaches to Web search results. This problem has been discussed quite extensively in terms of both supervised and unsupervised learning in the literature (e.g. [2, 14, 18]). It is also addressed by commercial systems such as Vivisimo [7].[3] Many existing methods extract keywords from documents and calculate their similarity based on the keywords to obtain a set of clusters. Our approach differs from these techniques in that instead of performing clustering directly on documents returned by search engines, we obtain a set of classes by clustering folksonomy data to aid classification of documents. We believe our proposed method is better as it is more focused in terms of the contexts discovered, while existing document clustering techniques might result in clusters which are not necessarily meaningful to the users.

While there have been no studies which directly address the problem of tag ambiguity, disambiguation of tags can be observed as a by-product in some research work which focuses on tag clustering. For example, latent semantic analysis is applied to study the co-occurrence of tags in [16]. Zhou et al. [19] also report that, in building a tag hierarchy by clustering, tags with multiple meanings are found to appear in different branches of the resulting hierarchy. In addition, collaborative tagging is also used to improve Web search in general, such as by providing a better ranking of the search results [4, 17]. In contrast to these prior studies, our work directly addresses the problem of tag ambiguity, proposes a feasible solution and studies how the extracted semantics of tags can be applied to novel applications.

7 Conclusions and Future Work

This paper presents a novel idea of how implicit semantics in folksonomies can be extracted to solve the problem of keyword ambiguity in Web search. By collecting data from a folksonomy we see that some unconventional meanings of a word can be discovered. For example *tube* is found to be used to refer to video-sharing Websites, and *bridge* is used to refer to a kind of design pattern in programming. These meanings are either new or are of specific domains, and they may not be available in dictionaries or thesauruses.

We plan to investigate how the performance of our proposed method can be improved. We will look into various methods to increase the comprehensiveness of the classifiers by exploring associations between tags in folksonomies to identify more related tags and by considering the possibility of enriching the discovered contexts by, for example, combining data from several folksonomies. In addition, we will investigate whether other clustering methods will allow us to change the granularity of the clusters so that we can adjust the specificity of the contexts discovered.

[3] The public version of Vivismo's Web search engine, Clusty, can be found at http://clusty.com/.

References

[1] Ching Man Au Yeung, Nicholas Gibbins, and Nigel Shadbolt. Understanding the semantics of ambiguous tags in folksonomies. In Liming Chen et al., editor, *Proc. of the First International Workshop on Emergent Semantics and Ontology Evolution, ESOE 2007, co-located with ISWC 2007 + ASWC 2007, Busan, Korea, November 12, 2007*, pages 108–121, 2007.

[2] Douglass R. Cutting, David R. Karger, Jan O. Pedersen, and John W. Tukey. Scatter/gather: a cluster-based approach to browsing large document collections. In *SIGIR '92: Proc. of the 15th annual international ACM SIGIR conference on Research and development in information retrieval, Copenhagen, Denmark*, pages 318–329. ACM, 1992.

[3] Francis Heylighen. Mining associative meanings from the web: from word disambiguation to the global brain. In *Proceedings of Trends in Special Language and Language Technology*, pages 15–44. Standaard Publishers, 2001.

[4] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Information retrieval in folksonomies: Search and ranking. In York Sure and John Domingue, editors, *The Semantic Web: Research and Applications*, volume 4011 of *LNCS*, pages 411–426. Springer, June 2006.

[5] iProspect. Search engine user behaviour study. April 2006.

[6] Bernard J. Jansen, Amanda Spink, and Tefko Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. *Inf. Process. Manage.*, 36(2):207–227, 2000.

[7] Sherry Koshman, Amanda Spink, and Bernard J. Jansen. Web searching on the vivisimo search engine. *Journal of the American Society for Information Science and Technology*, 57(14):1875–1887, 2006.

[8] R. Krovetz and W. B. Croft. Word sense disambiguation using machine-readable dictionaries. *SIGIR Forum*, 23(SI):127–136, 1989.

[9] Peter Mika. Ontologies are us: A unified model of social networks and semantics. *Journal of Web Semantics*, 5(1):5–15, 2007.

[10] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69:066133, 2004.

[11] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004.

[12] Hinrich Schütze and Jan O. Pedersen. Information retrieval based on word senses. In *Proc. of the 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, 1995.

[13] Craig Silverstein, Hannes Marais, Monika Henzinger, and Michael Moricz. Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1):6–12, 1999.

[14] Jerzy Stefanowski and Dawid Weiss. Carrot^2 and language properties in web search results clustering. In Ernestina Menasalvas Ruiz, Javier Segovia, and Piotr S. Szczepaniak, editors, *Proc. of First International Atlantic Web Intelligence Conference, AWIC 2003, Madrid, Spain, May 5-6, 2003*, volume 2663 of *LNCS*, pages 240–249. Springer, 2003.

[15] Thomas Vander Wal. Folksonomy definition and wikipedia. *http://www.vanderwal.net/random/ entrysel.php?blog=1750*, November 2, 2005. Retrieved on 13 Feb 2008.

[16] Xian Wu, Lei Zhang, and Yong Yu. Exploring social annotations for the semantic web. In *WWW '06: Proc. of the 15th international conference on World Wide Web, Edinburgh, Scotland, May 23-26, 2006*, pages 417–426. ACM Press, 2006.

[17] Yusuke Yanbe, Adam Jatowt, Satoshi Nakamura, and Katsumi Tanaka. Can social bookmarking enhance search in the web? In *Proc. of the 7th ACM/IEEE-CS joint conference on Digital libraries*, pages 107–116, New York, NY, USA, 2007. ACM.

[18] Hua-Jun Zeng, Qi-Cai He, Zheng Chen, Wei-Ying Ma, and Jinwen Ma. Learning to cluster web search results. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 210–217, New York, NY, USA, 2004. ACM.

[19] Mianwei Zhou, Shenghua Bao, Xian Wu, and Yong Yu. An unsupervised model for exploring hierarchical semantics from social annotations. In Karl Aberer et al., editor, *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007*, volume 4825 of *LNCS*, pages 680–693. Springer, 2007.