Applying community detection methods to cluster tags in multimedia search results

Teresa Bracamonte, Aidan Hogan, Barbara Poblete
Center for Semantic Web Research
Department of Computer Science, University of Chile
{tbracamo, ahogan, bpoblete}@dcc.uchile.cl

***Abstract*—Multimedia searches often return items that can be categorized into several "topics", allowing users to disambiguate and explore answers more efficiently. In this paper we investigate methods for clustering tags associated with multimedia search results, where each resulting cluster represents a topic computed online for that particular search. We specifically investigate the applicability of *community detection algorithms* to the tag graph induced from the search results. This type of approach allows us to exploit tag similarity and create ad-hoc topics for each search, without specify the number and sizes of clusters a priori.**

In this work we experiment with well-known algorithms in this field and propose two new methods based on *adaptive island cuts*. Using the *Social20* dataset (a collection gathered from Flickr) we evaluate several community detection methods, with quantitative analysis of each algorithm in terms of the relative number of communities (which we interpret as topics) that they produce and their sizes, as well as qualitative analysis of topics per human judgement. Our evaluation shows that it is possible to extract ad-hoc topics for search results using community detection, but that different community detection methods produce very different results. In particular, our proposed methods produce more compact and less noisy clusters as well as less relative recall when compared to methods that produce much larger clusters.

***Index Terms*—tag clustering; multimedia retrieval; community detection; topic detection**

I. Introduction

A main concern in information retrieval on the Web, and in particular for multimedia search engines, is how to improve user interaction with search results. A common mechanism used in commercial image search engines (and other types of multimedia searches) to help users quickly make sense of results is to apply *clustering* in order to organize similar results into groups [1], [2]. For example, results can be shown in groups with *similar content* (e.g., images with similar color schemes) or with a *similar topic*. In this paper we focus on topic-based clustering, which can help users understand search results by displaying multiple interpretations of ambiguous textual queries (e.g., the query "*boxers*" may refer to dogs, fighters, or underwear), or different aspects of the same interpretation (e.g., for "*boxer dogs*": puppies, faces, etc.).

As opposed to clustering by content, which relies on audio-visual features (e.g., [3], [4], [5]), topic similarity clustering traditionally relies on textual features, such as the text that surrounds an image in a Web page, or human annotations (*tags*) associated to multimedia objects. Many of the methods that exploit surrounding text extend LDA [6] (more recent works are described in [7], [8]). However, it is common to find multimedia repositories, such as Flickr[1] and Last.fm[2], for which only user-generated tags are usually available.

We focus on topic clustering using tags. Specifically, we address the problem of obtaining topics *using only the tag information in the multimedia search results for a given query*. We aim for a methodology that allows us to provide ad-hoc topics for any query in an online fashion without prior knowledge of the multimedia domain or access to the complete multimedia collection. This has the advantage of allowing lightweight distributed computation of topics (i.e., client-side) while at the same time addressing long-tail queries that cannot be precomputed on the server-side. Our methodology might be used to aggregate results from multiple search engines, with different types of multimedia, as long as results are tagged.

For this topical clustering, we research the application of well-known *community detection* algorithms on a tag-graph representation of the search results, with the hypothesis that densely connected communities of tags in the *tag co-occurrence graph* (a graph where tags are nodes and edges indicate two tags co-occurring on an item) correspond to topical clusters [9]. We thus propose a general framework for discovering topics on multimedia search results. Our framework leverages existing approaches for community detection. To address limitations discovered from the initial evaluation, we also propose two novel methods for community extraction. These methods require no prior knowledge other than tagged search results; likewise they do not try to "interpret" tags and hence our methods are language-agnostic. Taking the Social20 dataset, we compare the community detection techniques quantitatively in terms of the number and sizes of clusters produced, and qualitatively in terms of how related the terms are in each cluster according to human judgement.

II. Background

We provide a summary of the background literature on multimedia search results clustering, tag-based clustering and community detection algorithms.

Multimedia search results clustering: The deluge of multimedia data on the Web has raised the need for multimedia search engines to provide mechanisms to enhance user interaction with search results. Several studies have shown that topical clustering of documents help users to quickly make sense

[1]http://flickr.com
[2]http://last.fm

of the search results. Some approaches use text from Web pages [2], comments from multimedia sharing platforms [10], and metadata associated to multimedia objects [11].

Given that there is a large amount of unannotated multimedia objects on the Web, many works focus on how to automatically assign tags to unannotated multimedia [27] as well as how to refine tags [28] by aggregating user-generated content. Li et al. [29] survey related work on this topic.

Besides text-based multimedia search clustering, hybrid techniques fuse multimedia content and context data for building clusters. For example, adding audio-visual features (e.g., color distribution, texture) can boost Web content and structure (e.g., image surrounding text, image URL, hyperlinks) [1], or user-generated metadata (e.g., title, tags, description) associated to multimedia content [12]. Nevertheless, these approaches are not always scalable due to the high computation cost at the audio-visual feature extraction stage. Applications based on purely text-based analysis are hence considered relevant for modern applications. Moreno and Dias [13] study the application of text-based multimedia search clustering on the context of Web search via mobile devices.

Tag-based multimedia clustering: Multimedia resources often lack descriptive text; instead, tags have emerged as a convenient way to describe and (partially) organize multimedia objects. Although tag-based clustering could be considered as text-based clustering, when considering tags, we do not have the notion of word position or proximity (tags do not follow a relevance order), nor do we have word frequencies *within* a document (tags are assigned at most once). Thus, instead of applying text-based topic extraction techniques, the typical approach is to apply graph-based techniques.

Most of the literature on applying graph-based approaches to extract structure from tag co-occurrences focuses on building *taxonomies*. Such taxonomies allow users to browse tags and tagged resources in an intuitive manner. Strohmaier et al. [14] survey the most common taxonomy-induction approaches. Such hierarchical approaches are not always suitable for representing relationships between tags because tagging is inherently *flat*, with no explicit information on how broad or narrow tags are. Some proposals address this issue by using external hierarchical sources of knowledge, such as WordNet [15], or shallow taxonomies arising from user-specified collections [16], or domain ontologies [17]. However, domain-independent datasets often only cover general tags but not proper nouns or acronyms, while domain-dependent datasets are often not available.

In this work, we consider the induction of a topical taxonomy as orthogonal to the goal of clustering, wherein we wish to group related tags and tagged resources without worrying which are broader than which. We view the problem of clustering as finding closely-knit neighborhoods or *communities* in the flat tag co-occurrence graph.

Community Detection: A *community* is a densely connected sub-graph within a graph. *Community detection* is a graph partitioning problem whose goal is to identify the "densest" possible set of communities that form a partition. A naive approach would be to apply a standard *minimum-cut* method that partitions a graph while minimizing the cost of the cut (e.g., the number of edges cut). However, minimum-cut methods generally force a fixed number of cuts or partitions rather than identifying "natural" dense subgraphs. Instead, one of the main goals of community detection is to identify the best communities, irrespective of their number; this is very useful for our scenario since it avoids having to provide, a priori, a desired number of clusters. Fortunato [18] and Papadopoulos et al. [9] describe a variety of metrics and algorithms that address the community detection problem.

Many community detection methods focus on optimizing a general metric. *Modularity*, the most common metric, is the ratio of all edges in the graph that fall within the communities, minus what would be expected in a graph with the same number of vertexes and edges but where edges are assigned randomly. Unfortunately, finding the optimal community configuration is intractable; hence algorithms employ approximations such as local modularity measures [19], spectral analysis [20], etc. Another issue is the *resolution limit* of modularity. In large graphs, even a single edge between two communities is seen as an "unlikely event", causing small communities to be merged into a few very large communities.

Besides optimizing modularity, authors have explored other options for community detection, e.g., applying a top-down clustering approach [21], or analyzing the structure of the graph [22], or how information flows within it [23], [24].

Novelty: We apply a clustering of nodes in the tag co-occurrence graph without inducing a taxonomy and without needing external knowledge. Given the extensive body of work in community detection, and the lack of need to specify a number of clusters or cuts, we see it as natural to research applying such techniques to our clustering problem. We thus evaluate, both quantitatively and qualitatively, the clusters produced by various community detection techniques in terms of corresponding to topics recognizable to users. Based on initial results, we found that existing community detection techniques tend to return too few large clusters, or too many small clusters, etc.; hence we propose two novel methods based on island cuts [25], [26].

III. Framework for Topic Detection

We introduce a framework to detect semantically relevant groups of tags ("topics") associated to queries encoding user-information needs. We are motivated by the use-case of performing an online clustering of heterogeneous multimedia search results using only the tags in the results.

Figure 1 shows the proposed framework, which consists of the following four main stages:

1) *Retrieval of multimedia resources based on a specific query:* Search results are the starting point for our framework. In the interest of generality, we assume as input, a collection of items associated with a set of tags, over which we perform clustering using tag co-occurrence information. Since we reduce the representation of multimedia content

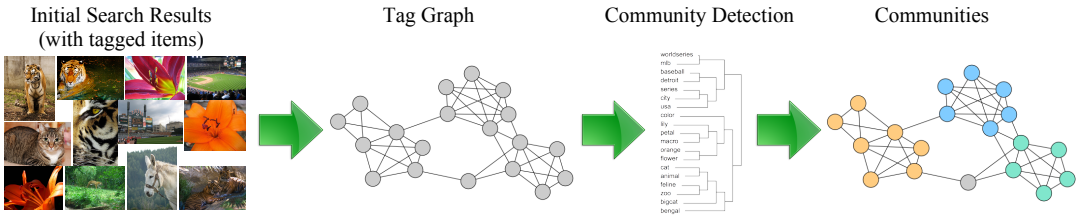

Fig. 1. Framework for topic detection.

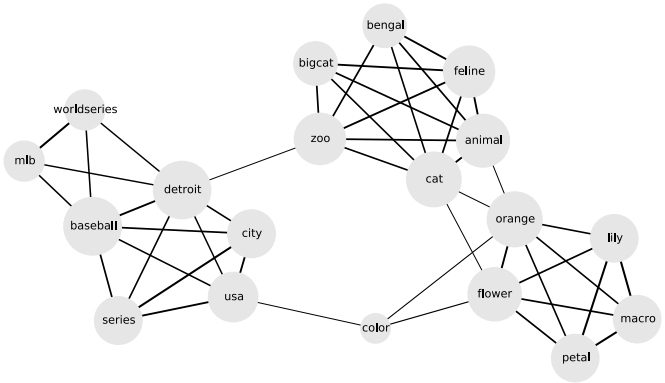

Fig. 2. Partial Tag Graph extracted from search results for query *"tiger"*.

to tags, in theory, our framework is able to combine different types of multimedia resources.

2) *Construction of the Tag Co-Occurrence Graph (Tag Graph):* The tag graph is the key structure over which our framework works. Given a finite bag[3] of resources R, where each resource is represented as a set of tags, we define the *tag co-occurrence graph* as $G_\lambda = (V, E, \lambda)$, where $V = \cup_{r \in R} r$ are the vertexes, $E = \{(v, v') \mid \exists r \in R$ such that $v \in r, v' \in r$ and $v \neq v'\}$ are the edges, and where $\lambda : E \to \mathbb{R}$ is a weighting function that labels each edge in E with a real value.
The simplest weighting scheme for λ counts the number of co-occurrences for a pair. Nevertheless, cardinality-based weights might be sensitive to the number of resources considered; thus we propose to use a more robust weighting scheme such as *structural similarity* [22]:

$$\mathrm{sim}(v, v') = \frac{\#(\mathrm{nv}(v) \cap \mathrm{nv}(v'))}{\sqrt{\#\mathrm{nv}(v) \times \#\mathrm{nv}(v')}}$$

where v and v' are nodes, $\mathrm{n}(v) = \{v'' \mid (v, v'') \in E\}$ are the neighbors of v in the undirected graph, $\mathrm{nv}(v) = \mathrm{n}(v) \cup \{v\}$ includes v and $\#S$ denotes the cardinality of the set S.

3) *Tag Clustering based on Community Detection Algorithms:* We employ community detection algorithms to avoid having to specify the number of clusters a priori. Under this approach, no additional knowledge about the multimedia resources is necessary to detect relevant groups of tags (other than tag co-occurrence). Initial empirical experiences revealed that using existing community detection algorithms [9] to cluster search results from the Flickr image search engine often led to unintuitive topics (e.g., we found clusters that were too large and grouped unrelated terms). We hence also propose two new community detection algorithms based on island cuts [25].

4) *Topic representation using tags:* Ideally, it would not be necessary to apply any additional process to the output of the community detection algorithms. However, for those algorithms that return large communities, we would need to apply some ranking technique in order to get a manageable subset of tags. The simplest approach to rank tags inside a community would be sorting them by frequency or degree.

The main characteristics of our framework are:

– *Multimedia-type independence*: We do not employ content-based features in the tag graph construction process. Our model could potentially discover topics across different types of multimedia resources in a transparent fashion.
– *Tag and topic independence*: Our framework does not require any training data, it is not fixed to domain or language, nor to a limited or fixed number of topics.
– *Query specific*: Tag graphs are built with respect to a specific query, which helps disambiguate (non-query) polysemous tags; e.g., the tag `jaguar` appearing on results for a query "*zoo*" will (likely) only refer to the cat, not the car[4].
– *Detection of concepts online*: Given adequate physical infrastructure and optimized community detection algorithms, it is possible to perform multimedia topic detection in an online fashion (e.g., on the client side).

IV. Graph Clustering Methods

Our novel graph clustering methods are based on the notion of *islands* first introduced by Zaveršnik & Batagelj [26].

Islands: An island is a subgraph that is maximal in its neighborhood for a given property of the graph [25], [26]. Zaveršnik & Batagelj consider two types of islands: *vertex islands* and *edge islands*. An island is defined relative to some vertex (or edge) property p, where no external neighbor of the island has a higher value for p than any vertex (or edge) in the island. Also, if no such neighbor has an equal p value to a vertex (or edge) within the island, that island is *regular*.

Islands are built following a "greedy" algorithm that initializes islands with the vertexes (or edges) that have the highest values for the given property, and then enlarge and/or combine those islands by traversing the graph from these starting points to include their neighbors. This process is similar to building

[3]A bag is a set that allows duplicates. We need to consider a bag for the weighted version since, e.g., two images may have the exact same set of tags.

[4]Of course, if the query is "*jaguar*", then the purpose of our methods is to identify the different senses, such as *cat* and *car*.

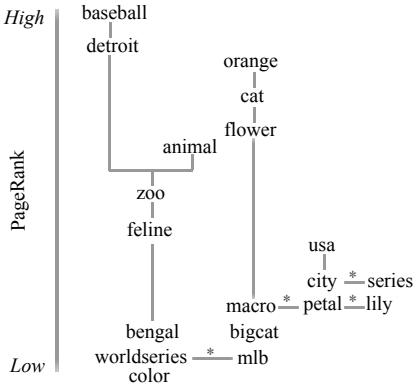

Fig. 3. Hierarchy used to identify vertex-islands ('$*$' indicates an edge affecting regularity)

a spanning tree, except that it records the order in which the vertexes (or edges) are added to the solution.

Identifying islands: We follow the high-level algorithm(s) for extracting islands proposed by Zaveršnik & Batagelj [25], [26].

Island hierarchy based on vertexes: We first initialize a hierarchy $H = (V_H, E_H)$. Given G, we iterate over its vertexes V in descending order of their value for the given vertex property p, where for each $v \in V$, we first add it to V_H. Next, we connect v to all *ports* v' (the last vertex added to an island) of existing islands in H that are neighbors of v in G: we add the edge (v_i, v_i') to E_H, where v now replaces each such v' as a port for a new larger island. For a vertex-island to be *regular*, it cannot have an edge in E_H that directly connect two nodes having the same value for p.

Example 1. Assume we take PageRank [30] as the vertex property p. If we compute the PageRank of every vertex in the tag graph in Figure 2 and sort them (descending) we obtain:

$\mathsf{V} = (\texttt{baseball}, \texttt{detroit}, \texttt{orange}, \texttt{cat}, \ldots, \texttt{color})$

We start with $H = (V_H, E_H)$ blank. Iterating over V, we first add `baseball` to V_H. Next, we add `detroit`, which is a neighbor of `baseball` in G. Since `baseball` is a port in H, we add the directed edge (`baseball`,`detroit`) to E_H. Next, we add `orange` but do not connect it to anything since it has no neighbor already in H. Fig. 3 shows the final hierarchy. □

Island hierarchy based on edges: We start with $H = (V_H, E_H)$ where V_H contains sets of vertexes representing islands, and $E_H \subseteq V_H \times V_H$. We initialize V_H with all singleton vertexes and E_H as empty. Given G, we iterate over its edges E in descending order with respect to a given property p. For each edge, we retrieve the largest islands in V_H containing both nodes. If they are not the same, we create a new island which is a union of the two and add a directed edge from the new island to the two old subislands. Once all edges are analyzed, H is a tree where all nodes in V_H are edge-islands and a subset of non-redundant edges in E_H represent a subisland relationship. For an edge-island to be *regular*, it cannot have an incoming edge in H that was derived from an edge with the same weight.

Example 2. Let's assume we sort the edges E in the tag graph of Fig. 2 in descending order with respect to the structural similarity value, giving us the list E:
$\mathsf{E} = \{(\texttt{worldseries}, \texttt{mlb}), (\texttt{city}, \texttt{series}), (\texttt{petal}, \texttt{lily}), \ldots, (\texttt{color}, \texttt{usa})\}$
This time, $H = (V_H, E_H)$ where V_H contains sets of vertexes representing islands, and $E_H \subseteq V_H \times V_H$. We take the first edge (`worldseries`, `mlb`) and retrieve the largest islands in V_H containing both nodes. Since they belong to different islands, we create a new island that contains the two old subislands. We repeat the process until all edges have been processed. We show the final hierarchy in Fig. 4. □

Adaptive Island Cuts (*AIC*): To avoid single-vertex islands, and single-island hierarchies, Zaveršnik & Batagelj [25], [26] select valid islands based on lower and upper bounds $[k, K]$. We aim to be more flexible with the size of islands, so we set a threshold on the *graph density* for what we consider to be valid islands. We state that the closer to a clique an island is, and the larger the island is, the better the island is. We capture the trade-off between the density of the island and its size using a *density threshold*:

$$\delta(x) = \frac{x(x-1)}{2} \cdot \max\left(\log_2\left(\frac{x+k}{x}\right), t\right)$$

where x is the number of vertexes in the island, k is the minimum number of vertexes allowed for an island ($k = 3$), and t is a fixed lower bound. The left term of the product is the number of edges in a clique with x vertexes (excluding loops). Assuming $t = 0$, the rightmost term is a logarithmically decaying ratio on the number of vertexes in the range $(0, 1]$. However, after our initial tests returned large islands with low density, we added t as a practical compromise: it offers a parameterizable fixed lower bound on the ratio to ensure a minimal density for larger islands. Also, in order to avoid *outliers* [22], we additionally require that all vertexes in the island $G' = (V', E')$ have at least $\log_2(\#V')$ edges for it to be considered a community.

To sum up, we consider an island $G' = (V', E')$ to be a community iff: (1) the island is regular, (2) $\#V' > k$ (where $k = 3$), (3) $\#E' \geq \delta(\#V')$, AND (4) there does not exist a $v' \in V'$ such that $\#n(v') < \log_2(\#V')$. Note that (1) and (2) correspond to the criteria proposed by Zaveršnik & Batagelj [25], [26], whereas we add (3) and (4) to avoid the need for a fixed upper bound K and to correspond with our intuition of a community. We start the selection of communities from the most general island (with all vertexes) and visit subislands, checking that the criteria are met.

Example 3. We return to the hierarchy in Fig. 4, where the final communities are highlighted with shaded boxes. In terms of how these communities were computed, we analyze every island in the hierarchy starting at the top (which corresponds to the bottom of the figure). We first evaluate the island that corresponds to the full graph. We assess the graph using all conditions previously defined, where **Th.** indicates the threshold value, **Ob.** the observed value and **Pass?** whether or not the condition is satisfied.

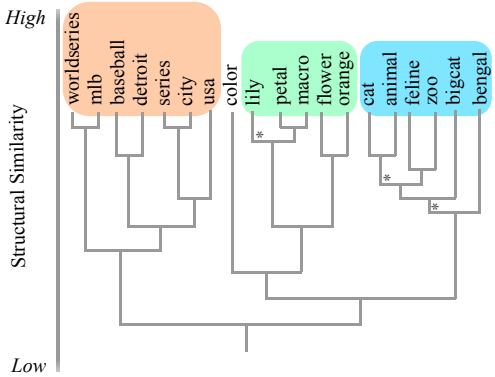

Fig. 4. Hierarchy for edge islands with final communities shaded ('$*$' indicates an edge affecting regularity)

Since the full graph does not satisfy all conditions (it fails criteria (3) and (4)), we proceed to analyze its subislands independently. We first analyze the left subisland with vertexes: $V' = \{\texttt{w.series}, \texttt{mlb}, \texttt{baseball}, \texttt{detroit}, \texttt{series}, \texttt{city}, \texttt{usa}\}$

	Th.	Ob.	Pass?
REGULAR?	✓	✓	✓
MIN VERTEXES (k)	3.0	7	✓
MIN EDGES ($\delta(\#V')$)	10.8	15	✓
MIN CONNECTIVITY ($\log_2(\#V')$)	2.8	3	✓

This subisland satisfies all conditions and is accepted as a community. Fig. 4 highlights the final set of communities. □

V. EVALUATION

The goal of this evaluation is to find out if community detection algorithms produce cohesive semantic topics and to measure, as well, the quality of the topics produced by each algorithm. Cohesiveness in this case is complex to measure, as it must evaluate whether the tags in a particular cluster are semantically related. This is a subjective task and thus we seek human judgement. In addition, the complete evaluation dataset is huge, including thousands of topics (clusters). Thus, we address this issue by sampling the resulting topics and only evaluating the resulting samples. The evaluation data is available for download at http://dcc.uchile.cl/~tbracamo/communities/.

Evaluation dataset: We use *Social20* [27], which is a public dataset containing metadata for 20,000 images from Flickr that match a set of 20 keyword queries (1,000 results per query). Each image has a unique ID, an owner ID, and a list of tags. All tags have been lemmatized. In addition, we remove tags that have been used by only one owner.

Algorithm settings: Given that our evaluation relies on human judgement, we limit our evaluation to two settings, which were chosen based on preliminary experiments.

AIC-EDGE: With structural similarity as edge property; and $k = 3$, $t = 0.33$ for the density function.
AIC-VERTEX: With PageRank as vertex property[5]; and $k = 3$, $t = 0.33$ for the density function.

[5]We use structural similarity as transitional weights.

Algorithms for comparison: We compare against the following community detection algorithms [9] using the same input graph as that of the AIC-* configurations.

EIGENVECTOR [20] (abbr. "EIGENVEC") aims to maximize modularity.
INFOMAP [24] uses random walks to emulate information flow in a network.
LABEL PROPAGATION [23] (abbr. "LBLPROP") uses a recursive voting scheme until it reaches a fixpoint.
MULTILEVEL [19] (abbr. "MULTILVL) is similar to a hierarchical agglomerative clustering technique.
SCAN [22] (abbr. "MSCAN") is a variation of the clustering algorithm DBSCAN [31] for graphs.

Community sizes: In Table I, we present statistics for the communities identified for the tag graphs by each algorithm. After applying the community detection algorithms in our dataset, we see a split of the algorithms into two groups based on an order of magnitude difference in the average community size computed: LBLPROP, EIGENVEC and MULTILVL produce much larger/fewer communities than AIC-*, INFOMAP or MSCAN. In fact, the former algorithms tended to produce one "super-community" with the vast majority of tags, and a few other small communities.

TABLE I
STATISTICS ABOUT COMMUNITIES COMPUTED BY DIFFERENT METHODS ACROSS ALL 20 QUERIES

Method	Total	Community Size			
		Max	Min	Avg	S.Dev
AIC-EDGE	649	216	3	7.33	17.94
AIC-VERTEX	533	142	3	6.47	15.27
EIGENVEC	157	542	4	81.91	107.43
INFOMAP	906	79	4	6.54	17.93
LBLPROP	123	1446	4	104.58	4.87
MULTILVL	189	393	4	68.17	82.80
MSCAN	1,269	166	4	7.31	8.24

User Study Design: We design a user evaluation to measure how well community detection algorithms identify topics from the tag graphs in query results. Each user is presented with a set of tags from a cluster that has been created using one of the community detection algorithms. Users are given the option to remove terms that they do not understand. The user is then asked to choose the largest subset of tags that they (subjectively) consider semantically related. In addition, the user can state that they do not find any of the terms to be related. We randomize the order in which tags are shown to remove any bias due to position. Fig. 5 shows an example.

As we mentioned earlier, some clusters contain over a hundred tags, and the complete set of clusters produced by all algorithms is too large for human evaluation. To mitigate this problem, we designed the following sampling method that we applied to the results of all algorithms: (1) For each query we identify the subset of tags assigned to a cluster for each of the algorithms (not all the tags are assigned to clusters). We refer to these tags as the *seed set* for the query. In addition, we

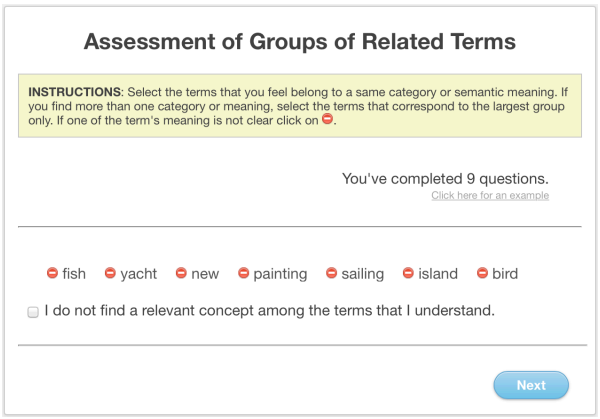

Fig. 5. Human Intelligent Task (HIT) design.

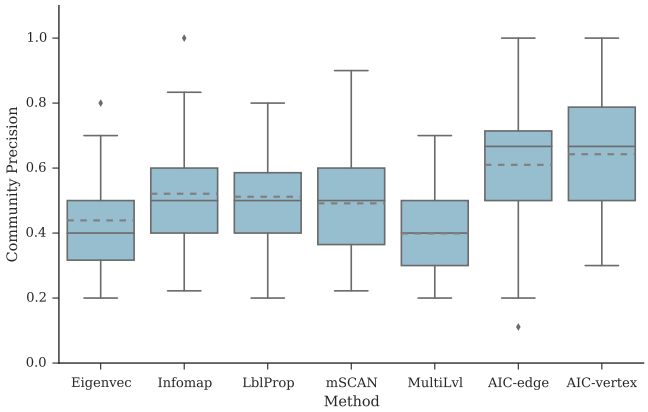

Fig. 6. Comparison of majority-based precision for the methods considering only sampled communities for which an agreement of $\alpha > 0.4$ was found and assessors found a concept.

remove tags that have one character and those that have non-ASCII encoded characters (e.g., `xab`, `xd4`). (2) We randomly select 10 terms from the seed set and retrieve the community that they belong to according to each algorithm. (3) For each community we show at most 10 tags for user evaluation. If the community contains more than 10 elements, we first add those tags that appear in the seed set and then randomly select from the remaining tags until we reach a total of 10. With this sampling approach, we ensure that we are evaluating similar topics for each algorithm.

In Table II, we show some of the characteristics of the sampled dataset that is evaluated by users. In total, we are left with 660 communities (17.3% of all communities), of which 633 correspond to unique sets of tags (27 identical communities were identified by more than one algorithm).

TABLE II
STATISTICS ABOUT COMMUNITIES SAMPLED FOR THE USER STUDY ACROSS ALL 20 QUERIES

Method	Count	Community Size			
		Max	Min	Avg	S.Dev
AIC-EDGE	89	10	3	6.06	2.93
AIC-VERTEX	81	10	3	5.64	3.00
EIGENVEC	66	10	4	9.67	1.27
INFOMAP	167	10	4	6.05	2.20
LBLPROP	39	10	4	9.33	1.69
MULTILVL	86	10	5	9.80	0.87
MSCAN	132	10	4	7.24	2.34

Evaluators: We recruited 40 students from two engineering schools in Santiago, Chile. Most of the tags in our evaluation were in English, therefore, participants were asked to have at least an intermediate level of English (i.e., to normally read and understand news and non-technical books in English). Evaluations were split into 3 sessions. We collected 3,165 evaluations, averaging 79.1 ($\pm$40.4) assessments per user, and 23.8 ($\pm$22.2) seconds per question. We found that 50% of users knew the meaning of at least 90% of the tags in the communities they evaluated, whereas all users knew the meaning of more than 67% of the tags.

Inter-assessor agreement: Each community in our evaluation dataset is reviewed by 5 users in order to allow computing of agreement measures. We use Krippendorff's alpha (α) [32] to measure agreement because it can be used in cases when some responses are left blank (i.e., cases in which users did not understand a tag). Using this metric, we find that the median value of α for most communities falls into the range $0.2 < \alpha \leq 0.4$ (fair agreement), except for those with the minimum 3 tags, where lower agreement was observed: we believe it was more difficult for users to identify a "topic" with such few terms.

Precision based on majority voting: In order to compute the precision of a community of tags T, we count each tag as: *relevant* (T_R or 1: if selected by the user as related to other tags in the cluster), *unknown* (T_U or 0: if the user marked the tag as unknown to them) or *irrelevant* (T_I or -1: if the user did not select the tag or if it was part of a community, which the user considered as only unrelated tags). We then consider T_R as true positives, T_I as false positives, and discard T_U; thus, we compute the precision as $P(T) = \frac{\#T_R}{\#T - \#T_U}$. We only mark a tag as T_R, T_U or T_I for the cases when there is a majority consensus.

Precision measures are obtained considering sets of tags for which there was moderate or high agreement ($\alpha > 0.4$), leaving 263 (39.8%) of the 660 sampled communities. Figure 6 analyzes the resulting precision[6]. The methods we propose based on islands (AIC-*) have higher mean and median precision (> 0.6) than the other community detection methods; though there was no significant difference between the median precisions of AIC-EDGE and AIC-VERTEX (p-value $\approx .287$), there was a significant difference from both AIC-* methods to all other methods included in the evaluation (p-value $< .05$).[7]

Relative recall: An algorithm that creates smaller communities tends to have higher precision, but also tends to split related terms into different communities. Hence, we must also measure the *recall* of each method. Since there is no

[6]The box-plots of this paper are Tukey box-plots where the solid line denotes median, the dashed line denotes mean, box-edges denote quartiles, whiskers denote the lowest/highest observation with 1.5 IQR of the box-edges, and other points denote outliers.

[7]p-values were computed using the Mann-Whitney U test.

groundtruth that tells us which tags are truly related for a given query, we create a data-driven groundtruth, and use it to measure *relative recall*. The idea behind this measure is that for each query, we compute the pairs of tags that users agreed by majority to be related across all algorithms. We call this number *total positives*. Then, for each algorithm we check how many of those pairs appear in the same community. We call this number *true positives*. Relative recall is then defined as the ratio of *true positives* over *total positives*. To identify pairs of related tags, given a user assessment for a set of tags $T = \{t_1, \ldots, t_n\}$, we use the function:

$$\text{rel}(t_i, t_j) = \begin{cases} 1 & t_i = 1 \wedge t_j = 1 \\ 0 & t_i = 0 \vee t_j = 0 \vee (t_i = -1 \wedge t_j = -1) \\ -1 & otherwise \end{cases}$$

where 1 indicates relatedness, 0 indicates neutral, and -1 indicates not related. We then take the sum of this function for all pairs across all user assessments for a specific query and algorithm. To compute the relative recall of a particular algorithm and query, we take the sum of all such pairs for all *other* algorithms and select those with a positive score (> 0) as related pairs by consensus[8]. We compute the relative recall for that algorithm and query as the ratio of related pairs appearing in the same community vs. all such pairs.

One concern using related terms selected by consensus is again that users may have different topics in mind for why terms are related. To help mitigate this issue, we compute relative recall on a per-query basis. We also compute a weighted version of relative recall where we take the sum of the $\text{rel}(\cdot, \cdot)$ function for all positive pairs appearing in the same community divided by the sum for all such pairs, thus giving more weight in the recall measure to pairs that were repeatedly considered related by different users for that query. In the end, both the weighted and non-weighted results were very similar, hence we show only the weighted results.

We present weighted relative recall for each algorithm across all queries as a box-plot in Figure 7. We see that algorithms producing much larger average community sizes have much higher relative recall: LBLPROP (avg. community size 104.58), EIGENVEC (81.91) and MULTILVL (68.17), have larger communities and thus higher relative recall than AIC-EDGE (7.33), AIC-VERTEX (6.47), MSCAN (7.31), and INFOMAP (6.54). On the other hand, amongst the four algorithms producing smaller communities, we see that AIC-* methods have significantly higher recall (and precision) than MSCAN or INFOMAP (p-value $< .01$).

VI. CONCLUSIONS AND FUTURE WORK

The work presented in this paper is motivated by the goal of producing a topical clustering of multimedia resources based on tags. We focus on community detection techniques since there is a variety of established methods proposed in the literature and they have the significant benefit of not requiring a fixed number of clusters to be provided beforehand. However, without this fixed criterion, different algorithms can produce very different results. We found two well-distinguished types of algorithm: three algorithms that produced large communities (avg. size > 68), and four algorithms that produced small communities (avg. size < 8) including the two we propose.

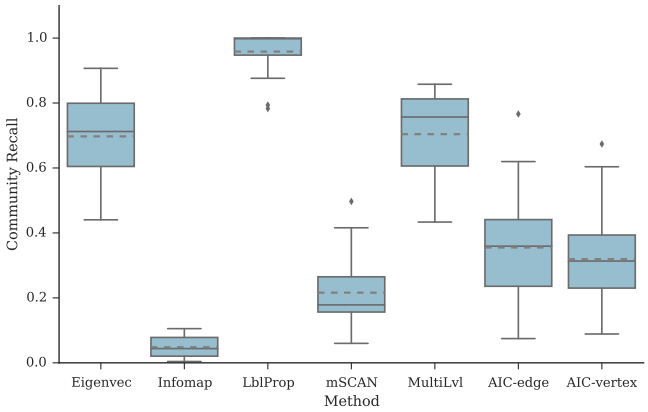

Fig. 7. Comparison of relative recall for the methods.

One major obstacle faced in this work was deciding on appropriate methods for evaluation. Our user evaluation was a costly process in terms of manual effort expended by human assessors, where this methodology puts a practical limit on the variety of algorithms, configurations, datasets, etc., that can be considered. On the other hand, it is not clear how one can create, *a priori*, a gold standard for tag clusters, particularly when, as we have seen from the results herein, users may often disagree on the relatedness of sets of terms. From our user study, we found that assessors had mixed agreement on which tags were related in the set presented. Looking at assessments with agreement ($\alpha > 0.4$), the two methods we propose had the highest mean and median precision. Considering relative recall, our methods were well-beaten by those producing large communities, but our methods outperformed the other two that produce equivalently-sized communities.

Query-dependent annotation clustering has some unique benefits: it is applied on smaller graphs and since it can be applied client-side, it can reduce server load and can even be used to aggregate results from multiple servers; also, by clustering only resources relevant to a specific topic, it is possible that the quality of clusters is improved with respect to that topic given that polysemous tags are more likely be used in the sense captured by the query.

There are still some open questions on which resolution of community detection is most desirable for clustering multimedia resources; our results are still inconclusive as to which community size is more useful when clustering search results themselves– whether smaller communities with better precision, or larger communities with better recall. Also, when considering online clustering, an important consideration is the number of results returned by the engine.

Our next major step is to use the results of our tag-clustering methods to investigate clustering on the level of resources, asking users to evaluate the topic relatedness of, e.g., image

[8]Unlike typical relative recall measures in IR, we do not include the pairs of the algorithm under testing since different algorithms may have different numbers of related pairs associated with them and each algorithm has a recall of 1 for its own pairs.

clusters or video clusters rather than the tags with which they are annotated. We also plan to investigate the effectiveness of community detection techniques for identifying topics in other datasets with tagged multimedia search results.

ACKNOWLEDGEMENTS

This work was partially funded by the Millennium Nucleus Center for Semantic Web Research under Grant No. NC120004, by Fondecyt Grant No. 11140900 and Grant No. 11121511. Teresa Bracamonte was also supported by Conicyt, Chile (CONICYT-PCHA/Doctorado Nacional/2013-63130260). We thank Ignacio Valderrama for helping with the evaluation, as well as the participants of the user study. We also thank Benjamín Bustos for his valuable feedback.

REFERENCES

[1] D. Cai, X. He, Z. Li, W.-Y. Ma, and J.-R. Wen, "Hierarchical clustering of www image search results using visual, textual and link information," in *Proceedings of the 12th Annual ACM International Conference on Multimedia*, ser. MULTIMEDIA '04. New York, NY, USA: ACM, 2004, pp. 952–959. [Online]. Available: http://doi.acm.org/10.1145/1027527.1027747

[2] F. Jing, C. Wang, Y. Yao, K. Deng, L. Zhang, and W. Ma, "Igroup: web image search results clustering," in *Proceedings of the 14th ACM International Conference on Multimedia, Santa Barbara, CA, USA, October 23-27, 2006*, 2006, pp. 377–384. [Online]. Available: http://doi.acm.org/10.1145/1180639.1180720

[3] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," *arXiv preprint arXiv:1310.1531*, 2013.

[4] K. Makantasis, A. Doulamis, and N. Doulamis, "A non-parametric unsupervised approach for content based image retrieval and clustering," in *Proceedings of the 4th ACM/IEEE International Workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Stream*, ser. ARTEMIS '13. New York, NY, USA: ACM, 2013, pp. 33–40. [Online]. Available: http://doi.acm.org/10.1145/2510650.2510656

[5] J. Yu, R. Hong, M. Wang, and J. You, "Image clustering based on sparse patch alignment framework," *Pattern Recognition*, vol. 47, no. 11, pp. 3512 – 3519, 2014.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003. [Online]. Available: http://www.jmlr.org/papers/v3/blei03a.html

[7] D. Putthividhya, H. T. Attias, and S. S. Nagarajan, "Topic regression multi-modal Latent Dirichlet Allocation for image annotation," in *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, 2010, pp. 3408–3415.

[8] J. Tian, T. Huang, Y. Huang, Z. Zhang, Z. Guo, and K. Fu, "A new method for image understanding and retrieval using text-mined knowledge," in *Advanced Data Mining and Applications*. Springer, 2014, pp. 684–694.

[9] S. Papadopoulos, Y. Kompatsiaris, A. Vakali, and P. Spyridonos, "Community detection in social media," *Data Mining and Knowledge Discovery*, vol. 24, no. 3, pp. 515–554, 2012.

[10] X. He, M.-Y. Kan, P. Xie, and X. Chen, "Comment-based multi-view clustering of web 2.0 items," in *Proceedings of the 23rd International Conference on World Wide Web*, ser. WWW '14. New York, NY, USA: ACM, 2014, pp. 771–782. [Online]. Available: http://doi.acm.org/10.1145/2566486.2567975

[11] P. Q. Nguyen, A. T. Nguyen-Thi, T. D. Ngo, and T. A. H. Nguyen, "Using textual semantic similarity to improve clustering quality of web video search results," in *Knowledge and Systems Engineering (KSE), 2015 Seventh International Conference on*, Oct 2015, pp. 156–161.

[12] A. Hindle, J. Shao, D. Lin, J. Lu, and R. Zhang, "Clustering web video search results based on integration of multiple features," *World Wide Web*, vol. 14, no. 1, pp. 53–73, 2011.

[13] J. G. Moreno and G. Dias, *Using Text-Based Web Image Search Results Clustering to Minimize Mobile Devices Wasted Space-Interface*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 532–544. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-36973-5_45

[14] M. Strohmaier, D. Helic, D. Benz, C. Körner, and R. Kern, "Evaluation of Folksonomy Induction Algorithms," *ACM TIST*, vol. 3, no. 4, p. 74, 2012.

[15] D. Laniado, D. Eynard, and M. Colombetti, "Using wordnet to turn a folksonomy into a hierarchy of concepts," in *Proceedings of the 4th Italian Semantic Web Workshop, Dipartimento di Informatica - Universita' degli Studi di Bari - Italy, 18-20 December, 2007*, 2007.

[16] A. Plangprasopchok, K. Lerman, and L. Getoor, "Growing a tree in the forest: constructing folksonomies by integrating structured metadata," in *ACM SIGKDD*, 2010, pp. 949–958.

[17] A. Passant, "Using ontologies to strengthen folksonomies and enrich information retrieval in weblogs," in *ICWSM 2007*, 2007. [Online]. Available: http://www.icwsm.org/papers/paper15.html

[18] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3-5, pp. 75–174, 2010.

[19] V. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, p. P10008, 2008.

[20] M. E. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical review E*, vol. 74, no. 3, p. 036104, 2006.

[21] M. Girvan and M. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, p. 7821, 2002.

[22] X. Xu, N. Yuruk, Z. Feng, and T. A. J. Schweiger, "Scan: a structural clustering algorithm for networks," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '07. New York, NY, USA: ACM, 2007, pp. 824–833. [Online]. Available: http://doi.acm.org/10.1145/1281192.1281280

[23] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E*, vol. 76, no. 3, p. 036106, 2007.

[24] M. Rosvall and C. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the National Academy of Sciences*, vol. 105, no. 4, p. 1118, 2008.

[25] V. Batagelj, N. Kejzar, S. Korenjak-Cerne, and M. Zaversnik, "Analyzing the structure of U.S. patents network," in *Data Science and Classification*, 2006, pp. 141–148.

[26] M. Zaveršnik and V. Batagelj, "Islands," *International Sunbelt Social Network Conference*, 2004.

[27] X. Li, C. Snoek, and M. Worring, "Learning social tag relevance by neighbor voting," *Multimedia, IEEE Transactions on*, vol. 11, no. 7, pp. 1310–1322, 2009.

[28] D. Liu, X.-S. Hua, L. Yang, M. Wang, and H.-J. Zhang, "Tag ranking," in *Proceedings of the 18th international conference on World wide web*, ser. WWW '09. New York, NY, USA: ACM, 2009, pp. 351–360. [Online]. Available: http://doi.acm.org/10.1145/1526709.1526757

[29] X. Li, T. Uricchio, L. Ballan, M. Bertini, C. G. M. Snoek, and A. D. Bimbo, "Socializing the semantic gap: A comparative survey on image tag assignment, refinement, and retrieval," *ACM Comput. Surv.*, vol. 49, no. 1, pp. 14:1–14:39, Jun. 2016. [Online]. Available: http://doi.acm.org/10.1145/2906152

[30] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks*, vol. 30, no. 1-7, pp. 107–117, 1998. [Online]. Available: http://dx.doi.org/10.1016/S0169-7552(98)00110-X

[31] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data mining*, vol. 1996. AAAI Press, 1996, pp. 226–231.

[32] K. Krippendorff, "Reliability in content analysis," *Human Communication Research*, vol. 30, no. 3, pp. 411–433, 2004.