# A Clustering Approach for Tag Recommendation in Social Environments

Ludovico Boratto[*1], Salvatore Carta[*1], Matteo Manca[*1], Fabrizio Mulas[*1], Paolo Pilloni[*1], G. Michele Pinna[*1], Eloisa Vargiu[2]

[*1]Dipartimento di Matematica e Informatica, Università di Cagliari
Via Ospedale, 72 – 09124 Cagliari, Italy
[2]Barcelona Digital (bDigital)
C/ Roc Boronat, 117 - 08018 Barcelona, Spain
[*1]{ludovico.boratto, salvatore, matteo.manca, fabrizio.mulas, paolo.pilloni, gmpinna}@unica.it; [2]evargiu@bdigital.org

***Abstract*- Collaborative tagging is the process by which users classify shared content using keywords. Although its popularity keeps growing on the Web, content retrieval can be difficult since people tag differently. Moreover, there are some well-known linguistic phenomena. Recently, several attempts to avoid such limitations by recommending tags to users or by creating tag clusters have been presented. In this paper we propose an approach to cluster tags by monitoring the activity of the users in a tagging system. The created clusters can be used to recommend tags when a user uploads or searches a resource, in order to facilitate content retrieval. Experiments are performed by comparing with a classic tag clustering approach and results show the capability of the approach to cluster strongly related tags.**

***Keywords*- *Tag Clustering; Social Recommendation; Tagging System***

## I. INTRODUCTION

With the development of Web 2.0, the use of the Web has become increasingly widespread. This led to a continuous growth of information sources, with daily uploaded resources shared by many users. Traditional techniques performed by experts to categorize and index data have been quickly joined with collaborative tagging approaches, by which a community of users adds tags to shared content. The success of collaborative tagging was driven by the fact that tagging does not require specific skills and seems a natural way for people to classify any kind of content.

A *tagging system* is a web application that allows users to add keywords (called *tags*) to classify resources [1]. Recently, several successful tagging systems have been developed (e.g., Del.icio.us, Flickr, Last.fm, CiteULike). A set of tags (tagspace) can be explored in several ways and many tagging systems usually define sets of related tags, called *tag clouds*, that help the tagspace visualization. However, as highlighted in [2], there are some well-known linguistic limitations that can inhibit information retrieval in those systems. In fact, the meaning or semantics of a tag is usually unknown. For example, tag "orange" may refer to a fruit or a color, and this can lead to incorrect relations between tags and resources. Moreover, people use different tags to select the same resources. For example, a resource related to a pasta dish could be tagged as "Italian food", "spaghetti", "first course" as well as many other terms. On the one hand, this allows users to choose freely which tags classify resources in a useful way; on the other hand, it can limit the searching activity of other users within the tagspace. Since users tag differently, a user might search a resource using a query that contains a set of tags different from the ones used to classify the resource, without finding it. So, in order to find a resource, it might be needed to search several times using different tags and people should evaluate the relevance of the retrieved resources.

In order to avoid these problems, social recommender systems have been developed [1, 3]. Some of these systems have been developed to recommend tags to users, considering the user profile or the content of the resource that is being uploaded [4, 5, 6, 7]. These systems are usually limited by the well known *cold start problem*, i.e., if the user is new or the resource is not similar to any of the existing resources, no tag can be recommended to the user.

As highlighted in [8], another way to avoid the previously mentioned limitations and simplify the exploration of a tagging system, would be grouping related tags together. In fact, the definition of sets of related tags would help the identification of a context, which would avoid polysemy and synonymy thus making resources retrieval easier.

Recently, several approaches have been proposed to cluster tags. When a user puts a resource in a tagging system, they usually create associations between the resource and the tags used to classify that resource. Then, if two tags are used for the same resource (*tags co-occurrence*), an association between those two tags is created. Tags associations are used to cluster together all the related tags in the tagging system. The other works in the literature, however, do not exploit the potential source of information coming from monitoring users' search activity performed inside the tagging system. Therefore, associations between tags and resources are created only when a resource is put into the system. This means that a resource can be associated to misleading tags, and the misleading associations would affect the performances of the system. Moreover, new meaningful associations between tags and resources would not automatically discovered by the system.

In this paper we propose *RATC* (Robust Automated Tag Clustering), an approach able to cluster related tags, in order to facilitate the upload and the retrieval of resources. Users' activity in the search engine of a tagging system is monitored, in order to exploit implicit feedbacks provided by users. A feedback is collected each time a user finds a relevant resource during a search in a tagging system. *RATC* uses the feedback to dynamically strengthen associations between the resource indicated by the user and the tags used in the search string. Tag-resource associations are then used to infer tag-tag associations by adopting a standard correlation measure. Tag-tag associations allow to cluster tags in order to find strongly related tag sets.

Each time a user uploads a resource into a social web application, these sets of tags might be recommended to her/him in order to help the classification of the resource. Moreover, the set of tags could be used to facilitate the retrieval of the resources to the users who might be interested in it, by recommending tags while the user is typing the query.

Results have been compared with those obtained by adopting the classic approach proposed in [9], showing an improvement in the presence of strongly related tags in a cluster.

*RATC* brings a relevant contribution both to the tag clustering research area and with respect to the existing social recommender systems that recommend tags. In fact, by supervising users activity in a tagging system and monitoring their searches, we can progressively create and update tag-resource associations and tag-tag associations, rewarding the real semantic relations among tags and penalizing the misleading ones. None of the existing works in tag clustering is able to dynamically strengthen associations between tags while the system is used. Moreover, none of the existing social recommender systems that operate with tags uses clustering to produce the recommendations. In a web scenario, in which things evolve quickly, a form of classification that does not require supervision like clustering is an extremely simple and strongly effective way to produce associations between similar tags. Moreover, recommendations can be produced without using neither the user profile nor the content of the resource, so *RATC* is not affected by the cold start problem.

This paper is an extension of the approach described in [10]. With respect to the previously published work, we present the contribution to the social recommender systems research area and illustrate the results of a new set of experiments that analyzes the structure of each cluster.

The rest of the paper is organized as follows: Section II contains a detailed description of the steps we followed to build *RATC*; Section III describes the performed experiments and outlines main results; Section IV presents the state-of-the-art in tag clustering and tag recommendation; Section V discusses conclusions and future work.

## II. RATC - ROBUST AUTOMATED TAG CLUSTERING

*RATC*, which stands for *Robust Automated Tag Clustering*, monitors the activity of users in the search engine of a tagging system. The approach has been defined "robust" to put into evidence its ability to overwhelm the misleading resource classification problem.

### A. Top Level View of the Approach

*RATC* encompasses four main steps:

**Tag-resource associations creation.** As in any tagging system, each time a new resource is put into the system, the system creates a *tag-resource* association among that resource and the tags used to classify it.

**Dynamic tag-resource associations evaluation.** Users activity in the tagging system search engine is monitored and exploited in order to update existing *tag-resource* associations and to create new ones.

**Tag-tag associations creation and quantification.** Dynamic *tag-resource* associations are used to create associations among tags (*tag-tag* associations). A standard similarity measure, i.e., cosine similarity, is used to evaluate the similarity among tags. The result of this process is a weighted graph (*tag similarity graph*) in which each node represents a tag and each weighted arc represents the similarity value of the tags it connects.

**Clustering.** The community detection algorithm proposed in [11] is applied to the tag similarity graph in order to identify the intrinsic communities of tags.

### B. Representation of the Tagging System

A tagging system is a community driven tool that allows users to classify resources by means of tags. We represent a tagging system as a bipartite graph that contains:

- a set $T$ of tags $t$;
- a set $R$ of resources $r$;
- a set $A: (T \times R)$ of weighted arcs $t$-$r$, that represent the tag-resource associations. The weight of the tag-resource associations represents the number of times that a tag has been associated to a resource by users.

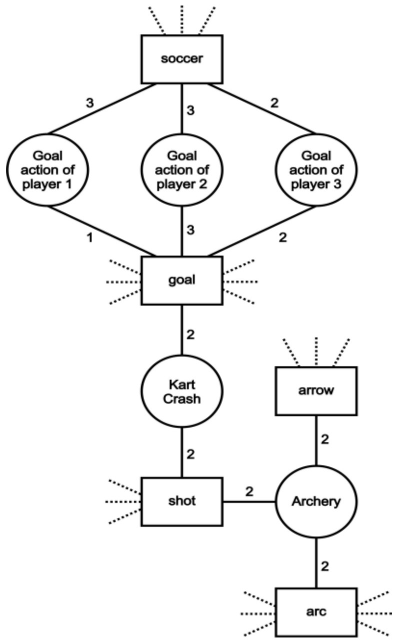



Fig. 1 An example of tagging system

As depicted in Fig. 1, the tagging system is composed of a set of tags (rectangular nodes) linked by a weighted arc to a subset of resources (round nodes). In the example in figure there are three resources concerning with "goal actions" in a soccer game (resources represent multimedia documents, like videos or pictures). All of those resources have been classified with the tags *soccer* and *goal* and the weight of each arc represents the strength of the association between a tag and a resource. Each tag has some outgoing dotted arcs, which indicate that there are other resources linked to those tags (not shown in the example).

As a final remark, let us note that *RATC* does not take into account different meaning associated to a same word (i.e., polysemy). For instance, in the example in Fig. 1, tag goal is used either as a successful attempt at scoring in a soccer game or as the place designed at the end of a race.

| | **Goal action of player 1** | **Goal action of player 2** | **Goal action of player 3** | **Kart Crash** | **Archery** |
|---|---|---|---|---|---|
| **soccer** | 3 | 3 | 2 | 0 | 0 |
| **goal** | 1 | 3 | 2 | 2 | 0 |
| **shot** | 0 | 0 | 0 | 2 | 2 |
| **arrow** | 0 | 0 | 0 | 0 | 2 |
| **arc** | 0 | 0 | 0 | 0 | 2 |

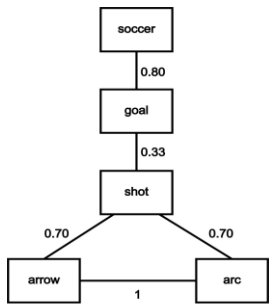



Fig. 2 Tag similarity graph

### *C. Quantification of the Tag-Resource Associations*

The standard search paradigm provided by tagging services is based on query strings containing one (or more) tags. The search returns a list of resources associated to these tags. To provide such list, a ranking of the results is derived according to the tag-resource associations available in the tagging system. The associations can be considered as the strength of the correlation between a resource and each tag used to classify it. While tagging systems usually associate tags and resources at upload time, implicit user feedback, coming from its search activity, can be exploited to improve *tag-resource* associations.

To represent the strength of *tag-resource* associations we adopted an algorithm based on counters. The algorithm exploits users feedback to discover and emphasize correct associations strength, while making negligible the contribution of "noisy" associations. The strength of each association evolves according to an extremely simple and effective mechanism. A *tag-resource* association is created each time a resource is added to the system by a user. After a tag-based search operation, each time the user selects a resource, the counter of the tag-resource association is increased (an example of *tag-resource* associations is shown in Fig. 1). Although a huge number of resources may be related to a single tag, their relevance will depend on the feedbacks provided by the community of users. In such a way the association of a misleading tag to a resource will give a negligible contribution.

In order to contain the counters relative to tag-resource associations, a matrix $W=\{w_{rt}\}$ is defined, where $w_{rt}$ is the association between a resource *r* and tag *t* (an example is depicted in Fig. 2).

Initial values are assigned when a new resource is uploaded and values are updated either when a user adds a tag already present in the database or when a feedback is given. This means that the initial values are just a starting point and that the tag-resource associations are updated when feedbacks are collected. When a new resource is uploaded to the tagging system together with some tags, the corresponding *tag-resource* counter is set to *1*. If such association is already in the system, the corresponding $w_{rt}$ is incremented. The matrix is also updated when a user performs a search in the tagging system and selects one of the results as relevant. At this stage, after the user selection took place, the counters between the selected resource and all the tags in the query list are incremented, namely $w_{rt} = w_{rt} + 1$.

The tagging system shown in Fig. 1 has been built using the tag-resource counters described above. Let us stress the fact that the strength of the relation between a tag and a resource in our tagging system is based on the implicit feedback left by the users during the use of the system. For example, tag *soccer* has been used three times to classify and search the second resource concerning with the goal action of a player.

### D. Quantification of the Tag-Tag Associations

Let $v_i$ be the vector of associations among a tag *i* and its related resources and $v_j$ be the vector of associations among a tag *j* and its related resources. The similarity $s_{ij}$ between tag *i* and tag *j* can be measured by the cosine similarity between the vectors as follows:

$$s_{ij} = \cos(v_i, v_j) = \frac{v_i \cdot v_j}{\|v_i\|_2 \times \|v_j\|_2}$$

These similarities can be represented in a graph, called *tag similarity graph*, which links each couple of associated tags with a weighted arc. An example, built using the associations among tags and the resources shown in Fig. 1, is represented in Fig. 2 (note that the values of the associations in the figure have been calculated considering the whole tagging system).

### E. Clustering

To perform clustering we adopted MCL (Markov Clustering Algorithm) [11]. MCL is a Community Detection algorithm, built to find cluster structure in simple graphs, considering the similarity between vertexes. The algorithm is based on the intuition that if nodes belong to the same cluster, the longest path between them is relatively short. On the contrary, for nodes that belong to different clusters its value is relatively high. That means that it should be difficult to move from one cluster to another with a random walk.

To explain how a random walk on a weighted graph works, suppose that a random walker is, at a certain instant, in a node *i*. Node *j*, where she/he will be at the following instant, is chosen among the first neighbors of *i*, with a probability proportional to the weight of the edge between *i* and *j*. In such a way it is possible to create a transition matrix *M* of size $N \times N$.

The algorithm works using a bootstrapping procedure, i.e., the probability of random walks in the graph is calculated iteratively using two sets of operators (named *expansion* and *inflation*) and applied to the matrices. The expansion operator computes the square of the matrix (the product of that matrix with itself). Inflation operator is the entry-wise Hadamard-Schur product of the matrix combined with a diagonal scaling, to allow the resulting matrix to be a stochastic matrix. The inflation operator and the expansion operator are subsequently applied by the algorithm that converges quadratically in the neighborhood of doubly idempotent stochastic matrices, i.e., matrices that do not change under the action of the two operators. The obtained matrix returns a disconnected graph, in which each component contains nodes that belong to the same cluster.

The inflation operator depends from a parameter *r*, known as *granularity*. By incrementing this operator, the strength of the inflation operator is higher and this causes a higher number of clusters.

## III. EXPERIMENTS AND RESULTS

To evaluate *RATC*, we first adopted a tagging system with an internal experimental search engine [12], and then we compared the performances with a classic approach in tag clustering [9].

Several aspects have been taken into account while performing comparisons regarding the robustness of the two approaches. In particular, to analyze the impact of noise in the performances, we suitably added noisy tags to the tagging system.

The quality of the obtained clusters has been evaluated in two ways.

**Quality of the Clustering.** In the first set of experiments, in order to evaluate the capability of the tagging systems to create a good clustering, i.e., a partitioning into sets of correlated tags, a domain engineer evaluated the clusters in terms of precision and recall. In other words, it was evaluated how close is the clustering created by an approach to the clustering that should have been created.

**Analysis of the Clusters.** In the second set of experiments, the domain engineer evaluated the quality of each cluster that was produced, in order to measure the percentage of tags in a cluster that were actually correlated.

The rest of the section is organized as follows: Subsection A describes how the dataset was collected; Subsection B briefly presents the approach used to compare the performances of *RATC*; Subsection C presents the strategy adopted to conduct the experiments and metrics used to evaluate the performances; Subsection D presents the results obtained in each set of experiments.

### A. Dataset Collection and Pre-processing

To conduct the experiments 10 volunteers populated a tagging system [12] (*resource acquisition* step). They were asked to select as many videos as they wanted from YouTube[1] and add them to the tagging system. The application domain was limited to "sport" as specific topic, which can be considered a concept domain. Each video was classified with four tags related to the resource and two tags (the noisy tags) not related to the resource. Noisy tags were added to simulate the noise that typically occurs in practice.

Once the tagging system was populated, volunteers were asked to perform normal searches in the tagging system (*feedback collection* step). During this step, *RATC* improves its performances by monitoring users search activity. Videos are chosen based on a preview shown to the user and their original description.

This step started as soon as the resource acquisition step was completed. The reason was to neatly separate the initial values of the correlations from their evolutions caused by the feedbacks of the users.

#### 1) Acquisition of the Resources:

Each time a volunteer added a new video to the application, she/he had to create two sets of tags. The former is devoted to contain (at least) four characteristic tags strongly related to the video; the latter contains two tags not related to the video but in the same domain (in this experiments, "sport"). This tag set is required to create some verifiable noise and it has been used to monitor the progressive decreasing of their correlation with the video they had been initially introduced with. Such noise is useful to evaluate the clustering approach, both to monitor how the structure of the cluster changes and to evaluate the quality of the clusters.

The tagging system was populated with a total of 406 videos, 1021 tags, 2597 video-tag correlations.

Users were left free to use any tag they wanted while populating the tagging system. However, the collected data was pre-processed, in order to remove tags that express emotions or feelings and cluster only tags useful for recommendations purposes (it would not be useful to recommend tags like "good" or "beautiful", neither when a resource is uploaded, nor when a user is performing a search). After the pre-processing, the system involves 964 tags.

#### 2) Collection of the Feedbacks:

During this step, each volunteer performed 300 searches in the tagging systems. For each search, each volunteer: (i) entered a list of tags as query for the search; and (ii) selected, from the videos in the results list, the video most related with the query.

A feedback is then collected each time a user performed a search and consequently tag-resource counters are incremented. After entering a list of tags, she/he was free to analyze the videos resulting from the search (during this phase the user could also play all the videos to help her/his choice). At the end of this activity, the user had to pick a video from the output list providing a feedback. This emulates a real world scenario in which a user, after the result of a search is displayed, selects the resources she/he is interested in.

### B. Description of the Benchmark Approach

In order to choose how to compare *RATC*, we considered the most similar works both in the tag clustering area (i.e., the one presented in [9]) and in the tag recommendation area (i.e., the one presented in [13]).

Since in [13] tag recommendations are produced based on the co-occurrences of tags and [9] also presents an approach based on co-occurrences that also clusters tags, we decided to compare to the one proposed in [9]. Moreover, the fact that "promotion function" used by [13] is based on the tagging behavior of the users in Flickr, it would not be significant to applicate the function to a completely different domain.

The approach selected for comparison with *RATC*, i.e., the one proposed in [9]), will be hereinafter named *ATC* (which stands for *Automated Tag Clustering*).

*ATC* is aimed at clustering tags to improve user experience in the use of a tagging system and minimize the classical linguistic limitations. An algorithm to find strongly related tags by counting the number of tag co-occurrences used for a page is defined. A cut-off point is determined to evaluate when a counter is useful. Then, a sparse matrix is produced and its

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

elements are the similarities among tags.

A graph representation of the similarities is defined and the tags are grouped with a graph clustering algorithm based on the spectral bisection. The quality of the partitioning is measured with the “modularity function” $Q$ [14].

*ATC* performs the following steps: (i) spectral bisection to split the graph into two clusters; (ii) comparison with the value of the modularity function $Q_0$ of the original unpartitioned graph to the value of the modularity function $Q_1$ of the partitioned graph, if $Q_1 > Q_0$ accepts the partitioning, otherwise rejects the partitioning; and (iii) a recursive step on each accepted partition.

A similarity counter is increased for each pair of tags that belong to the same cluster and the top similar pairs of tags are extracted.

*C. Strategy and Evaluation Measures*

To assess the ability of *RATC* to learn from monitoring the activity of the users, the state of the tagging system (i.e., the current values of each tag-resource association) has been saved and used to evaluate clusters quality each 50 feedbacks. In this way, six tagging system sessions, which can be used to compare the two tag clusterings, are available.

As already pointed out, a subset of known tags was added to the tagging system to create some verifiable noise. To evaluate the quality of the clusters created by each approach in presence of noise we conducted experiments considering both the original dataset and a dataset in which we removed the noisy tags.

The choice to make the evaluations at the end of the six sessions and to not recommend tags from the second session while a user uploaded her/his resource or while she/he was performing the searches was made in order to make evaluations not affected by the approach.

In other words, we wanted to collect a dataset that could allow us to evaluate how *RATC* would perform in presence of noise, without having the results biased by the fact that the tags collected were suggested by the system itself.

The only parameter that had to be set is the inflation value in the clustering step (set to 3.0).

Two sets of experiments, described below along with the used evaluation measures, were conducted.

*1) Evaluation of the Clusterings:*

The first set of experiments aims to evaluate the capability of *RATC* (*ATC*) to build a significant partitioning of the tags. To make fair comparisons, first, a domain engineer clustered the involved tags. Each cluster was created considering tags that refer to the same *concept*, i.e., a particular event or a clear “topic” that groups tags. Subsequently, the tag clustering obtained by the domain engineer is compared with the clusters automatically generated by using *RATC* and the ones obtained by applying *ATC*.

In order to perform a comparative evaluation of the clusterings, each single cluster of tags created by *RATC* (*ATC*) was compared to the cluster created by the domain engineer that contained the highest number of corresponding tags and the following sets were produced:

- *true positive tags* (TP): tags that appear both in a cluster generated by *RATC* (*ATC*) and in the cluster of the domain engineer partition.
- *true negative tags* (TN): tags that do not appear both in a cluster generated by *RATC* (*ATC*) and in the cluster of the domain engineer partition.
- *false positive tags* (FP): tags that appear in a cluster generated by *RATC* (*ATC*) and do not appear in the cluster of the domain engineer partition.
- *false negative tags* (FN): tags that do not appear in a cluster generated by *RATC* (*ATC*), but appear in the cluster of the domain engineer partition.

To validate the approach, we resort to classical information retrieval measures, such as micro- and macro-averaging of precision and recall [15].

Let us recall here that micro- and macro-averaging are aimed at obtaining estimates of $\rho$ and $r$ relative to the whole category set. In particular, micro-averaging evaluates the overall $\rho$ and $r$ by globally summing over all individual decisions.

In symbols:

$$\pi^{\mu} = \frac{TP}{TP+FP} = \frac{\sum_{i=1}^{m} TP_i}{\sum_{i=1}^{m}(TP_i + FP_i)}; \rho^{\mu} = \frac{TP}{TP+FN} = \frac{\sum_{i=1}^{m} TP_i}{\sum_{i=1}^{m}(TP_i + FN_i)};$$

where the “M” superscript stands for macroaveraging.

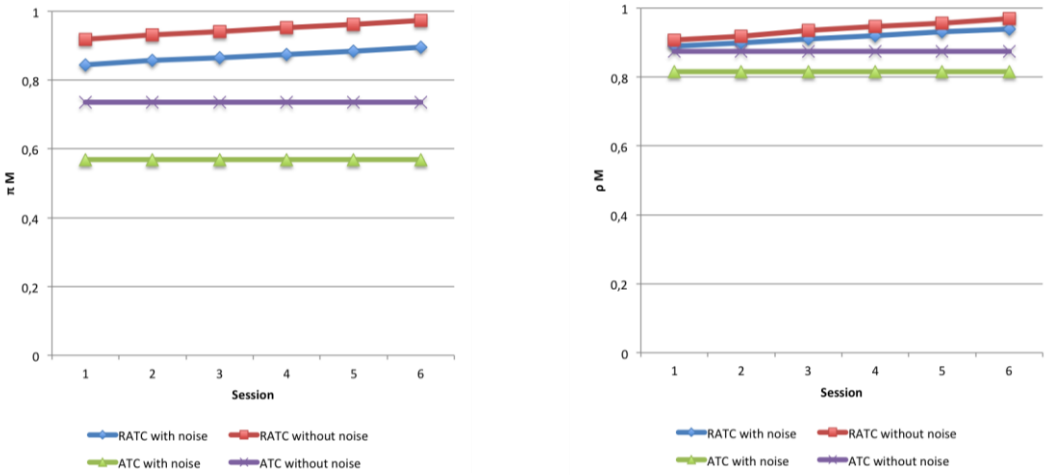
*1) Analysis of the Clusters:*

The second set of experiments was conducted in order to evaluate how correlated are the tags in each cluster. Note that the evaluation did not take into account the clusterings previously created by the engineer, but only the clusters produced by each approach.

The engineer evaluated each cluster created by *RATC* (*ATC*), in order to identify the concept that connects the tags. Then the meaningful tags in the cluster for that concept have been counted. The percentage of meaningful tags in a clustering with respect to the total number of tags in the clustering is then calculated.

## D. Results

This section presents the results for the previously presented set of experiments.




(a) (b)

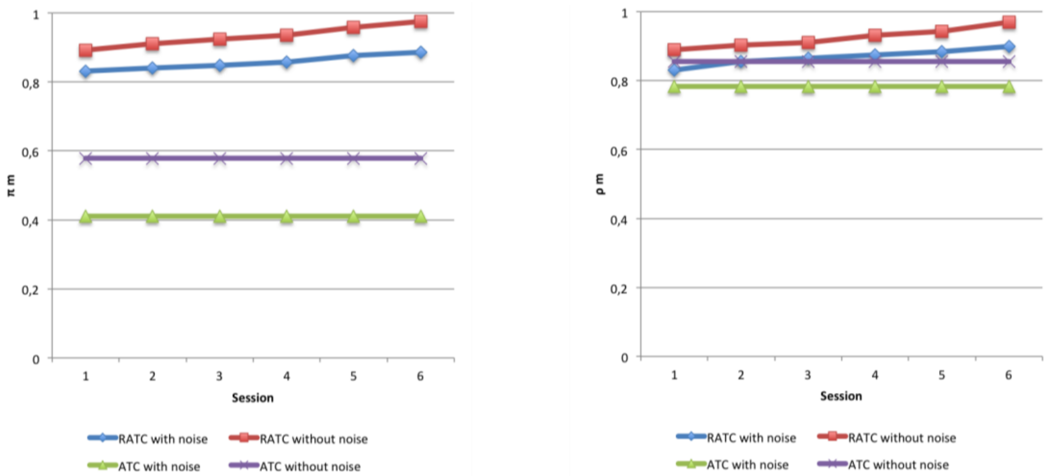
Fig. 3 Macro-averaging precision (a) and recall (b)

*1) Evaluation of the Clusterings:*

There is a great difference between the clusterings created in each session by *RATC* and *ATC* and the clusterings created by the domain engineer. In fact, the number of clusters created by the engineer was significantly lower than the number of clusters created by the two approaches studied in this paper. We can give an idea of such difference by saying that in the last session *RATC* without noise involved 266 clusters, while the domain engineer created 148 clusters.

This is because the clusters created by *RATC* and *ATC* also contain clusters composed by a very small number of tags (for example, the approaches detect a lot of clusters that contain only two tags), while the manual classification was limited only to bigger clusters strongly correlated with a concept.

Fig. 3 compares the results in terms of macro-averaging precision (Fig. 3-a) and recall (Fig. 3-b), both obtained by adopting *RATC* and *ATC* with and without noisy tags.




(a) (b)

Fig. 4 Micro-averaging precision (a) and recall (b)

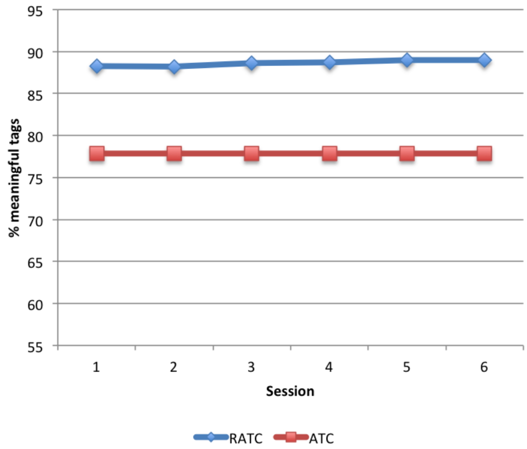



Fig. 5 Meaningful tags in the dataset without noisy tags

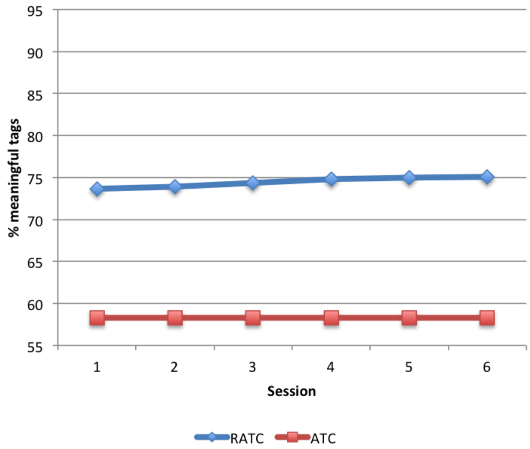



Fig. 6 Meaningful tags in the dataset with noisy tag

Fig. 4 compares the results in terms of micro-averaging precision (Fig. 4-a) and recall (Fig. 4-b), both obtained by adopting *RATC* and *ATC* with and without noisy tags.

Results show that *RATC* performs always better than *ATC*, and that such performances improve session by session, due to the fact that tag-resource associations and tag-tag associations get better with the use of the system (i.e., by applying the feedback mechanism).

In order to give an example of the capability of *RATC* to create more significant associations and to create a better clustering with respect to *ATC*, we report an example of the clusters created by the two approaches and by the domain engineer, that involve the same tags.

*RATC*: record, 100m, 1988, ben, championship, doping, johnson, powell

*ATC*: meters, record, 100m, 1988, ben, championship, track-and-field, doping, mennea, powell, cadets, livorno, ancona, debut

*Domain engineer*: 100m, 1988, championship, doping, ben, johnson, powell, track-and-field, record

The cluster created by *RATC* and by the domain engineer is clearly related to an event at the 1988 olympic games in which Mike Powell won the gold medal in the 100 meter race, after Ben Johnson was disqualified because of doping. In the cluster created by *ATC* a set of unrelated tags is present. Even if Pietro Mennea was a track-and-field athlete (like Powell and Johnson), tag mennea is not related to the other tags. Moreover, tags like cadets, livorno and ancona are not related to the concept of the cluster. This is the sign that *ATC* is not able to create strong enough associations that allow the clustering algorithm to operate a good partitioning of the tags.

*2) Analysis of the Clusterings:*

Fig. 5 shows the percentage of meaningful tags for each experiment conducted on the dataset without the noisy tags. We can see that our approach achieves a higher percentage of meaningful tags and its performances improve session by session, since tag-resource associations and tag-tag associations strengthen with the use of the system. In the first session the tag-resource associations have the same values for both the approaches, as no search activity was done in the system. Since *RATC* achieves far better results even in this session, this is the evidence that cosine similarity represents a better way to measure associations between tags.

In Fig. 6 we can see the percentage of meaningful tags in the experiments that used the dataset with noisy data. Even in this case our approach achieves better results and its performances improve session by session. We can also see that our approach is less affected by the introduction of the noisy tags, with just a 15% worsening in the first session.

## IV. RELATED WORK

In the literature, several approaches related to the one proposed in this paper, that either aim at clustering tags or at recommending tags to users, have been presented. This section illustrates them, divided by research area (i.e., tag clustering and tag recommendation).

### A. Tag Clustering

The work presented in [16] tries to derive the semantics behind a tagspace, so that the collaborative tagging can help finding groups of concepts and partial ontologies. This is achieved by using a combination of shallow pre-processing strategies and statistical techniques together with knowledge provided by ontologies available on the semantic web. This technique starts pre-processing the data and cleaning up the tagspace, then finds associations between tags evaluating their co-occurrences and clusters them. Semantic relations are extracted from the clusters and groups of highly related tags that conceptualize specific facets of knowledge and correspond to elements in ontologies are created. This approach differs from the one proposed in this paper since we do not pre-process the tagspace. *RATC*, in fact, is able to adaptively remove noisy tags by monitoring user interactions.

In [17], Hamasaki et al. propose a way to integrate a social network with collaborative tagging for ontology extraction. The usual tripartite models of ontologies based on users, tags and instances, are integrated with user-user relations. Concepts in each community (named *p-concepts*) are modeled, in order to resolve the polysemy/homonymy problem. The technique aims at grouping p-concepts and at finding keywords associations, with an algorithm that considers the interactions between users and p-concepts. *RATC* differs in the sense that we consider users interaction just to link resources to tags, without creating explicit associations among users and resources.

In [18] a technique to generate groups of semantically related tags through a probabilistic model is proposed. The technique is based on evaluating co-occurrence of tags, resources, and users. Each entity (a user, a resource, or a tag) is represented by a multi-dimensional vector, called *conceptual space*. *RATC* does not rely on a probabilistic model and it does not consider users as entities in the tag building task.

In [19], a co-clustering approach (based on the one proposed in [20]), in which elements of different datasets (tags and resources) are clustered together, is employed. The clustering activity is based on a similarity metric that uses tag co-occurrences and semantic knowledge about the tags. The relations among the elements are used to enrich ontologies and train multimedia processing algorithms (in case of multimedia social data). On the contrary, the clustering activity of *RATC* is based just on tags and new knowledge is inferred by clustering elements of the same dataset.

In [21], clusters of queries are created. Related queries are clustered together, in order to recommend a better query to users. This is achieved by finding the most descriptive words in a cluster and recommending better queries to users. We use queries in a different way, since associations between tags are not inferred by clustering queries themselves, but considering the resources that they classify.

Begelman et al. [9] cluster strongly related tags, in order to avoid the common limitations of tagging services. The proposed algorithm is based on counting the number of co-occurrences (i.e., tags that are used for the same resource) of any pair of tags and a cut-off point is determined to decide when the co-occurrence count is significant enough to be used. Tags are clustered with an algorithm based on the spectral bisection and the modularity function is used to measure the quality of a partitioning. Related tags are then automatically discovered by incrementing a counter or each pair of tags that belong to the same cluster. Although *RATC* works in a similar way, the main difference is that tag-resource associations are continuously updated during the use of the system.

### *B. Tag Recommendation*

In [4], Symeonidis et al. present an algorithm to produce tag recommendations using a Higher Order Singular Value Decomposition technique to model the entities of a tagging system (i.e., users, items and tags) and recommend tags given by similar users to similar resources. Similarly, in [5] an algorithm that builds tag recommendations by first modeling the interactions between users, items and tags, is presented. Once interactions are modeled, a Bayesian Personalized Ranking optimization criterion is adopted to build the tag recommendations. *RATC* significantly differs from these ones, since it does not use a probabilistic model to exploit the interaction of the user with the system. Moreover, we not consider the interactions of the users with the items (i.e., it does not keep track of which items the user interacts with).

Carmel et al. [6], produce tag recommendations by analyzing how similar each tag is to the tags previously used by the user and how similar each tag is to the document that has to be tagged. Each tag is weighted (the weight is a combination of the user similarity and the document similarity), in order to choose the tags to recommend. The difference with the approach that we are presenting is that *RATC* is able to recommend tags similar to the ones considered by the user, without considering the tags previously used by the user or the similarity with the document. As mentioned in the introduction, this flexibility allows to produce recommendations even for new users or new documents, avoiding the cold start problem.

In [22], Hotho et al. present an adaptation of the PageRank algorithm, named FolkRank, that considers the tagging system as a graph of undirected vertices (users, resources and tags), connected by co-occurrences of tags and users, users and resources, tags and resources (the algorithm was tested on del.iciou.us, so the associations consider the tags a users chooses to classify a bookmark). The idea behind FolkRank is that a resource tagged with important tags by important users is important. Like PakeRank, the algorithm uses a random walk technique in order to give a weight to each node. Given a tag *t*, tags are recommended by returning the top ranked tags associated to it. Jäschke et al. [23] similarly produce tag recommendations by randomly choosing a post of a user and using the FolkRank algorithm to predict the tags that the user will choose to classify the considered post. *RATC* differs from these approaches in several aspects. In fact, associations between tags and resources are updated also while the system is used and not only when the resource is added to the system. Moreover, similarity between tags is calculated considering how many times tags have been associated to each resource and not performing a random walk on the graph.

In [7], Givon and Lavrenko automatically annotate full text of books with social tags. The algorithm selects the keywords from the books using the tf-idf score, then builds a relevance model that contains the probability that a social tag $t$ can be assigned to a book *b* that contains keywords $w_1 ... w_n$. The model allows to choose the tags to recommend for a book. The algorithm uses a content-based system, while *RATC* focuses on associating tags to a resource while the system is used.

In [13], tag recommendations for Flickr photos, based on tag co-occurrences, are produced. In order to select the tags to recommend, the authors use a "promotion function" that selects the most descriptive tags, in order to make the recommendations. The function is based on a number of observations on users tagging behavior done on a Flickr dataset. Even if co-occurrences of tags in resources are considered (like *RATC* does), we continuously and implicitly monitor the tagging behavior of users. Similarity between tags is not calculated using a promotion function, built with an observation of the tagging system at a certain time, but considering the use of the tags at the moment in which similarities are calculated.

## V. CONCLUSIONS AND FUTURE WORK

In this paper we proposed an approach able to cluster tags in a tagging system, which can be used to produce tag recommendations that facilitate the exploration of a tagging system. *RATC* has the capability to dynamically improve its performances while the tagging system is being used, by monitoring users activity and exploiting implicit feedbacks left by users. Experimental results highlight its effectiveness in presence of strongly related tags in a cluster.

Future work will involve the adoption of a multi-layered clustering algorithm that, for each tag, takes into account the different contexts in which a tag is used.

## ACKNOWLEDGMENT

This work is partially funded by Regione Sardegna under project CGM (Coarse Grain Recommendation) through Pacchetto Integrato di Agevolazione (PIA) 2008 "Industria Artigianato e Servizi".

## REFERENCES

[1] Xujuan Zhou, Yue Xu, Yuefeng Li, Audun Jøsang, and Clive Cox. "The state-of-the-art in personalized recommender systems for social networking". *Artif. Intell. Rev.*, 37(2):119–132, 2012.

[2] Scott A. Golder and Bernardo A. Huberman. "Usage patterns of collaborative tagging systems". *J. Information Science*, 32(2):198–208, 2006.

[3] Ido Guy and David Carmel. "Social recommender systems". In *Proceedings of the 20th International Conference on World Wide Web, WWW 2011*, pages 283–284. ACM, 2011.

[4] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. "Tag recommendations based on tensor dimensionality reduction". In *Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys 2008*, pages 43–50. ACM, 2008.

[5] Steffen Rendle and Lars Schmidt-Thieme. "Pairwise interaction tensor factorization for personalized tag recommendation". In Proceedings *of the Third International Conference on Web Search and Web Data Mining, WSDM 2010*, pages 81–90. ACM, 2010.

[6] David Carmel, Haggai Roitman, and Elad Yom-Tov. "Social bookmark weighting for search and recommendation". *VLDB J.*, 19(6):761–775, 2010.

[7] Sharon Givon and Victor Lavrenko. "Large scale book annotation with social tags". In *Proceedings of the Third International* Conference *on Weblogs and Social Media, ICWSM 2009*. The AAAI Press, 2009.

[8] Kai Bielenberg and Marc Zacher. "Groups in social software: Utilizing tagging to integrate individual contexts for social navigation". Digital *Media. Bremen, Germeny, University Bremen. Master of Science in Digital Media*, 120, 2005.

[9] Grigory Begelman, Philipp Keller, and Frank Smadja. "Automated tag clustering: Improving search and exploration in the tag space". In Proceedings *of the WWW Collaborative Web Tagging Workshop*, Edinburgh, Scotland, 2006.

[10] Ludovico Boratto, Salvatore Carta, and Eloisa Vargiu. "RATC: A robust automated tag clustering technique". In *E-Commerce and Web Technologies, 10th International Conference, EC-Web 2009. Proceedings*, volume 5692 of Lecture Notes in Computer Science, pages 324–335. Springer, 2009.

[11] Stijn van Dongen. "Graph Clustering by Flow Simulation". *PhD thesis*, University of Utrecht, 2000.

[12] Salvatore Carta, Andrea Alimonda, Maria Laura Clemente, and Maurizio Agelli. "Glue: Improving tag-based contents retrieval exploiting implicit user feedback". In *Proceedings Of The 8th Dutch-Belgian Information Retrieval Workshop (DIR 2008)*, pages 29–35, 2008.

[13] Börkur Sigurbjörnsson and Roelof van Zwol. "Flickr tag recommendation based on collective knowledge". In *Proceedings of the 17th* International *Conference on World Wide Web, WWW 2008*, pages 327–336. ACM, 2008.

[14] M. E. J. Newman and M. Girvan. "Finding and evaluating community structure in networks", August 2003.

[15] Fabrizio Sebastiani. "Machine learning in automated text categorization". *ACM Comput. Surv.*, 34(1):1–47, 2002.

[16] Lucia Specia and Enrico Motta. "Integrating folksonomies with the semantic web". In *The Semantic Web: Research and Applications, 4th European Semantic Web Conference, ESWC 2007, Proceedings*, volume 4519 of Lecture Notes in Computer Science, pages 624–

639. Springer, 2007.

[17] Masahiro Hamasaki, Yutaka Matsuo, Takuichi Nisimura, and Hideaki Takeda. “Ontology extraction using social network”. In International *Workshop on Semantic Web for Collaborative Knowledge Acquisition (SWECKA2007)*, 2007.

[18] Xian Wu, Lei Zhang, and Yong Yu. “Exploring social annotations for the semantic web”. In *Proceedings of the 15th international conference on World Wide Web, WWW 2006*, pages 417–426. ACM, 2006.

[19] Eirini Giannakidou, Vassiliki A. Koutsonikola, Athena Vakali, and Yiannis Kompatsiaris. “Co-clustering tags and social data sources”. In *The Ninth International Conference on Web-Age Information Management, WAIM 2008*, pages 317–324. IEEE, 2008.

[20] Inderjit S. Dhillon. “Co-clustering documents and words using bipartite spectral graph partitioning”. In *Proceedings of the seventh ACM* SIGKDD *international conference on Knowledge discovery and data mining*, pages 269–274. ACM, 2001.

[21] Ricardo A. Baeza-Yates. “Applications of web query mining”. In *Advances in Information Retrieval, 27th European Conference on IR* Research*, ECIR 2005, Proceedings*, volume 3408 of Lecture Notes in Computer Science, pages 7–22. Springer, 2005.

[22] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. “Folkrank: A ranking algorithm for folksonomies”. In Workshop *Information Retrieval 2006 of the Special Interest Group Information Retrieval (FGIR 2006)*, pages 111–114, 2006.

[23] Robert Jäschke, Leandro Balby Marinho, Andreas Hotho, Lars Schmidt-Thieme, and Gerd Stumme. “Tag recommendations in folksonomies”. In *LWA 2007: Lernen - Wissen - Adaption, Halle, September 2007, Workshop Proceedings*, pages 13–20. Martin-Luther-University Halle-Wittenberg, 2007.