FOLKSONOMY-BASED TAG RECOMMENDATION FOR ONLINE AUDIO CLIP SHARING

Frederic Font[1], **Joan Serrà**[2] **and Xavier Serra**[1]
[1]Music Technology Gorup, Universitat Pompeu Fabra, Barcelona, Spain
[2]Artificial Intelligence Research Institute (IIIA-CSIC), Bellaterra, Barcelona, Spain
frederic.font@upf.edu, jserra@iiia.csic.es, xavier.serra@upf.edu

ABSTRACT

Collaborative tagging has emerged as an efficient way to semantically describe online resources shared by a community of users. However, tag descriptions present some drawbacks such as tag scarcity or concept inconsistencies. In these situations, tag recommendation strategies can help users in adding meaningful tags to the resources being described. Freesound is an online audio clip sharing site that uses collaborative tagging to describe a collection of more than 140,000 sound samples. In this paper we propose four algorithm variants for tag recommendation based on tag co-occurrence in the Freesound folksonomy. On the basis of removing a number of tags that have to be later predicted by the algorithms, we find that using ranks instead of raw tag similarities produces statistically significant improvements. Moreover, we show how specific strategies for selecting the appropriate number of tags to be recommended can significantly improve algorithms' performance. These two aspects provide insight into some of the most basic components of tag recommendation systems, and we plan to exploit them in future real-world deployments.

1. INTRODUCTION

Online platforms where people share user generated content have stressed the need for efficient methods to describe and retrieve such content. Freesound [1] is an online audio clip sharing site which clearly reflects this need. It contains more than two million users and 140,000 user-contributed sound samples covering a wide variety of sounds (from field recordings and sound effects to drum loops and instrument samples), which have to be well described to allow proper retrieval.

In recent years, collaborative tagging has emerged as an efficient way to describe online resources shared by a community of users. In collaborative tagging systems, users describe information items by annotating them with a number of "free-form" semantically-meaningful textual labels, called tags, that act as keywords and that can be later used for retrieval purposes. A collection of tags together with their associations to content resources is commonly known as a *folksonomy*.

In the majority of collaborative tagging systems, including Freesound, users are not constrained by any particular number of tags to assign, nor by the use of any specific vocabulary where to pick the tags from. Therefore, descriptions can be done at many different levels of detail and accuracy. Description inconsistencies can then arise due to the ambiguity of tag meanings, tag scarcity, the use of personal naming conventions, typographical errors, or even the use of different languages [2].

One strategy for trying to overcome some of these problems, and thus obtain more comprehensive and consistent descriptions, has been the use of tag recommendation systems to help users in the tagging process. These systems analyze the first (usually few) tags that users introduce when describing a particular item, and quickly suggest new tags that can also be meaningful or relevant for the item being described. The same algorithms for tag recommendation can be used, in an off-line mode, to extend the descriptions of information items by analyzing their tags and automatically adding new ones (what is normally called tag propagation).

In this paper we present and evaluate four variants of an algorithm for tag recommendation in Freesound. Our recommendation is based on tag semantic similarity derived from tag co-occurrence in the Freesound folksonomy. A novel aspect of the algorithm is a step focused on automatically selecting the number of tags to recommend given a sorted list of candidate tags. Tag propagation methods found in related work (see below) do not perform this step, and usually evaluate algorithms at different values of N recommended tags. We compare our algorithm with simpler versions which either always recommend a fixed number of tags, or only recommend tags that are repeated in the list of candidates.

The rest of the paper is organized as follows. In Sec. 2 we review the related work and in Sec. 3 we briefly describe the Freesound folksonomy. Sec. 4 explains the proposed algorithm for tag recommendation. Secs. 5 and 6 describe the evaluation methodology and present the obtained results. In Sec. 7 we conclude the paper with a discussion about our findings and future work.

2. RELATED WORK

Collaborative tagging has been widely researched in the last few years. Some studies focus on a general description of the dynamics of collaborative tagging and user behavior when tagging [2–5]. Other studies have looked at the motivations that users have at the moment of tagging, proposing then automatic tag classification methods to organize types of tags according to these motivations [6, 7]. Most of the work done in the analysis of collaborative tagging systems takes as case studies well-known sites such as Delicious (bookmark sharing), CiteULike (scientific reference sharing) and Flickr (photo sharing).

A variety of methods have been proposed for tag propagation and tag recommendation, especially for the case of image annotation. In [5] and [8], content analysis of images is used to obtain similar images and then propagate or recommend tags from these images to the source. Instead of using content analysis, Sigurbjörnsson and Zwol [9] propose a method for image tag recommendation based on tag similarities derived from a folksonomy. Their approach is similar to the one we describe in this paper, though they use different strategies for sorting candidate tags and do not perform a final selection of the number of tags to recommend. In [10] and [11], more complex strategies for tag recommendation based on folksonomies are described and evaluated with data from Delicious, BibSonony and Last.fm (using hierarchical tag structures [10] and the *FolkRank* ranking algorithm [11]). Again, none of these approaches performs any selection of the number of tags to recommend.

In the field of sound and music, most of the work related with tag propagation or recommendation is not based on folksonomy analysis, but on the extraction of content-based audio features that can later be used to annotate songs with labels or tags (which is more commonly known as semantic annotation). Sordo [12] describes a method based on audio content similarity for propagating tags (related with style and mood) between acoustically similar songs. Martínez et al. [13] use a similar idea for automatically propagate tags in scarcely annotated samples of Freesound. In [14] and [15], a different approach for automatic annotation of music and sound effects is described, which is based on using machine learning techniques to learn mappings between tags and audio features. Due to the content-based nature of these strategies, they are not directly comparable to the approach we propose in this paper.

3. FREESOUND FOLKSONOMY

In Freesound, users can upload sound samples and then describe them with as many tags as they feel appropriate [1]. For building the folksonomy we use in our experiments, we considered user annotations between April 2005 and September 2011. The folksonomy includes a total of 785,466 annotations that assign 30,985 unique tags (not necessarily semantically unique, but with different string representations) to 118,620 sounds. As opposite to other well studied collaborative tagging systems such as Delicious or CiteULike, Freesound has what is called a *narrow folksonomy* [16], meaning that sound annotations are shared among all users and therefore one single tag can only be assigned once to a particular sound (e.g. the tag `field-recording` cannot be added twice to the same sound).

[1] Since a recent software upgrade, Freesound requires a minimum of three tags to annotate a sound. However, the data we analyze is prior to the introduction of this requirement.

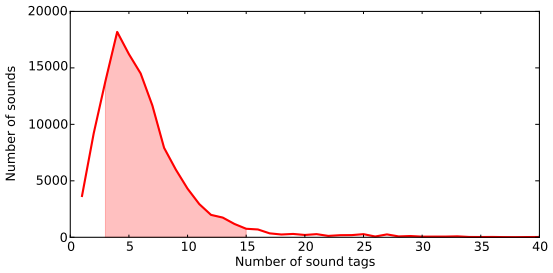

Figure 1: Distribution of sounds per number of tags. The global average of tags per sound is 6.16 and the standard deviation is 6.23.

Fig. 1 shows the distribution of the number of tags per sound in Freesound. We are particularly interested in recommending tags for the sounds that fall in the range of $[3, 15]$ tags (shadowed zone in Fig. 1), which are more than 80% of the total. The reason for focusing on these sounds is that the algorithm variants we present take as input the tags that have already been assigned to a sound. We consider 3 tags as enough input information for our algorithms to provide good recommendations. For sounds with less tags, content-based strategies such as the ones outlined in Sec. 2 are probably more suitable. On the other hand, sounds with more than 15 tags are, in general, enough well described.

Among the total number of 30,985 unique tags present in the folksonomy, we have applied a threshold to take only into consideration the tags that have been used at least 10 times, i.e. the tags that appear on at least 10 different sounds. By this we assume that tags that have been used less than 10 times are irrelevant for our purposes. In addition, by discarding less frequent tags, we reduce the computational complexity of the calculations described in Sec. 4.1. After applying this threshold, we are left with 6,232 unique tags, representing 20% of the total. Nonetheless, 93% of all annotations associate one of these 6,232 unique tags with a sound, thus we still take into account the vast majority of the original information.

4. PROPOSED ALGORITHM

The tag recommendation algorithm described in this paper consists of the three steps depicted in the diagram of Fig. 3. Variants are obtained by combining the different strategies proposed for the second and third steps. Feeding the algorithm variants with a number of $inputTags$, they output a set of $recommendedTags$. In the following subsections we describe each one of the depicted steps.

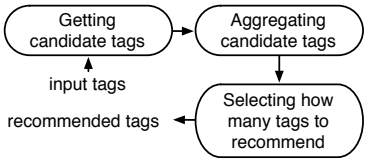

Figure 3: Block diagram of the described tag recommendation algorithm.

4.1 Getting candidate tags

The first step in the recommendation processes is getting a number of candidate tags according to the set of $inputTags$. For this purpose we build a tag similarity matrix based on tag co-occurrences in sound descriptions, following the Actor-Concept-Instance model proposed by Mika [17]. This tag similarity matrix gathers information from the whole folksonomy and only needs to be computed once.

The Actor-Concept-Instance model proposes to represent a folksonomy F as a tripartite hypergraph $H(F) = \langle V, E \rangle$, where vertices are given by three finite sets of objects, $V = U \cup T \cup R$ (U, T, and R denoting users, tags, and resources, respectively), and each edge represents a tag-resource association done by a user $E = \{\{u, t, r\} | (u, t, r) \in F\}$. We unfold this tripartite hypergraph into the bipartite graph TR, which reflects only the associations between tags and resources (sounds in our case). We can represent the bipartite graph TR as a matrix $D = \{d_{ij}\}$, where $d_{ij} = 1$ if tag t_i has been used to label resource r_j (otherwise $d_{ij} = 0$). We can then define the matrix $S = DD'$, which corresponds to a one-mode network connecting tags on the basis of shared resources. Elements s_{ij} of S indicate the number of sounds in which tags t_i and t_j appear together. Therefore, the diagonal of S represents the total number of different sounds labeled with a tag $t_{i=j}$. We then normalize S by dividing each element by $\sqrt{s_{ii}}\sqrt{s_{jj}}$. In this manner, we obtain the cosine similarity measures between tags t_i and t_j (the cosine similarity between the i-th and the j-th rows of D). Preliminary experiments using other distances such as Jaccard reported worse results. Furthermore, cosine similarity has been widely used in the literature and has been shown to be effective as a semantic relatedness measure in folksonomies [18]. Fig. 2 shows a graph visualization of an excerpt of S.

Having calculated the tag similarity matrix S, we iterate over $inputTags$ and get, for each element i, a set of candidates $C_{inputTag_i}$. For that we select the N most similar tags of $inputTag_i$ (i.e. the N most similar graph neighbors). We keep these similarity values for further processing in the following steps. In all our experiments we use $N = 100$. Hence, for instance, if our algorithm is feeded with three input tags, it will get a maximum of 300 candidate tags (provided that all three input tags have at least 100 neighbors). In preliminary experiments we observed that using values of $N > 100$ did not alter our recommendation results.

4.2 Aggregating candidate tags

The next step of our algorithm is to aggregate and sort the obtained candidate tags into a single list C_{sorted}. For this purpose we propose two different strategies which are now described.

4.2.1 Similarity-based strategy

In the first strategy we construct C_{sorted} by aggregating all sets of candidate tags $C_{inputTag_i}$ into a single list C_{raw}, and then ordering them by their similarity values (taken from the previous steps). As we do not want to recommend tags that are already part of $inputTags$, we remove any occurrences of these tags in C_{raw}. If there are repeated candidate tags in C_{raw}, their similarity values are added as a way of promoting these tags that appear in $C_{inputTag_i}$ of more than one input tag i. We finally normalize the similarity values by dividing them by the number of $inputTags$ (thus maintaining values in the original range).

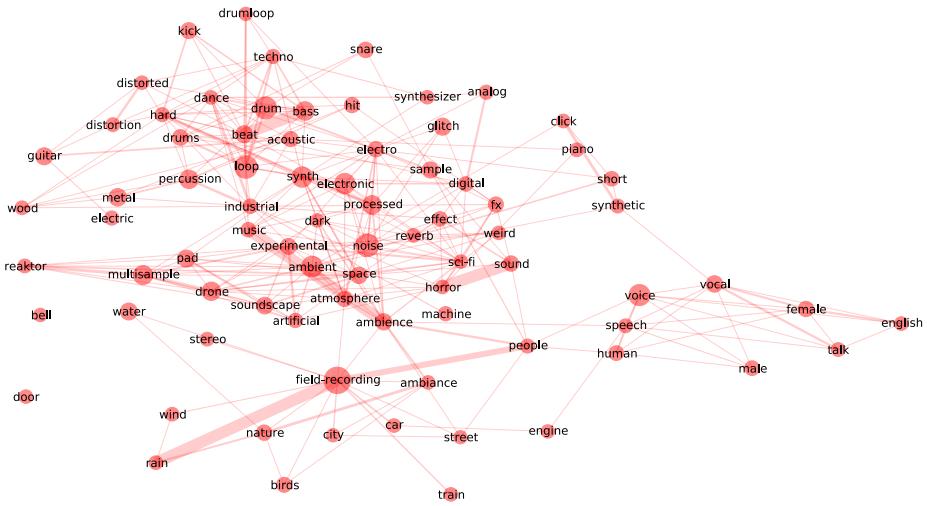

Figure 2: Graph visualization of the tag similarity matrix S. Edge widths represent the similarity between two tags. Node size is a logarithmic function of the absolute tag frequency. For the sake of clarity, only edges above a certain threshold and tags above a certain level of absolute frequency are shown.

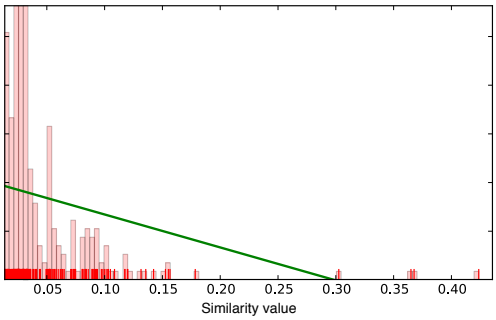

Figure 4: Example of the linear regression strategy for selecting how many tags to recommend. The straight line shows the linear regression of the histogram. Recommended tags are those placed at the right of the point where the linear regression crosses the y-axis.

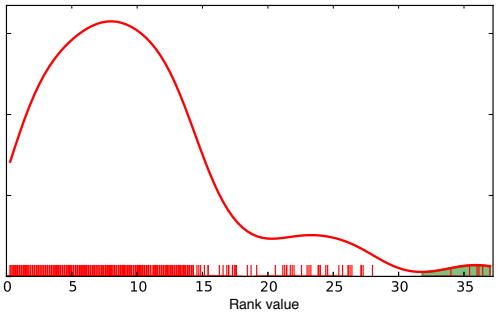

Figure 5: Example of the statistical test strategy for selecting how many tags to recommend. The curve represents the estimated PDF of C_{sorted}. Markers on the x-axis show the actual positions of candidate tags. Recommended tags are those under the shaded zone in the right.

4.2.2 Rank-based strategy

The second strategy is based on assigning a rank value to each candidate tag. For this purpose, we sort each set of $C_{inputTag_i}$ and then assign rank values as:

$$rank(neighbor_n) = N - (n-1),$$

where n is the position of the neighbor in $C_{inputTag_i}$ (thus n ranges from 1 to N). This way, the closest tag to every input tag will be assigned with a rank value of N. We then perform the aggregation as we would do in the similarity-based strategy, but using the assigned rank values as similarities.

4.3 Selecting how many tags to recommend

Once we have computed C_{sorted} (either using similarity-based or rank-based strategies), we select how many of these tags should be outputted as $recommendedTags$. Our approach is based on the hypothesis that given the distribution of similarity or rank values in C_{sorted}, it will be possible to determine a threshold that separates a set of meaningful tags from the other candidates. That is to say, that "good" tags for recommendation will appear as an isolated group from the rest of the distribution. In order to automatically determine this threshold, we again propose two different strategies.

4.3.1 Linear regression strategy

The first strategy consists in calculating the least-squares linear regression of the histogram of C_{sorted}. The threshold is set to the point where the linear regression crosses the y-axis. Fig. 4 shows an example of using this strategy with a histogram of tag similarity values. In that case threshold is set at 0.29. Therefore, all candidate tags with a similarity value higher than 0.29 would be outputted as $recommendedTags$.

4.3.2 Statistical test strategy

The second strategy has two steps. First, we estimate the probability density function (PDF) of C_{sorted}. For that purpose, we use a kernel density estimator [19]. Second, we iteratively take consecutive samples of the PDF (starting from the side where the candidates with highest rank or similarity lay) and perform a statistical test for normality. For that purpose we use the Anderson-Darling test [20]. The threshold is set at the point of the PDF where the test fails for the first time (i.e.the probability of having an independent Gaussian distribution is not statistically significant). The idea behind this process is that there will be a set of good tags for the recommendation that will exhibit a normal, independent distribution separated from the rest of candidate tags. The statistical test fails when it detects departures from normality, and according to our hypothesis this happens when non-meaningful candidate tags start affecting the PDF. Fig. 5 shows an example of applying this strategy using rank values for C_{sorted}. All tags under the shaded zone in the right will be recommended.

5. EVALUATION METHODOLOGY

In order to compare and evaluate the efficacy of the proposed variants we followed a systematic approach based on removing a number of tags from the Freesound sound descriptions and then trying to predict them. The advantage of this approach is that it allows us to quickly evaluate different tag recommendation methods without the need of human input. The main drawback is that tags that could be subjectively considered as good recommendations for a particular sound description but that are not present in the set of removed tags, will not count as positive results (see Sec. 7).

We performed a 10-fold cross validation following the methodology described in [21]. For each fold, we build a tag similarity matrix using the subset of the folksonomy corresponding to the training set of sounds. Then, for each one of the sounds in the evaluation set, we remove a random number of their tags ($removedTags$) and run tag recommendation methods using the tag similarity matrix derived from the training set. We compute standard precision, recall and f-measure for each evaluated sound according to:

Method name	Aggregation step	Selection step
Proposed algorithm variants		
RankST	Rank-based	Statistical test
SimST	Similarity-based	Statistical test
RankLR	Rank-based	Linear regression
SimLR	Similarity-based	Linear regression
Basic methods		
RankFIX@K	Rank-based	Fixed number ($K \in [1, 10]$)
SimFIX@K	Similarity-based	Fixed number ($K \in [1, 10]$)
Repeated@R	Repeated tags in C_{raw} ($R \in [2, 6]$)	
Random baselines		
Random (for every method)	Random selection of tags from C_{raw}, with the same length as $recommendedTags$	

Table 1: Evaluated tag recommendation methods.

$$precision = \frac{|recommendedTags \cap removedTags|}{|recommendedTags|},$$

$$recall = \frac{|recommendedTags \cap removedTags|}{|removedTags|}, \text{and}$$

$$f_{measure} = 2 \cdot \frac{precision \cdot recall}{precision + recall}.$$

Table 1 shows the tag recommendation methods that we compare. The first group of methods (*Proposed algorithm variants*) are the four possible combinations of aggregation and selection strategies described in Secs. 4.2 and4.3. *Basic methods* correspond to more basic tag recommendation methods that we used for comparison. On the one hand, we compare with two simpler versions of our proposed algorithm (RankFIX@K and SimFIX@K) which skip the last step of the recommendation process and always recommend a fixed number of K tags. We run these methods for values of K ranging from 1 to 10. On the other hand, we compare with an even simpler method (Repeated@R), which only recommends tags that appear more than R times in C_{raw} (independently of any rank or similarity values). We run these methods for values of R ranging from 2 to 6. Finally, we also compute a random baseline for each one of the previous methods by replacing the set of $recommendedTags$ with a random selection (of the same length) taken from C_{raw}. We choose as the general random baseline the one that gets the highest f-measure.

6. RESULTS

Table 2 shows the results of our evaluation as described in the previous section. The first group of results (under the label *with input tags range filter*) has been obtained by limiting the number of input tags we used to feed our algorithms to the range of $[3, 15]$, thus avoiding scarcely tagged sounds. The second group of results does not apply any restriction to the number of input tags (provided that there is at least one input tag).

A first observation is that using the input tags range filter produces an average increase in f-measure of 0.150 among our proposed algorithm variants (statistically significant using pairwise Kruskal-Wallis test, p≈0). Other methods present an average increase of 0.074 (p≈0). This means that, as expected, our algorithm works better in the range of $[3, 15]$ input tags. This is due to the notable increase in recall when using the input tags range filter (bigger than the increase in precision), suggesting that when feeded with at least three tags, our algorithm is able to select more relevant candidates. That supports the idea that for sounds with less tags, content-based approaches for tag recommendation would probably be more appropriate.

Algorithm	Precision	Rrecall	F-measure
With input tags range filter (83,010 sounds)			
RankST	0.443	0.537	0.432
RankLR	0.394	0.564	0.419
RankFIX@2	0.395	0.466	0.391
SimLR	0.348	0.397	0.325
SimST	0.381	0.333	0.317
RankFIX@5	0.233	0.614	0.308
SimFIX@2	0.303	0.344	0.294
SimFIX@5	0.181	0.467	0.237
Repeated@3	0.177	0.679	0.236
RankFIX@10	0.136	0.696	0.212
SimFIX@10	0.111	0.566	0.173
Random	0.006	0.033	0.010
Without input tags range filter (118,620 sounds)			
RankST	0.317	0.290	0.258
RankFIX@2	0.310	0.246	0.244
RankFIX@5	0.214	0.366	0.238
RankLR	0.236	0.301	0.221
SimLR	0.271	0.223	0.212
SimST	0.294	0.195	0.202
SimFIX@2	0.256	0.195	0.196
SimFIX@5	0.176	0.294	0.193
RankFIX@10	0.142	0.447	0.192
SimFIX@10	0.120	0.371	0.161
Repeated@3	0.095	0.262	0.110
Random	0.020	0.054	0.026

Table 2: Average of precision, recall and f-measure results for the evaluated tag recommendation methods. For the sake of readability, we only show some representative results of FIX and Repeated methods using $K = 2, 5, 10$ and $R = 3$. Methods are ordered by f-measure.

We can also see that all methods using the rank-based strategy for aggregating candidate tags always report higher f-measure than their similarity-based counterparts. Among the group with the input tags range filter, the average increase is of 0.104 (p≈0), while in the group without the filter the increase is of 0.033 (p≈0). That difference between both groups suggests that rank-based strategy works better when aggregating longer sets of candidates.

Regarding the selection step of our algorithm, both using the statistical test (ST) or the linear regression (LR) strategy significantly improves the performance with respect to the basic methods. When using the input tags filter, we observe an average increase in f-measure of 0.114 and 0.111 for the ST and LR methods, respectively (p≈0). Without using the filter the average increase is less important, of 0.045 and 0.036 for ST and LR, respectively (p≈0). It is surprising that methods recommending a fixed number of two tags (FIX@2) perform quite close to their counterparts using ST or LR strategies (and even in some cases scoring higher when not using the input tags range filer). This might be due to the fact that the average number of removed tags among all the experiments is 2.5. There-

Sound id	Input tags	Removed tags	Recommended tags	F-measure
8780	analog, glitch, warped	lofi	noise, electronic	0.0
124021	newspaper, reading, paper, page, news	read	magazine	0.0
38006	hit, glass, oneshot	percussion	singlehit, singlebeat, single, tap, hits, house, **percussion**, place, thuds, drum, plock	0.17
54374	spring, nightingale, nature, bird	field-recording, birdsong, binaural	birds, **field-recording**, forest, **birdsong**	0.5
78282	metal, medium-loud, interaction	impact	**impact**, wood	0.67

Table 3: Example of tag recommendations using the method RankST. Corresponding sounds can be listened at the following url: `http://www.freesound.org/search?q=`[Sound id].

fore, precision errors are minimized when recommending that amount of tags. Moreover, the good performance of FIX@2 reflects the effectiveness of the aggregation strategies, that successfully promote the most relevant tags on the first positions. On the other hand, ST and LR strategies perform generally better while at the same time recommending more tags (average of 3.16 and 4.6 respectively). This suggests that the selection step is able to choose, for each sound, the appropriate number of tags to recommend. Overall, the method that reports the highest f-measure is RankST (both with and without the input tags filter), and all our proposed algorithm variants perform much better than the random baseline.

7. CONCLUSION AND FUTURE WORK

In this paper we have described and evaluated four algorithm variants for tag recommendation based on the Freesound folksonomy. We have found that using a ranking instead of raw tag similarity values for sorting a list of candidate tags produces significantly better recommendations. The most novel aspect of the described algorithm is a step focused in automatically selecting the number of tags to recommend from a sorted list of candidate tags. The two strategies proposed for this step (statistical test and linear regression) have proved to be effective and statistically significantly increase the performance of the algorithm.

Although the systematic evaluation we have conducted allowed us to compare the different tag recommendation methods using a lot of sounds, the results in terms of f-measure are probably much worse than what a user-based evaluation could have reported. To exemplify this observation, Table 3 shows a few examples of tag recommendations performed using the RankST method (the one with the highest f-measure). We have marked in bold the tags that are considered good recommendations under our evaluation framework. Notice that many of the recommended tags which are not in italics could also be judged as meaningful recommendations if we listen to the sounds. In future work we would like to perform some user-based evaluation. Additionally, we plan to further improve our tag recommendation algorithm by introducing more tag-specific information such as characterizations of tag relevance, semantic category or usage context. Finally, we also plan to include our tag recommendation system in future deployments of Freesound.

8. ACKNOWLEDGEMENTS

This work is partially supported under BES-2010-037309 FPI grant from the Spanish Ministry of Science and Innovation for the TIN2009- 14247-C02-01 DRIMS project. JS acknowledges 2009-SGR-1434 from Generalitat de Catalunya, JAEDOC069/2010 from Consejo Superior de Investigaciones Científicas, TIN2009-13692-C03-01 from the Spanish Government, and EU Feder Funds.

9. REFERENCES

[1] V. Akkermans et al.:"Freesound 2: An Improved Platform for Sharing Audio Clips," *Late-braking demo abstract of the Int. Soc. for Music Information Retrieval Conf.*, 2011.

[2] H. Halpin et al.: "The dynamics and semantics of collaborative tagging," *Proc. of the 1st Semantic Authoring and Annotation Workshop*, 1-21, 2006.

[3] S. A. Golder and B. A. Huberman: "Usage patterns of collaborative tagging systems," *Journal of Information Science*, 32(2), 198-208, 2011.

[4] C. Marlow et al.: "HT06, Tagging Paper, Taxonomy, Flickr, Academic Article, ToRead," *Proc. of the 17th Conf. on Hypertext and Hypermedia*, 31-40, 2006.

[5] U. Farooq et al.: "Evaluating Tagging Behavior in Social Bookmarking Systems: Metrics and design heuristics," *Human-Computer Interaction*, 1, 351-360, 2007.

[6] K. Bischoff et al.:"Can all tags be used for search?," *Proc. of the 17th ACM Conf. on Information and Knowledge Management*, 32(2), 193202. ACM, 2008.

[7] I. Cantador et al.: "Categorising social tags to improve folksonomy-based recommendations," *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(1), 1-15, 2011.

[8] I. Ivanov et al.:"Object-based tag propagation for semi-automatic annotation of images," *Proc. of the Int. Conf. on Multimedia Information Retrieval*, 497-506, 2010.

[9] B. Sigurbjörnsson and R. Van Zwol: "Flickr tag recommendation based on collective knowledge," *Proc. of the 17th Int. Conf. on World Wide Web*, 327-336, 2008.

[10] P. De Meo et al.: "Exploitation of semantic relationships and hierarchical data structures to support a user in his annotation and browsing activities in folksonomies," *Information Systems Journal*, 34(6), 511-535, 2009.

[11] R. Jäschke et al.: "Tag Recommendations in Folksonomies," *Knowledge Discovery in Databases PKDD*, 34(6), 506-514, 2009.

[12] M. Sordo: "Semantic Annotation of Music Collections: A Computational Approach," *PhD thesis*, Universitat Pompeu Fabra, 2012.

[13] E. Martínez et al.: "Extending the folksonomies of freesound.org using content-based audio analysis," *Proc. of the Sound and Music Computing Conf.*, 23-25, 2009.

[14] D. Turnbull et al.: "Semantic Annotation and Retrieval of Music and Sound Effects," *IEEE Transactions On Audio Speech And Language Processing*, 16, 467-476 2008.

[15] L. Barrington et al.: "Audio Information Retrieval using Semantic Similarity," *IEEE Int. Conf. on In Acoustics, Speech and Signal Processing*, 16, 725-728, 2007.

[16] T. Vander Wal: "Explaining and showing broad and narrow folksonomies," http://www.personalinfocloud.com/ 2005/02/explaining_and_.html, 2005.

[17] P. Mika: "Ontologies are Us: A Unified Model of Social Networks and Semantics," *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(1), 5-15, 2007.

[18] C. Cattuto et al.: "Semantic Analysis of Tag Similarity Measures in Collaborative Tagging Systems," *Data Engineering*, 805, 5-11, 2008.

[19] B.W. Silverman: "Density Estimation for Statistics and Data Analysis," *Applied Statistics*, 37(1), 1986.

[20] F. W. Scholz and M. A. Stephens: "K-Sample Anderson-Darling Tests," *Journal of the American Statistical Association*, 82, 918-924, 1987.

[21] S. L. Salzberg: "On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach ," *Data Mining and Knowledge Discovery*, 1(3), 317-328, 1997.