Folksonomies, the Semantic Web, and Movie Recommendation

Martin Szomszor[1], Ciro Cattuto[3,2], Harith Alani[1], Kieron O'Hara[1],
Andrea Baldassarri[2], Vittorio Loreto[2,3], Vito D.P. Servedio[2,3]

[1]School of Electronics and Computer Science
University of Southampton, SO17 1BJ, UK
[2]Dipartimento di Fisica, Università di Roma "La Sapienza"
P.le A. Moro, 2, 00185 Roma, Italy
[3]Museo Storico della Fisica e Centro Studi e Ricerche Enrico Fermi
Compendio Viminale, 00184 Roma, Italy

Abstract. While the Semantic Web has evolved to support the meaningful exchange of heterogeneous data through shared and controlled conceptualisations, Web 2.0 has demonstrated that large-scale community tagging sites can enrich the semantic web with readily accessible and valuable knowledge. In this paper, we investigate the integration of a movies folksonomy with a semantic knowledge base about user-movie rentals. The folksonomy is used to enrich the knowledge base with descriptions and categorisations of movie titles, and user interests and opinions. Using tags harvested from the Internet Movie Database, and movie rating data gathered by Netflix, we perform experiments to investigate the question that folksonomy-generated movie tag-clouds can be used to construct better user profiles that reflect a user's level of interest in different kinds of movies, and therefore, provide a basis for prediction of their rating for a previously unseen movie.

1   Introduction

Recommendation systems have evolved in recent years to support users in the discovery of new items through the construction of profiles that represent their interests, and networks that connect them to other users who share similar tastes. Many of these recommendation strategies rely on the modelling of intrinsic attributes about each item (e.g. the keywords for a document or the genre of a CD) so that the items can be categorised, and the level of interest a user has can be expressed in terms of these attributes. This knowledge is usually gathered over time, by monitoring and logging various user interactions with the system (e.g. buying, browsing, bookmarking). Amazon.com, for example, provides a recommendation service that is based on collaborative filtering: if a user buys an item that has been bought by a number of other users in combination with some other items, then those other items will be recommended by Amazon.com to the user. These recommendations are entirely based on what goes on inside the system (Amazon.com in this case), ignorant of any external knowledge about the items or the users themselves.

To improve on such recommendation techniques, we think it might be useful to incorporate data from as many sources as possible to build richer profiles that model many facets of interest that might be difficult and impractical to capture by a single system or service. In recent years, many Web 2.0 applications, such as folksonomies and blogs, have become popular places where individuals provide and share various type of information. This information may, directly or indirectly, represent the interests of those individual users. There could be much to learn about a user from analysing their shared profile in MySpace, bookmarks in del.iciou.us, photos in Flickr, references in Connotea, and any other popular Web 2.0 applications.

Although folksonomies provide structures that are considered to be formally weak or unmotivated, they do have two advantages in this particular context. First, they are strongly connected with the actual use of the terms in them and the resources they describe. And second, they are relatively cheap to develop and harvest, as they emerge from individual tagging decisions that are cheap for the user. To that extent, they may provide data about the *perceptions* of users, which is what counts in this particular recommendation context. In this respect, a folksonomy, we hypothesise, will be of greater value than an ontology of films, which might provide a more objective sense of whether two films are similar, but which need not map onto viewers' perceptions. However, it is important to note that the Semantic Web and folksonomies are not in competition here; folksonomies are not "cheaper" or "simpler" or "bottom up" versions of ontologies. As the system we will be experimenting with brings together data from a number of sources, then Semantic Web technology is certainly required.

In this paper, we test the hypothesis that folksonomies describing movie titles can be used as a basis for building recommendation profiles by associating each user with *tag-clouds* that represent their interests. By using Semantic Web technology to integrate heterogeneous data sources, a large collection of movie titles, ratings on these titles, and tagging data is assembled, providing the basis for empirical testing of our algorithms.

This paper is organised as follows: Section 2 contains a brief literature survey on tagging systems, the Semantic Web, and recommendation strategies, and is followed by a description of our architecture in Section 3. In Section 4, we present our recommendation algorithm that is based on the construction of interest tag-clouds before the results of our experiments are shown in Section 5. Finally, Section 6 contains our conclusions and directions for future work.

2 Background

2.1 Folksonomies

The term *folksonomy* was first coined by T. Vander Wal [23] to describe the taxonomy-like structures that emerge when large communities of users collectively tag resources. These *folk taxonomies* reflect a communal view of the attributes associated to items, essentially supplying a bottom-up categorisation of resources [14, 10]. Since individuals from different communities utilise different

tags, often reflecting their degree of knowledge in the domain, folksonomies can support highly personalised searching and navigation. For example, an article in the social bookmarking site `del.icio.us` [7] concerning web programming may have the tags `programming`, `ajax`, `javascript`, `tutorial`, and `web2.0`. With tags describing resources at varying levels of granularity, users may seek out their desired resources using terms they are familiar with.

Examination [8] of social tagging sites, such as `del.icio.us`, has revealed a rich variety in the ways in which tags are used, allowing tags themselves to be categorised in a number of ways:

- Tags may be used to identify the topic of a resource using nouns and proper nouns such as `news`, `microsoft`, `vista`.
- To classify the type of resource (e.g. `book`, `blog`, `article`, `review`, `event`).
- To denote the qualities and characteristics of the item (e.g. `funny`, `useful`, and `cool`).
- A subset of tags, such as `mystuff`, `myphotos`, and `myfavourites`, reflect a notion of self reference, often used by individuals to organise their own resources.
- Much like self referencing tags, some tags are used by individuals for task organisation (e.g. `to read`, `job search`, and `to print`).

Another important aspect of tagging systems is how they operate. Marlow *et al* [13] provide an extensive classification of tagging systems that enables us to compare the benefits and deficiencies of different systems according to seven characteristics:

1. **Tagging rights**
   The permission a user has to tag resources can effect the properties of an emergent folksonomy. The spectrum of tagging permissions ranges from *self-tagging*; where users are only allowed to tag the resources they have created, to a tagging *free-for-all*; where users may tag any resources. Some compromise between the two may be supported, for example, by allowing users to tag resources created by those in their social network.

2. **Tagging Support**
   One important aspect of a tagging system is the way in which users assign tags to items. They may assign arbitrary tags without prompting (*blind tagging*), they may add tags while considering those already added to a particular resource (*viewable tagging*), or tags may be proposed (*suggestive tagging*). While it has been shown [8] that suggestive tagging results in faster convergence to a folksonomy, it is not clear whether it effects the quality or diversity.

3. **Aggregation**
   Tagging events may be recorded at different levels of granularity. For example, all tagging events may be uniquely logged, keeping track of all the tags assigned by all of the users (the *bag-model*). This method allows tag weighting to be derived to reflect the popularity of a given tag for a particular resource. On the other hand, a simple *set-model*, resource centric approach

may be used where a set of tags is maintained for each resource, meaning the popularity of assignment for each tag is unknown.

4. **Types of Object**
   The types of resource tagged allow us to distinguish different tagging systems. Popular systems include Web pages (`del.icio.us`), bibliographic data (`CiteULike`), blogs (`technorati`), images (`flickr`), video (`You Tube`), audio objects (`last.fm`), and movies (`imdb`, `movielens`).

5. **Sources of Material**
   Tagging systems may allow users to upload resources (e.g. `You Tube`), or resources may be managed by the system (e.g. `last.fm`, `imdb`). In some situations, such as `del.icio.us`, arbitrary Web resources may also be referenced.

6. **Resource Connectivity**
   Within a tagging system, resources may be connected independently of their tags. For example, Web pages may be connected via hyperlinks, or items may be grouped together (e.g. photo albums in `flickr`). When such linking occurs, additional analysis can reveal correlations between items that correspond with the co-occurrence of tags.

7. **Social Connectivity**
   Finally, it is useful to consider how users of the system may be connected. Many tagging systems include social networking facilities that allow users to connect themselves to each other based on their location, areas of interest, educational institutions and so forth. These social networks provide an excellent opportunity to explore the correlation between localised substructures in folksonomies and social connectivity.

2.2 Semantic Web

The Semantic Web (SW) has proven to be a useful data integration tool, facilitating the meaningful exchange of heterogeneous data, particularly in areas such as e-science and medicine. However, as is well known, there are costly overheads in the use of the SW; in particular, the effort involved in building, and maintaining, useful ontologies and acquiring rich and well structured RDF can be relatively high, a fact often blamed for slowing down the wide adoption of Semantic Web technology [5, 1]. Web 2.0, and the notion of community tagging, is showing promise as an alternative way to quickly and cheaply produce structured semantic models [9] through the study of emergent semantics [22]. It has been argued that harnessing the knowledge embedded in folksonomies can lead to building shallow ontologies that are more receptive to knowledge change over time [16].

Nevertheless, we should not think that Web 2.0 and the SW, tags and RDF, folksonomies and ontologies are competing for the same space [2]. Folksonomies are essentially a development in information retrieval, an interesting variant on the keyword-search theme. This makes them particularly interesting in the context of film recommendations: they help answer the question "how can I find films relevant to the concept in which I am interested." Ontologies are tools for

4

data integration: they are attempts to regulate part of the world of data, and to facilitate mappings and interactions between data held in disparate formats or locations.

The important question with respect to SW technology and Web 2.0 is not how to manage a trade-off, but rather, how to use them together for the best advantage. Much will depend on the particular context of use, but in the case of film recommendation, a fairly basic architecture suggests itself. The use of Web 2.0 data for the purpose of recommendation makes sense, as this emerges from tagging based on perceptions. Folksonomies, being organic structures that mirror the understanding users have of resources, can provide a better foundation for the expression of user's interests. This idea has been investigated in the context of social bookmarking [19] to build a Web Page recommender system and provided encouraging results.

Nevertheless, the hypothesis with which we are working is whether we can improve the performance of recommender systems by giving the systems access to greater quantities of information, which implies the need to integrate relevant data acquired from heterogeneous sources. This immediately suggests a role for SW technologies. As noted, the issues to be addressed in this part of the architecture include the developing a suitable ontology and acquiring RDF without driving up the cost of development.

2.3   Recommender Systems

Recommender systems are usually used in one of two contexts: (1) to help users locate items of interest they have not previously encountered, (2) to judge the degree of interest a user will have in item they have not rated. With the growing popularity of on-line shopping, E-commerce recommender systems [20] have matured into a fundamental technology to support the dissemination of goods and services. Much research has been undertaken to classify different recommendation strategies [6, 11], but for the purposes of this paper, we divide them broadly into two categories.

Collaborative recommendation is probably the most widely used and extensively studied technique that is founded on one simple premise: if user A is interested in items w, x, and y, and user B is interested in items w, x, y, and z, then it is likely that user A will also be interested in item z. In a collaborative recommender system, the ratings a user assigns to items is used to measure their commonality with other users who have also rated the same items The degree of interest for an unseen item can be deduced for a particular user by examining the ratings of their neighbours. It has been recognised that users interest may change over time, so time-based discounting methods have been developed [3, 21] to reflect changing interests.

Content-based recommendation represents the culmination of efforts by the information retrieval and knowledge representation communities. A set of attributes for the items in the system is conceived, such as the keywords and term frequencies for documents in a repository, so the system can build a profile for each user based on the attributes present in the items that user has rated highly.

The interest a user will have in an unrated item can then be deduced by calculating its similarity to their profile based on the attributes assigned to the item.

Such systems are not without their deficiencies, the most prominent of which arise when new items and new users are added to the system - commonly referred to as the *ramp-up* problem [12]. Since both content-based and collaborative recommender systems rely on ratings to build a user's profile of interest, new users with no ratings have neutral profiles. When new items are added to a collaborative recommender system, they will not be recommended until some users have rated them. Collaborative systems also depend on the overlap in ratings across users and perform badly when ratings are sparse (i.e. few users have rated the same items) because it is hard to find similar neighbours.

Hybrid recommender systems, i.e. those which make use of collaborative and content based approaches, have been developed to overcome some of these problems. For example, collaborative recommender systems do not perform well with respect to items that have not been rated, but content-based methods can be used to understand their relationship to other items. Hence, a mixture of the two approaches can be used to provide more robust systems. More recent recommender systems have also investigated the use of ontologies to represent user profiles [15]. Benefits of this approach are more intuitive profile visualisation and the discovery of interests through inferencing mechanisms.

3 Recommendation Architecture

To gather the information necessary to construct profiles that describe the kinds of movies a user is interested in, we combine data harvested from two sources, and also combine the use of Web 2.0 and SW technology. This section first presents the Web 2.0 data sources we use to construct a knowledge base about movies and how users rate movies (Section 3.1), and second the semantic technologies to represent the information in this knowledge base (Section 3.2).

3.1 Data Sources

For movie tagging data, we make use of the Internet Movie Database (IMDB) [25]; an online database containing extensive information on movies, actors, television shows, and production personnel. IMDB holds information on approximately 960,000 titles and 2,300,000 people, and is the largest known accumulation of data about films [24]. In terms of tagging, IMDB allows users to add *keywords* to titles to describe arbitrary features of the movie. Typically, these are used to denote important scenes in the film (e.g. `sword-fight`, `kidnapping`, `car-chase`), plot themes (e.g. `love`, `revenge`, `time-travel`), locations (e.g. `space`, `california`), film genres (e.g. `independent-film`, `non-fiction`, `cult-favorite`), and background data (e.g. `based-on-novel`, `based-on-true-story`). On average, a popular movie has between 50 and 150 keywords attached to it.

Currently, IMDB uses this tagging data to create a movie search tool that helps users to find popular movies based on their keywords. A screen shot of this interface in shown in Figure 1 and contains two panels: on the left, a tag cloud is

used to display keywords; and on the right, a list of the top movies that contain the currently viewed keywords. In this particular example, the keywords `space` and `android` are used as the search terms.

Fig. 1. A screen shot of the IMDB keyword search interface.

With respect to the tagging system categorisation presented earlier in Section 2.1, IMDB is a tagging *free-for-all*. Although the addition of keywords to a movie is moderated, it is used mainly to prevent spam attacks and not to manage the keywords used. When adding keywords to a movie, users can see the keywords that have already been added, but they are not prompted with suggestions (*viewable* tagging support). In terms of aggregation, IMDB falls into the set-model category because the individual keyword assignments by each user cannot be seen. Instead, a simple list of keywords is maintained for each movie and duplicates are not allowed.

To test our keyword-based recommendation approach, we use data provided by Netflix [17] as part of the Netflix Prize [18]. Netflix is an online DVD rental service, established in 1998, the provides a flat rate, mail-based, rental service to customers in the United States. Their current DVD collection contains around 75,000 titles, offered to a customer base of over 6 million individuals. After renting a movie, customers may enter their rating of the movie into the Netflix database via the website, using a discrete score from 1 to 5.

In October 2006, Netflix began a competition to find better recommendation systems, offering a grand prize of $1 million to anyone managing to improve on their own algorithm by 10%. To drive this competition, Netflix published a large set of movie rating data from their database featuring 480,189 customers and 100,480,507 ratings across 17,770 movie titles.

3.2 Data Representation

To combine the IMDB database and the Netflix rating data, we import both data sets into a standard relational database. String matching is then used to correlate the movie titles in the Netflix data dump with their counterparts in the IMDB data set, providing a way to retrieve IMDB keywords for each Netflix movie

title. To provide a homogeneous view over both data sources, an ontology is used in conjunction with the D2RQ [4] mapping technology, supplying a SPARQL end-point which can be queried to find extensive amounts of information on movies such as: the keywords assigned; the actors appearing in the film; the writers, directors and production crew; as well as rating information for movies featured in the Netflix data set. The two perceived issues with semantically-enabled technologies mentioned in 2.2 are thereby addresses. Instead of having to convert data to RDF triples, D2RQ allows this to be done on the fly. Within the well-structured domain of the system, the ontology was deliberately kept as lightweight as possible.

The ontology used is illustrated in Figure 2 where classes depicting IMDB data are shown in white boxes, and classes describing Netflix data are shown in grey boxes. The IMDB data set is centered around the concepts of *Movie*, *Person*, and *Role*. The movie class has properties describing the certificate information, keywords, rating data, and release date information. A Person is anyone who is associated with a movie, i.e. an actor or director, and a Role is used to define how a person is connected to a movie. This abstraction of roles allows the same person to have different functions for the same movie, for example, being a writer and director.
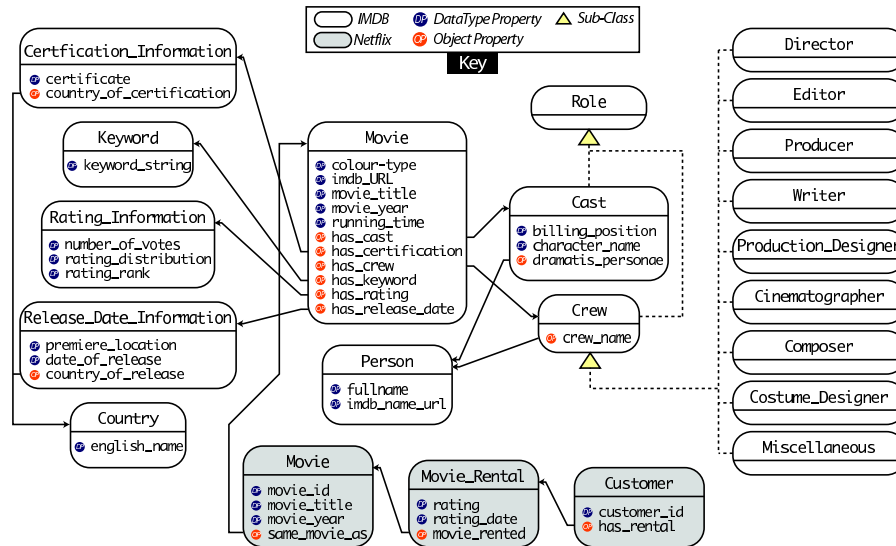

Fig. 2. The ontology used to integrate IMDB and Netflix data.

4 Recommendation Method

To explore the relationship between the way a user rates movies and the keywords that are assigned to movies, we have devised two prediction algorithms that guess

the rating a user would give to a previously unrated movie based on tag-clouds that depict their interests. For comparison, we also specify a naive average-rating algorithm were the average rating for a movie across all users is used as the predicted rating.

4.1 Notation

Let us denote a given user by $u \in U$, where U is the set of all users, a movie by $m \in M$, where M is the set of all available movies, and a rating value by the integer $r \in \{1, 2, 3, 4, 5\} \equiv R$. We indicate the set of movies rated by user u as M_u. On this set we define the rating function for user u as $f_u : m \in M_u \mapsto f_u(m) \in R$.

When keywords or tags are available for a movie m, we denote by K the global set of keywords, by K_m the set of keywords (or tags) associated with movie m, and by N_k the global frequency of occurrence of keyword k for all movies. We can then introduce a notion of *rating tag-cloud* $T_{u,r}$ for a given user u and rating r as the set of couples (k, n_k), where $k \in K$ indicates a keyword (or tag) and $n_k = n_k(u, r)$ is its frequency of occurrence for all movies that user u has associated with rating r. That is,

$$n_k(u, r) = |\{m \in M_u \mid k \in K_m \wedge f_u(m) = r\}| . \tag{1}$$

Two sample rating tag-clouds are shown in Figure 3; the left one is a rating 1 tag-cloud, and the right one is a rating 5 tag-cloud. The size of keywords is proportional to the logarithm of their frequency of occurrence in the tag-cloud they belong to.
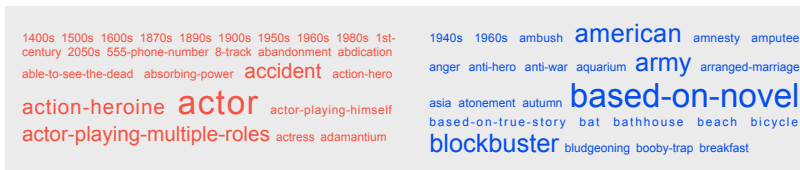

Fig. 3. Sample rating tag-clouds (left: rating 1, right: rating 5).

4.2 Average-based Rating

A very simple rating prediction strategy can be implemented by assuming that a given user u^* will rate a new movie m^* ($m^* \notin M_{u^*}$) according to the average rating that the movie received by all other users. We compute the average rating of movie m as

$$\bar{r}_m = \frac{1}{|U_m|} \sum_{u \in U_m} f_u(m) , \tag{2}$$

where $U_m = u \in U | m \in M_u$ is the set of users that have rated movie m, and $|U_m|$ is its cardinality. In this scheme, the predicted rating for movie m^* is the integer $r^* \in R$ that is nearest to $\bar{r}_{m^*}$.

4.3 Simple Tag-Cloud Comparison

In this scheme we guess the rating that user u^* would give to movie m^* by comparing the set of keywords K_{m^*} associated with the movie against the rating tag-clouds $T_{u^*,r}$ of user u^* for different ratings. We guess the rating r^* as the one corresponding to the tag-cloud (of user u^*) that most closely resembles the set of keywords K_{m^*}, as measured by the number of keywords that K_{m^*} shares with the tag-clouds of user u^* for different ratings:

$$\sigma(u^*, m^*, r) = |\{(k, n_k) \in T_{u^*,r} \,|\, k \in K_{m^*}\}| \,. \tag{3}$$

4.4 Weighted Tag-Cloud Comparison

In this hybrid scheme we try to take into account weights both at the keyword level (through their frequencies n_k) and at the tag-cloud level, though a measure of tag-cloud similarity. Given a new (in the sense of unrated) movie m^*, we consider the set of keywords K_{m^*} and introduce a notion of "similarity" between K_{m^*} and a given tag-cloud $T_{u,r}$. We define such a measure of similarity as:

$$\sigma(u, m, r) = \sum_{\{(k, n_k) \in T_{u,r} \,|\, k \in K_m\}} \frac{n_k}{\log(N_k)} \,, \tag{4}$$

that is we sum over all keywords which K_{m^*} and the tag-cloud $T_{u,r}$ have in common, and we weight each keyword k proportionally to its frequency n_k in the tag-cloud, and inversely proportional to the logarithm of its *global* frequency N_k, as commonly done in TFIDF term-weighting schemes.

We subsequently define the weighted average rating as

$$\bar{\sigma}(u, m) = \frac{1}{S(u, m)} \sum_{r \in R} r \, \sigma(u, m, r) \,, \tag{5}$$

where $S(u, m) = \sum_{r \in R} \sigma(u, m, r)$ is a normalization factor. Thus, $\bar{\sigma}(u, m)$ is an estimate of a user rating based on the weighted similarity between the set of movie keywords and the user's rating tagclouds (themselves weighted). This information can be used by itself, to guess a user rating, or it can be used to improve a prediction based on other techniques.

In our experiment we decided to use the rating $\bar{\sigma}(u, m)$, estimated from the tag-cloud similarity, to improve the simple rating estimate based on the per-movie average rating (see section 4.2). We combine the two estimates by computing their weighted average. That is, given a user u^* and a movie m^*, our estimate for the rating is

$$\sigma^*(u^*, m^*) = (1 - \gamma) \, \bar{r}_{m^*} + \gamma \, \bar{\sigma}(u^*, m^*) \,, \tag{6}$$

where $0 < \gamma < 1$ is a factor weighting the contribution of the two estimates. In our experiment we set $\gamma = 1/2$. We guess the rating r^* as the integer in R that lies closest to the weighted average $\sigma^*(u^*, m^*)$.

Of course, the above strategy can only be used when the set of keywords K_{m^*} associated with movie m^* is non-empty. If K_{m^*} is empty our implementation resorts to using the simple strategy of section 4.2 (equivalent to setting $\gamma = 0$ in Eq. 6).

10

5 Experiment and Results

To test the algorithms presented earlier in Section 4, we extract a training set from the full Netflix data dump containing the ratings of 500 randomly chosen users. For each user, a test set made up from their last 100 ratings is removed from the training set so the accuracy of our algorithms can be tested. For each user, the root mean squared error (RMSE) is recorded, along with the percentage of exactly matched ratings. Given a set of predicted ratings $\{r_i\}$ and the corresponding set of actual ratings $\{r_i^*\}$, the RMSE is defined as:

$$\text{RMSE}(\{r_i\}, \{r_i^*\}) = \sqrt{\frac{1}{N} \sum_i (r_i - r_i^*)^2}\,. \qquad (7)$$

A summary of the results follows:

	Average Rating	Unweighted	Weighted
Correct	36.12%	44.15%	42.47%
Incorrect	63.99%	55.85%	57.53%
RMSE	1,131	1.074	0.961

The unweighted tag-cloud comparison does perform better than the naive average rating, with a moderate increase in the percentage of correctly rated movies. Using the weighted tag cloud comparison improves the RMSE, but with a slight drop in the fraction of exactly matched ratings. Figure 4 contains two scatter plots (unweighted and weighted tag-cloud comparison techniques) showing the RMSE for each user against the number of movies in their training set. These plots show two interesting features: (i) the weighted comparison technique has a smaller error range than the unweighted comparison (ii) the error rate seems to be independent of the number of movies rated. To visualise the distribution of predicted ratings for each of the algorithms, we present two histograms in Figure 5: one showing the distributions of the predicted ratings, and
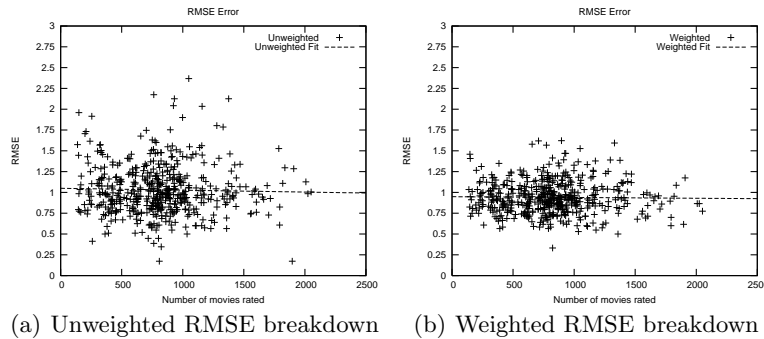

(a) Unweighted RMSE breakdown   (b) Weighted RMSE breakdown

Fig. 4. Scatter plots to show the level of accuracy for each rating technique in terms of the number of movies rated by the user.

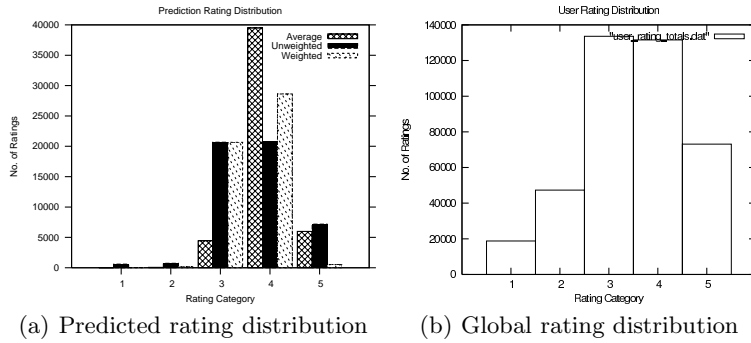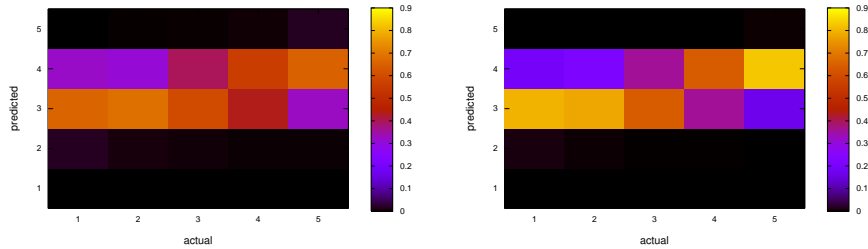(a) Predicted rating distribution   (b) Global rating distribution

Fig. 5. Histograms showing the number of predictions made in each rating category, and the overall rating distribution

one showing the global distribution of actual ratings. From these charts, it is clear that the rating categories 1 and 2 are being neglected.

In order to gain more insight into the behavior of our prediction schemes, we study the distribution of predicted ratings as a function of the actual rating. Fig. 6 shows the (color-coded) probability distribution of predicted ratings as a function of the actual movie rating, for the simple average-based scheme (left figure) and the weighted tag-cloud comparison scheme (right figure).

(a) Distribution of predicted ratings for average-based method

(b) Distribution of predicted rating for weighted tag-cloud comparison method

Fig. 6. Distribution of predicted ratings as a function of actual movie rating, for the simple average-based scheme (Figure 6(a)) and the weighted tag-cloud comparison scheme (Figure 6(b)). For each value of the actual rating (horizontal axis), a normalized histogram of the predicted ratings (vertical axis) was built, displaying how predicted values are distributed. Because of normalization, the sum of values along all columns is 1.

A perfect prediction scheme would appear as a unity matrix, with ones along the main diagonal and zeros elsewhere. Fig. 6 shows that both prediction schemes

behave poorly for low (1 and 2) and high (5) values of the actual rating, as both schemes predict intermediate ratings (3 and 4) with high probability, independent of the actual rating (bright rows in the plots).

We observe that the weighted tag-cloud scheme provides enhanced contrast throughout the rating range. For intermediate values of the actual rating (3 and 4) it improves significantly over the average-based scheme, with a better separation of the diagonal elements (3-3 and 4-4, correct predictions) over the off-diagonal ones, in particular over the elements corresponding to the incorrect predictions 3-4 and 4-3. For the highest actual rating (5) the weighted tag-cloud scheme features a distribution of predicted values which is more skewed towards high ratings, but on average it still fails to predict the correct rating. The same happens for low actual ratings (1 and 2), where the weighted tag-cloud scheme displays a distribution of predicted values which is more skewed towards low-values, but still fails to predict 1s and 2s with a significant probability.

In terms of future work, this evaluation shows that intermediate ratings are predicted rather well, and additional work is needed to make better prediction of extreme rating values, both high and low.

6   Conclusions and Future Work

In this paper, we have demonstrated that a movie recommendation system can be built purely on the keywords assigned to movie titles via collaborative tagging. By building different tag-clouds that express a user's degree of interest, a prediction for a previously unrated movie can be made based on the similarity of its keywords to those of the user's rating tag-clouds. With further work, we believe our recommendation algorithms can be improved by combining them with more traditional content-based recommender strategies. Since IMDB provides extensive information on the actors, directors, and writers of movies, as well as demographic breakdowns of the ratings, a more detailed profile can be constructed for each user. Also, our recommendation algorithms have not exploited any collaborative recommender techniques. Further research may show that rating tag-clouds are a useful and more efficient way to find neighbours with similar tastes.

References

1. Harith Alani, Yannis Kalfoglou, Kieron O'Hara, and Nigel Shadbolt. Towards a killer app for the semantic web. In *International Semantic Web Conference*, pages 829–843, 2005.
2. Tim Berners-Lee, Wendy Hall, James A. Hendler, Kieron O'Hara, Nigel Shadbolt, and Daniel J. Weitzner. A framework for web science. *Foundations and Trends in Web Science*, 1(1):1–134, 2006.
3. D. Billsus and M. Pazzani. User modeling for adaptive news access, 2000.
4. Chris Bizer, Richard Cyganiak, Jorg Garbers, and Oliver Maresch. D2RQ v0.5 - treating non-RDF relational databases as virtual RDF graphs. Technical report, Freie Universitat Berlin, 2006.

5. Elena Paslaru Bontas, Christoph Tempich, and York Sure. ONTOCOM: A cost estimation model for ontology engineering. In *Proceedings of the 5th International Semantic Web Conference (ISWC 2006)*, pages 625–639.

6. Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.

7. Del.icio.us: A social bookmarks manger homepage.

8. Scott Golder and Bernardo A. Huberman. The structure of collaborative tagging systems, Aug 2005.

9. Thomas Gruber. Ontology of folksonomy: A mash-up of apples and oranges, 2006.

10. Tony Hammond, Timo Hannay, Ben Lund, and Joanna Scott. Social bookmarking tools (i): A general review. *D-Lib Magazine*, 11, Apr 2005. 10.1045/april2005-hammond.

11. Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, January 2004.

12. J. A. Konstan, J. Reidl, A. Borchers, and J.L. Herlocker. Recommender systems: A grouplens perspective. In *Recommender Systems: Papers from the 1998 Workshop (AAAI Technical Report WS-98-08)*, pages 60–64. AAAI Press, 1998.

13. Cameron Marlow, Mor Naaman, Danah Boyd, and Marc Davis. Ht06, tagging paper, taxonomy, flickr, academic article, to read. In *HYPERTEXT '06: Proceedings of the seventeenth conference on Hypertext and hypermedia*, pages 31–40, New York, NY, USA, 2006. ACM Press.

14. Adam Mathes. Folksonomies – Cooperative Classification and Communication Through Shared Metadata, December 2004. http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html.

15. Stuart E. Middleton, Nigel R. Shadbolt, and David C. De Roure. Ontological user profiling in recommender systems. *ACM Trans. Inf. Syst.*, 22(1):54–88, 2004.

16. Peter Mika. Ontologies are us: A unified model of social networks and semantics. In *International Semantic Web Conference*, LNCS, pages 522–536. Springer, 2005.

17. Netflix homepage.

18. Netflix prize homepage.

19. Satoshi Niwa, Takuo Doi, and Shinichi Honiden. Web page recommender system based on folksonomy mining for itng '06 submissions. In *ITNG '06: Proceedings of the Third International Conference on Information Technology: New Generations (ITNG'06)*, pages 388–393, Washington, DC, USA, 2006. IEEE Computer Society.

20. Ben J. Schafer, Joseph A. Konstan, and John Riedi. Recommender systems in e-commerce. In *ACM Conference on Electronic Commerce*, pages 158–166, 1999.

21. I. Schwab, A. Kobsa, and I. Koychev. Learning user interests through positive examples using content analysis and collaborative filtering, 2001.

22. S. Staab, S. Santini, F. Nack, L. Steels, and A. Maedche. Emergent semantics. *IEEE Intelligent Systems*, 17(1):78–86, 2002.

23. Vander Wal. T. Folksonomy definition and wikipedia, November 2005.

24. The internet movie database - wikipedia entry.

25. The internet movie database (IMDB) homepage.