HT'14 Workshop / SP 2014: First International Workshop on Social Personalization
- Preface -

This year we commemorate 25 years of the invention of the World Wide Web by Tim Berners-Lee in the CERN, an invention that has shaped our lives in the last decades. Several changes have occurred since its inception and one of most significant ones is the notion of a Personalized and Adaptive Hypermedia. Another important trend that has had an enormous impact in the last decade is the Social Web. Though several conferences and workshops already focus on these topics, in this 1st International Workshop on Social Personalization we merge these trends into one event, where Social context plays a fundamental role on the fields of User Modeling, Personalization and Recommendations. This combined topic is important because it involves leveraging new sources of information that are specific for social systems such as shared items and tags, user public profiles, social connections, and logs of user social activities in order to improve people's information access in a wide variety of tasks and across different devices. These social information sources offer social personalization systems a chance to compensate for the lack of information and structure that is used by traditional personalization technologies ranging from recommender systems to E-learning. Thus, the goal of this workshop is to share and discuss research that goes hopefully beyond classic personalization techniques, trying to capitalize potentially useful information available in social data for paving the way to more efficient personalized information access technologies.

Overall, we are grateful of the participation of the research community interested in this topic. The call for papers attracted 15 submissions, from which we accepted seven as regular papers and five as posters based on a rigorous reviewing process. Additionally, the workshop features the invited talk of Luca Maria Aiello from Yahoo! Research. The accepted papers cover a variety of topics, including social media and social tagging systems, group recommendation, event-based analysis, visualization and sentiment analysis.

We thank all participants of the workshop for their contributions and ACM and the organizers of the HT 2014 conference for their support, especially Luca Maria Aiello, our invited keynote speaker. We also want to thank our reviewers for their careful help in selecting and improving the provided submissions. We hope that you will find this program interesting and thought-provoking and that the workshop will provide you with a valuable opportunity to share ideas with other researchers and practitioners from institutions around the world. We are looking forward to a very exciting and interesting workshop.

Peter Brusilovsky
University of Pittsburgh
Pittsburgh, PA, USA

Leandro Balby Marinho
UFCG
Campina Grande, Brasil

Denis Parra
PUC Chile
Santiago, Chile

Eliana Scheihing
UACh Chile
Valdivia, Chile

Christoph Trattner
Know-Center, TU-Graz
Graz, Austria

HT'14 Social Personalization 2014 Workshop Program

1. Emanuel Lacic, Dominik Kowald, Paul Seitlinger, Christoph Trattner and Denis Parra. *Recommending Items in Social Tagging Systems Using Tag and Time Information (Full Paper)*

2. Augusto Queiroz de Macedo and Leandro Balby Marinho. *Event Recommendation in Event-based Social Networks (Full Paper)*

3. Eduardo Graells-Garrido, Mounia Lalmas and Ricardo Baeza-Yates. *Sentiment Visualisation Widgets for Exploratory Search (Full Paper)*

4. Smitashree Choudhury and Harith Alani. *Personal Life Event Detection from Social Media (Twitter) (Full Paper)*

5. Sarik Ghazarian, Nafiseh Shabib and Mohammadali Nematbakhsh. *Improving Sparsity Problem in Group Recommendation (Full Paper)*

6. Paulo Cavalin, Maira Gatti and Claudio Pinhanez. *Towards Personalized Offer by Means of Life Event Detection on Social Media and Entity Matching (Full Paper)*

7. Simen Fivelstad Smaaberg, Nafiseh Shabib and John Krogstie. *A User-Study on Context-aware Group Recommendation for Concerts (Full Paper)*

8. Michal Kompan and Maria Bielikova. *Voting Based Group Recommendation: How Users Vote (Poster)*

9. Dirk Ahlers and Mahsa Mehrpoor. *Semantic Social Recommendations in Knowledge-Based Engineering (Poster)*

10. Jordan Barría, Eliana Scheihing and Denis Parra. *Visualizing Student Participation in a Collaborative Learning Environment (Poster)*

11. Petr Saloun, Adam Ondrejka and Ivan Zelinka. *Estimating Users' Areas of Research by Publications and Profiles on Social Networks (Poster)*

12. Marharyta Aleksandrova, Armelle Brun, Anne Boyer and Oleg Chertov. *What about Interpreting Features in Matrix Factorization-based Recommender Systems as Users? (Poster)*

HT'14 Social Personalization 2014 Workshop Organization

Workshop Chairs:
Peter Brusilovsky (University of Pittsburgh, USA)
Leandro Balby Marinho (Universidade Federal de Campina Grande, Brasil)
Denis Parra (Pontificia Universidad Católica de Chile, Chile)
Eliana Scheihing (Universidad Austral de Chile, Chile)
Christoph Trattner (Know-Center, TU-Graz, Austria)

Program Committee:
Nazareno Andrade, University of Campina Grande, Brazil
Jussara Almeida, University of Minas Gerais, Brazil
Martin Atzmueller, University of Kassel, Germany
Alejandro Bellogin, Universidad Autónoma de Madrid, Spain
Shlomo Berkovsky, NICTA, AU
Anmol Bhasin, LinkedIn, USA
Danny Bickson, GraphLab, USA
Steven Bourke, UCD, Ireland
Robin Burke, de Paul, USA
Ed Chi, Google, USA
Alvin Chin, Microsoft, China
Vania Dimitrova, University of Leeds, UK
Lucas Drumond, University Hildesheim, Germany
Alexander Felfernig, TU-Graz, Austria
Zeno Gantner, Nokia, Germany
Ruth Garcia, Yahoo! Research Barcelona, Spain
Ido Guy, IBM Research, Israel
Eelco Herder, L3S, Germany
Andreas Hotho, University of Würzburg, Germany
Geert-Jan Houben, TU-Delft, Netherlands
Sharon Hsiao, Columbia University, USA
Bart Knijnenburg, University of California Irvine, USA
Milos Kravcik, RWTH Aachen, Germany
Mounia Lalmas, Yahoo! Research Barcelona, Spain
Neal Lathia, Cambridge University, UK
Elisabeth Lex, Graz University of Technology, Austria
Tobias Ley, Tallinn University, Estonia
Alan Said, TU-Delft, NL
Shaghayegh Sahebi, University of Pittsburgh, USA
Eduardo Veas, Know-Center, Austria
Katrien Verbert, Vrije Universiteit Brussel, Belgium
Tao Ye, Pandora, USA
Arkaitz Zubiaga, New York City University, USA

Recommending Items in Social Tagging Systems Using Tag and Time Information

Emanuel Lacic*
Knowledge Technology Institute
Graz University of Technology
Graz, Austria
elacic@know-center.at

Dominik Kowald*
Know-Center
Graz University of Technology
Graz, Austria
dkowald@know-center.at

Paul Seitlinger
Knowledge Technology Institute
Graz University of Technology
Graz, Austria
paul.seitlinger@tugraz.at

Christoph Trattner
Know-Center
Graz University of Technology
Graz, Austria
ctrattner@know-center.at

Denis Parra
CS Department
Pontificia Universidad Católica de Chile
Santiago, Chile
dparra@ing.puc.cl

ABSTRACT

In this work we present a novel item recommendation approach that aims at improving Collaborative Filtering (CF) in social tagging systems using the information about tags and time. Our algorithm follows a two-step approach, where in the first step a potentially interesting candidate item-set is found using user-based CF and in the second step this candidate item-set is ranked using item-based CF. Within this ranking step we integrate the information of tag usage and time using the Base-Level Learning (BLL) equation coming from human memory theory that is used to determine the reuse-probability of words and tags using a power-law forgetting function.

As the results of our extensive evaluation conducted on datasets gathered from three social tagging systems (BibSonomy, CiteULike and MovieLens) show, the usage of tag-based and time information via the BLL equation also helps to improve the ranking and recommendation process of items and thus, can be used to realize an effective item recommender that outperforms two alternative algorithms which also exploit time and tag-based information.

*Both authors contributed equally to this work.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data mining*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering*

Keywords

recommender systems; social tagging; collaborative filtering; item ranking; base-level learning equation

1. INTRODUCTION

Over the past few years social tagging gained tremendously in popularity, helping people for instance to categorize or describe resources on the Web for better information retrieval (e.g., BibSonomy or CiteULike) [13, 23]. Although the process of tagging has been well explored in the past and in particular the task of predicting the right tags to the user in a personalized manner [12, 20], studies on predictive models to recommend items to users based on social tags are still rare. To contribute to this sparse field of research, in this paper we present preliminary results of a study that aims at addressing this issue. In particular, we provide first results of a novel attempt to improve item recommendations by taking into account peoples' social tags and the information of the time the tags have been applied by the users. As shown in related work, recommending items to users in a collaborative manner relying on social tagging information is not an easy task in general (e.g., [24] or [17]). However, other related work has also proved that the information of time is an important factor to make the models more accurate in the end (e.g., [26] or [10]).

Contrary to the previous work mentioned above, we suggest a less data-driven approach that is inspired by principles of human memory theory about remembering things over time. As shown in our previous work on tag recommender systems [15], the base-level learning (BLL) equation introduced by Anderson and Schooler [16] (see also Anderson et al. [1]), which integrates tag frequency and recency (i.e., the time since the last tag usage), can be used to implement an effective tag recommendation and ranking algorithm. In partic-

ular, the BLL equation models the time-depended drift of forgetting of words and tags using a power-law distribution in order to determine a probability value that a specific tag will be reused by a target user.

In this work, we apply this equation for ranking and recommending items to users. To this end, we present a novel recommender approach called *Collaborative Item Ranking Using Tag and Time Information (CIRTT)* that firstly identifies a potentially interesting candidate item set and secondly, ranks this candidate set in a personalized manner (similar to [10]). In this second step of personalization, we integrate the BLL equation to include this information about tags and time. To investigate the question as to whether tag and time information can improve the ranking and recommendation process, we conducted an extensive evaluation using folksonomy datasets gathered from three social tagging systems (BibSonomy, CiteULike and MovieLens). Within this study we compared our approach to two alternative tag and time based recommender algorithms [26, 10] amongst others. The results show that integrating tag and time information using the BLL equation helps to improve item recommendations and to outperform state-of-the-art baselines in terms of recommender accuracy.

The remainder of this paper is organized as follows. We begin with explaining our tag and time based approach CIRTT in Section 2. Then we describe the experimental setup of our evaluation in Section 3 and summarize the results of this study in Section 4. Finally, in Section 5, we close the paper with a short conclusion and an outlook into the future.

2. APPROACH

In this section we provide a detailed description of our item recommendation approach called *Collaborative Item Ranking Using Tag and Time Information (CIRTT)*. In general, our CIRTT algorithm uses a similar strategy as the approach proposed by Huang et al. [10] and thus, consists of two steps relying on a combination of user- and item-based CF: in the first step, a potentially interesting candidate item set for the target user u is determined and in the second step, this candidate item set gets ranked using item similarities and tag and time information.

Step one (i.e., determining candidate items) is conducted using a simple user-based CF approach. Hence, we first find the most similar users for the target user u (i.e., the neighborhood) based on the binary user-item matrix $B_{u,i}$ (see also [26]) and then, use the bookmarked items of these neighbours as our candidate item set. We use a neighbourhood of $k = 20$ users and the Cosine similarity measure [7] (see also Section 3.3).

In the second step (i.e., ranking candidate items) we use an item-based CF approach in order to determine the relevance of each candidate item for the target user based on the items she has bookmarked in the past. Hence, for each candidate item i in the candidate item set we calculate this combined similarity value $sim(u, i)$ by the item-based CF formula:

$$sim(u,i) = \sum_{j \in items(u)} sim(i,j) \tag{1}$$

Dataset	$\|B\|$	$\|U\|$	$\|R\|$	$\|T\|$	$\|TAS\|$
BibSonomy	82,539	2,437	28,000	30,919	339,337
CiteULike	36,471	3,202	15,400	20,937	99,635
MovieLens	53,607	3,983	5,724	14,883	92,387

Table 1: Properties of the datasets, where $|B|$ is the number of bookmarks, $|U|$ the number of users, $|R|$ the number of resources, $|T|$ the number of tags and $|TAS|$ the number of tag assignments.

, where $items(u)$ is the set of items the target user u has bookmarked in the past. This item-based CF step helps us to give a higher ranking to candidate items that are more similar to the items the target user has bookmarked in the past (see also [10]).

To finally realize CIRTT in order to integrate tag and time information we make use of the base-level learning (BLL) equation proposed by Anderson et al. [1]. As described in our previous work [15], the BLL equation can be used to determine a relevance value for a tag t in the tag assignments of a target user u based on tag frequency and recency:

$$BLL(u,t) = ln(\sum_{i=1}^{n} t_i^{-d}) \tag{2}$$

, where n is the number of times t has been used by u and t_i is the recency, i.e., the time since the i^{th} occurrence of t in the tag assignments of u. The exponent d is used to model the power law of forgetting memory items and is usually set to .5 (see [1]). In order to map these BLL values on a range of 0 - 1, we used the same normalization method as used in our previous work [15].

We adopt this equation for the ranking of items in social tagging systems using a similar method as proposed in [26] and [10]. Thus, a user is assumed to prefer an item if it has been tagged with tags of high relevance for the user, that is, with tags exhibiting a high BLL value. Given this assumption, the BLL value of a given item i for the target user u is determined using the following formula:

$$BLL(u,i) = \sum_{t \cup tags(u,i)} BLL(u,t) \tag{3}$$

, where $tags(u, i)$ is the set of tags u has used to tag i.

Taken together, the prediction value $pred(u, i)$ of a candidate item i using our CIRTT approach is given by:

$$pred(u,i) = \underbrace{\sum_{j \in items(u)} sim(i,j)}_{sim(u,i)} \times BLL(u,i) \tag{4}$$

This approach enables us to weight higher the items within the candidate set that are more important to the target user (i.e., items associated with tags exhibiting a high BLL value that integrates tag frequency and recency). CIRTT and the baseline algorithms presented in this work are implemented in the Java programming language, are open-source software and can be downloaded online from our Github Repository[1] [14].

[1]https://github.com/learning-layers/TagRec/

3. EXPERIMENTAL SETUP

In this section we describe in detail the datasets, the evaluation methodology and metrics as well as the baseline algorithms used for our experiments.

3.1 Datasets

In order to evaluate our approach and for reasons of reproducibility we used freely-available folksonomies gathered from three well-known social-tagging systems. We used datasets of the social bookmark and publication sharing system BibSonomy[2], the reference management system CiteULike[3] and the movie recommendation site MovieLens[4]. As suggested by related work in the field (e.g. [11, 9]), we excluded all automatically imported and generated tags (e.g., bibtexImport). In the case of CiteULike we randomly selected 10% of the user profiles for reasons of computational effort (see also [7]).

We did not use a full p-core pruning technique, since this would negatively influence the recommender evaluation results in social tagging system as shown by Doerfel and Jäschke [6], but excluded all unique resources (i.e., resources that have been bookmarked only by a single user). The final dataset statistics can be found in Table 1.

3.2 Evaluation Methodology

To evaluate our item recommender approach we used a training and test-set split method as proposed by popular and related work in this area [10, 26]. Hence, for each user we sorted her bookmarks in chronological order and used the 20% most recent bookmarks for testing and the rest for training. With the training set we examined then whether a recommender approach could predict the bookmarked resources of a target user in the test set. This procedure also simulates well a real environment where the bookmarking behavior of a user in the future is tried to be predicted based on the bookmarking behavior in the past [3].

To finally quantify the recommendation accuracy of our approaches, we used a set of well-known information retrieval metrics. In particular, we report Normalized Discounted Cumulative Gain (nDCG@20), Mean Average Precision (MAP @20), Recall (R@20), Diversity (D) and User Coverage (UC) [21, 8]. All performance metrics are calculated and reported based on the top-20 recommended items. Moreover we also show the performance of the algorithms in the plots of all three accuracy metrics (nDCG, MAP and Recall) for 1 - 20 recommended items (see also [4]).

3.3 Baseline Algorithms

In order to evaluate our tag and time based approach, we compared CIRTT to several baseline methods in terms of recommender accuracy. The algorithms have been selected with respect to their popularity, performance and novelty.

MostPopular (MP): The most basic approach we utilized is the simple *Most Popular (MP)* approach that recommends for any user the same set of items. These items are weighted by their frequency in all bookmarks, meaning that the most frequently occurring items in the dataset are recommended.

User-based Collaborative Filtering (CF): Another approach we benchmarked against is the well-known *User-based Collaborative Filtering (CF)* recommendation algorithm [19]. The main idea of CF is that users that are more similar to each other (i.e., have similar taste), will probably also like the same items. Thus, the CF approach first finds the k most similar users for the target user and afterwards recommends their items that are new to her (i.e., have not been bookmarked before). We calculated the user-similarities based on both, the binary user-item matrix as proposed in [26] (hereinafter referred to as CF_B) and the tag-based user profiles as proposed in [10] (hereinafter referred to as CF_T). Although we also considered using *Item-based CF* [18], we dismissed this method based on the tag-based recommender experiments of Bogers et al. [2] showing that user-based CF always beats item-based CF. They explain the result given that the number of items in their dataset is larger than the number of users, and this is also the case in our three datasets (Table 1).

Collaborative Filtering Using Tag and Time Information (Z / H): We also compared our approach to two alternative algorithms that focus on improving Collaborative Filtering for social tagging systems using tag and time information. The first one has been proposed by Zheng et al. [26] (hereinafter referred to as Z) and improves the traditional CF approach based on the binary user-resource matrix using tag and time information. As in our CIRTT approach this is done using information about tag frequency and recency but in contrast to our solution the authors model the forgetting process using an exponential distribution rather than a power-law distribution. Moreover, this information is already used in the user similarity calculation step and not in the item ranking step as it is done in our approach.

The second tag and time-based mechanism we tried to benchmark against was proposed by Huang et al. [10] (hereinafter referred to as H). As in our approach, this algorithm uses a 2-step recommendation process, where in the first step a potentially interesting candidate item-set for the target user is determined using user-based CF and in the second step this candidate item-set is ranked using item-based CF. In contrast to CIRTT, the authors calculate the user and item similarities based on user tag-profiles rather than based on the binary user-item matrix. Furthermore, in this algorithm the forgetting process is modeled using a simple linear function rather than a power-law distribution.

All CF-based approaches mentioned in this section use a neighborhood of 20 users and make use of the Cosine similarity measure as it is also done in CIRTT (see also [7]).

4. RESULTS

In this section, we present the results of the evaluation comparing our CIRTT approach to the baseline algorithms described in Section 3.3 with respect to recommender accuracy on three different folksonomy datasets (BibSonomy, CiteULike and MovieLens).

[2]http://www.kde.cs.uni-kassel.de/bibsonomy/dumps
[3]http://www.citeulike.org/faq/data.adp
[4]http://grouplens.org/datasets/movielens/

Dataset	Metric	MP	CF_T	CF_B	Z	H	$CIRTT$
BibSonomy	nDCG@20	.0143	.0448	.0610	.0621	.0564	**.0638**
	MAP@20	.0057	.0319	.0440	.0447	.0394	**.0464**
	R@20	.0204	.0618	.0820	.0834	.0816	**.0907**
	D	.8307	.8275	.8852	.8528	.6209	.8811
	UC	100%	99.76%	99.52%	99.52%	99.76%	99.76%
CiteULike	nDCG@20	.0062	.0407	.0717	.0762	.0706	**.0912**
	MAP@20	.0036	.0241	.0453	.0484	.0459	**.0629**
	R@20	.0077	.0630	.1033	.1077	.0928	**.1225**
	D	.8936	.7969	.8642	.8145	.6318	.8640
	UC	100%	98.38%	96.44%	97.32%	98.38%	97.61%
MovieLens	nDCG@20	.0198	.0361	.0602	.0614	.0484	**.0650**
	MAP@20	.0075	.0201	.0347	.0367	.0263	**.0413**
	R@20	.0366	.0561	.1031	.1013	.0763	**.1058**
	D	.9326	.8861	.9267	.9119	.7789	.9176
	UC	100%	97.82%	95.90%	98.43%	97.82%	95.90%

Table 2: nDCG@20, MAP@20, R@20, D and UC values for BibSonomy, CiteULike and MovieLens showing that CIRTT, that integrates tag and time information using the BLL-equation, outperforms state-of-the-art baseline algorithms (highest accuracy values are highlighted in bold).

In an extensive empirical study, Cremonesi et al. [5] have shown that standard Information Retrieval accuracy metrics (e.g., Recall or nDCG) are well suited to evaluate recommender systems, at least in case of top-N recommendation tasks. Therefore, Table 2 provides measures of accuracy (nDCG@20, MAP@20, R@20) and - additionally - measures of Diversity (D) and User Coverage (UC) for each approach and for each of the three datasets.

As expected, the MP baseline, which is not personalized at all, resulted in the lowest accuracy estimates. Regarding the two traditional CF algorithms, CF_B, which constructs a binary user-item matrix based on bookmarks, performs better than CF_T, which is based solely on the user tag-profiles. Regarding the two alternative tag- and time-based approaches, a same phenomenon can be observed as the algorithm of Zheng et al. (Z) [26], that is also based on the binary user-item matrix, performs better than the method of Huang et al. (H) [10], that is based on the user tag-profiles.

With respect to all accuracy metrics (nDCG@20, MAP@20, R@20), our CIRTT approach, that integrates tag and time information using the BLL-equation, performs best in all three datasets (BibSonomy, CiteULike and MovieLens). This may suggest that applying a power-law function as it is done via the BLL-equation is more appropriate to account for effects of recency than an exponential function (Zheng et al. [26]) or a linear function (Huang et al. [10]). A same pattern of results can be observed when looking at Figure 1 that reveals estimates of the nDCG, MAP and Recall measures for different sizes of the recommended item set. These plots show that only in the case of BibSonomy the approach of Zheng et al. reaches slightly higher accuracy estimates than our method for the first 7 recommended items. However, this changes when increasing the number of recommended items where our approach again produces the best recommender quality. Furthermore, we have also tried to integrate an exponential recency function [26] in our approach which resulted in lower accuracy estimates than the BLL power-law forgetting function.

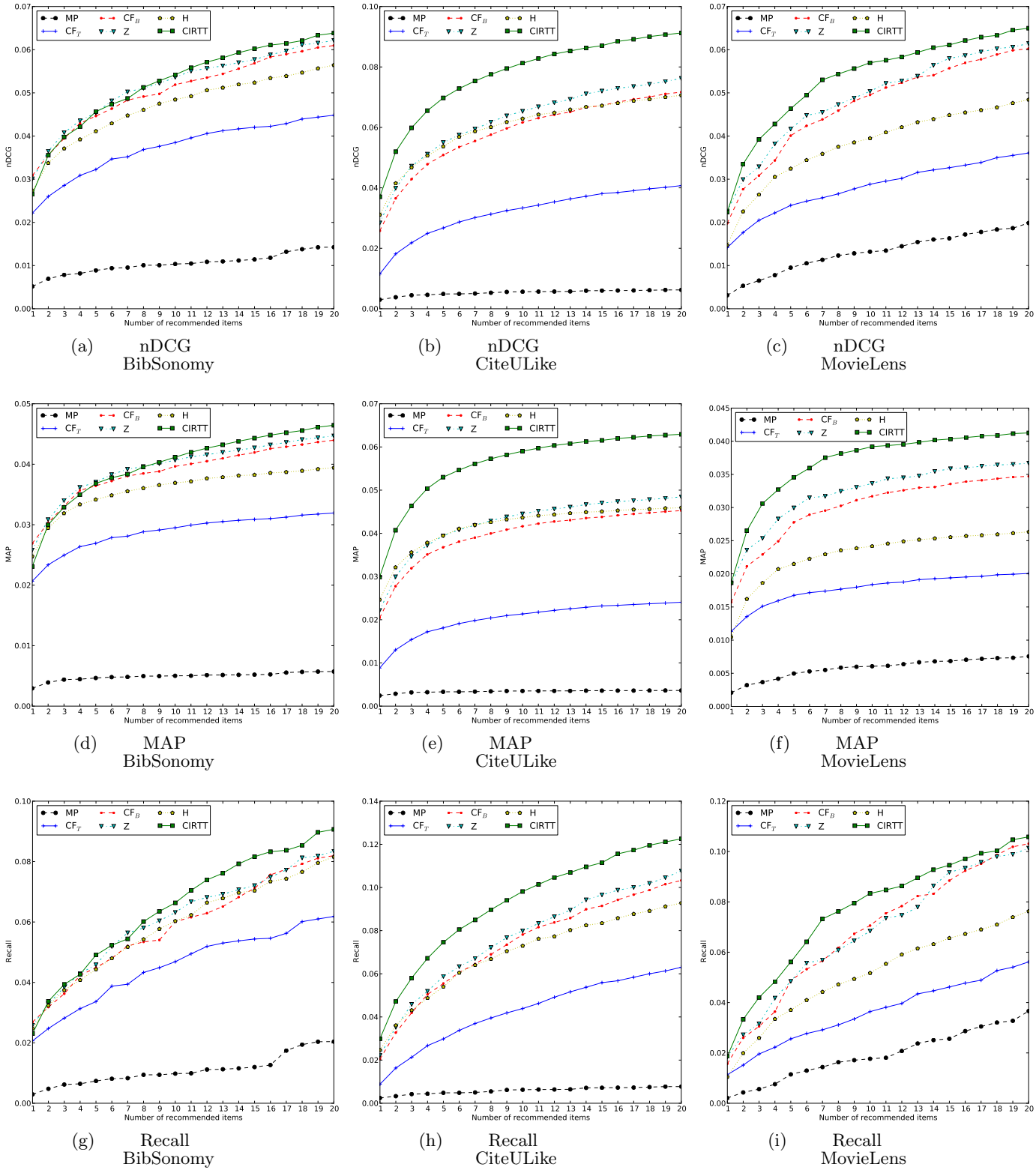
When looking at the other two not accuracy-based metrics, interestingly, the approach of Huang et al. (H) always results in the lowest Diversity (D) of recommended items. This result might appear because this approach is based on the user tag-profiles and the Diversity metric is calculated based on tags. Finally, as all personalized approaches utilize a user-based CF approach for finding similar users, the measure of User Coverage (UC) does not appear to deviate between the different algorithms. We observed the maximum deviation of 2.53% within the MovieLens dataset.

5. CONCLUSIONS & FUTURE WORK

In this work we have presented preliminary results of a novel recommendation approach called *Collaborative Item Ranking Using Tag and Time Information (CIRTT)* that aims at improving Collaborative Filtering in social tagging systems. Our algorithm follows a two-step approach as also done in [10], where in the first step a potentially interesting candidate item set is found performing user-based CF and in the second step this candidate item set is ranked performing item-based CF. Within this ranking step we integrate the information of frequency and recency of tag use applying the Base-Level Learning (BLL) equation [1]. Thus, in contrast to existing approaches that also consider information about tags and time (e.g., [26, 10]), CIRTT draws on an empirically well established formalism modeling the reuse probability of memory items (tags) in form of a power-law forgetting function. In recent work, the same formalism has turned out to substantially improve the ranking and recommendation of tags [15].

The current evaluation conducted on datasets gathered from three social tagging systems (BibSonomy, CiteULike and MovieLens) reveals that applying the BLL equation also helps to improve the ranking and recommendation process of items. Most important, the results speak in favor of an integrative research endeavor that places a data-driven approach on a theoretical foundation provided by research on human cognition and semiotics.

Our future work will aim at improving the approach presented in this paper. For example, we will examine as to



(a) nDCG BibSonomy (b) nDCG CiteULike (c) nDCG MovieLens

(d) MAP BibSonomy (e) MAP CiteULike (f) MAP MovieLens

(g) Recall BibSonomy (h) Recall CiteULike (i) Recall MovieLens

Figure 1: nDCG, MAP and Recall plots for BibSonomy, CiteULike and MovieLens showing the recommendation accuracy of our tag and time based CIRTT approach along with state-of-the-art baseline algorithms for 1 - 20 recommended items (k). We can see that CIRTT reaches the highest levels of recommender accuracy over all three metrics and on all datasets.

whether the BLL equation can also help to improve the calculation of user similarities and thus, to find more suitable user neighborhoods and candidate items. Additionally, we will put more emphasis on dynamics that have been found to play out in tagging systems (e.g., [22]) and how individual learning and forgetting processes are influenced by other individuals' behavior in the system. Moreover, we also plan to further improve the item ranking process using insights of relevant research dealing with recommender novelty and diversity (e.g., [25]) in order to increase the user acceptance. Finally, it would also be interesting to evaluate our proposed approach against state-of-the-art matrix factorization item recommender methods (e.g., SLIM or CLiMF).

Acknowledgments: This work is supported by the Know-Center, the EU funded project Learning Layers (Grant Nr. 318209) and the Austrian Science Fund (FWF): P 25593-G22. The Know-Center is funded within the Austrian COMET Program - Competence Centers for Excellent Technologies - under the auspices of the Austrian Ministry of Transport, Innovation and Technology, the Austrian Ministry of Economics and Labor and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency (FFG).

6. REFERENCES

[1] J. R. Anderson, M. D. Byrne, S. Douglass, C. Lebiere, and Y. Qin. An integrated theory of the mind. *Psychological Review*, 111(4):1036–1050, 2004.

[2] T. Bogers and A. van den Bosch. Recommending scientific articles using citeulike. In *Proc.*, RecSys '08, pages 287–290, New York, NY, USA, 2008. ACM.

[3] P. G. Campos, F. Díez, and I. Cantador. Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. *User Modeling and User-Adapted Interaction*, pages 1–53, 2013.

[4] P. Cremonesi, P. Garza, E. Quintarelli, and R. Turrin. Top-n recommendations on unpopular items with contextual knowledge. In *Workshop on Context-aware Recommender Systems '11*.

[5] P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proc.*, RecSys '10, New York, NY, USA. ACM.

[6] S. Doerfel and R. Jäschke. An analysis of tag-recommender evaluation procedures. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 343–346. ACM, 2013.

[7] J. Gemmell, T. Schimoler, M. Ramezani, L. Christiansen, and B. Mobasher. Improving folkrank with item-based collaborative filtering. *Recommender Systems & the Social Web*, 2009.

[8] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004.

[9] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. In *The semantic web: research and applications*. Springer, 2006.

[10] C.-L. Huang, P.-H. Yeh, C.-W. Lin, and D.-C. Wu. Utilizing user tag-based interests in recommender systems for social resource sharing websites. *Knowledge-Based Systems*, 2014.

[11] R. Jäschke, L. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme. Tag recommendations in folksonomies. In *Knowledge Discovery in Databases: PKDD 2007*, pages 506–514. Springer, 2007.

[12] R. Jäschke, L. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme. Tag recommendations in social bookmarking systems. *Ai Communications*, 21(4):231–247, 2008.

[13] C. Körner, D. Benz, A. Hotho, M. Strohmaier, and G. Stumme. Stop thinking, start tagging: tag semantics emerge from collaborative verbosity. In *Proceedings of the 19th international conference on World wide web*, pages 521–530. ACM, 2010.

[14] D. Kowald, E. Lacic, and C. Trattner. Tagrec: Towards a standardized tag recommender benchmarking framework. In *Proc.*, HT '14, New York, NY, USA, 2014. ACM.

[15] D. Kowald, P. Seitlinger, C. Trattner, and T. Ley. Long time no see: The probability of reusing tags as a function of frequency and recency. In *Proc. WWW '14*. ACM.

[16] J. R. A. Lael J. Schooler. Reflections of the environment in memory. *Psychological Science*, 1991.

[17] D. Parra-Santander and P. Brusilovsky. Improving collaborative filtering in social tagging systems for the recommendation of scientific articles. In *WI-IAT, 2010 IEEE/WIC/ACM*.

[18] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proc.*, WWW '01, pages 285–295, New York, NY, USA, 2001. ACM.

[19] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer, 2007.

[20] P. Seitlinger, D. Kowald, C. Trattner, and T. Ley. Recommending tags with a model of human categorization. In *Proc.*, CIKM '13, pages 2381–2386, New York, NY, USA, 2013. ACM.

[21] B. Smyth and P. McClave. Similarity vs. diversity. In D. Aha and I. Watson, editors, *Case-Based Reasoning Research and Development*, LNCS. Springer, 2001.

[22] L. Steels. Semiotic dynamics for embodied agents. *Intelligent Systems, IEEE*, 21(3):32–38, 2006.

[23] C. Trattner, Y.-l. Lin, D. Parra, Z. Yue, W. Real, and P. Brusilovsky. Evaluating tag-based information access in image collections. In *Proc.*, HT '12, pages 113–122, New York, NY, USA, 2012. ACM.

[24] K. H. Tso-Sutter, L. B. Marinho, and L. Schmidt-Thieme. Tag-aware recommender systems by fusion of collaborative filtering algorithms. In *Proc. of SAC '08*. ACM.

[25] S. Vargas and P. Castells. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proc.*, RecSys '11. ACM.

[26] N. Zheng and Q. Li. A recommender system based on tag and time information for social tagging systems. *Expert Syst. Appl.*, 2011.

Event Recommendation in Event-based Social Networks

Augusto Q. de Macedo
Federal University of Campina Grande
Aprigio Veloso 882
Campina Grande, Brazil
augusto@copin.ufcg.edu.br

Leandro B. Marinho
Federal University of Campina Grande
Aprigio Veloso 882
Campina Grande, Brazil
lbmarinho@computacao.ufcg.edu.br

ABSTRACT

With the large number of events published all the time in event-based social networks (EBSN), it has become increasingly difficult for users to find the events that best match their preferences. Recommender systems appear as a natural solution to this problem. However, the event recommendation scenario is quite different from typical recommendation domains (e.g. movies), since there is an intrinsic new item problem involved (i.e. events can not be "consumed" before their occurrence) and scarce collaborative information. Although some few works have appeared in this area, there is still lacking in the literature an extensive analysis of the different characteristics of EBSN data that can affect the design of event recommenders. In this paper we provide a contribution in this direction, where we investigate and discuss important features of EBSN such as sparsity, events life time, co-participation of users in events and geographic features. We also shed some light on the performance and limitations of several well known recommendation algorithms and combinations of them on real data collected from meetup.com.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information filtering; H.2.8 [**Database Applications**]: Data mining

General Terms

Algorithms, Experimentation

Keywords

Recommender systems, Statistical Analysis, Social network, Cold-Start

1. INTRODUCTION

In the last few years the Event-Based Social Networks (EBSN), such as Meetup[1] and Plancast[2], have gained momentum due to their ability to connect people around the events they attended or are likely to attend in the future. In EBSN people can create events of any kind, for example, musical concerts and political manifestations, and share it with other users. With the large number of events available all the time, especially in large and touristic cities, it has become increasingly difficult for the users to find the events that best match his/her preferences. Recommender systems appear as a natural solution for this problem.

The event recommendation problem, however, is quite different from the classic recommendation scenarios (e.g., movie recommendation), where the items to be recommended have already been consumed/rated by other users. In EBSN, the events to be recommended can not be "consumed" or rated before its occurrence, so, in principle, there is a lack of collaborative data available for traditional collaborative filtering-based algorithms to operate upon, which raises the issue known as the new item cold-start problem. One way to alleviate this problem is to use the intention of users on going or not to events, through their RSVPs[3], as explicit feedback data. But as we will show along the paper, even this kind of data is very sparse.

Although some few works have appeared recently in this area, there is still a gap in the literature concerning an extensive analysis of the different characteristics of EBSN data that can affect the design of effective event recommenders. In this paper we try to fill in this gap by addressing the following questions:

- How sparse is the RSVP data and how it affects collaborative-filtering algorithms?
- In which point of the event life time users tend to provide RSVPs?
- How the geographic distance between the users home and active events affect their decision on attending these events?
- Are past RSVPs usefull for predicting future RSVPs?

We derive important insights from this investigation that we believe will pave the way to the design of more efficient and informed recommendation algorithms. Moreover,

[1] www.meetup.com
[2] www.plancast.com
[3] RSVP stands for the French expression "répondez s'il vous plaît", meaning "please respond"

we compare several well known recommendation algorithms and discuss their performances and limitations on real data collected from the Meetup platform, a popular EBSN that offers large portions of event data through their API.

The rest of this paper is organized as follows. In Section 2 we discuss related works. In Section 3 we present the data collection and analysis. In Section 4 compare several well known algorithms from the literature and discuss their performances and limitations. Section 5 concludes the work.

2. RELATED WORKS

In this section we summarize the most relevant related work on event recommendation.

Minkov et al. [4] approach the event recommendation problem through a ranking-based matrix factorization algorithm. For composing the training data, explicit feedback was required through a form where users had to indicate which events, in this case scientific seminars, they were likely to attend. The results of this paper show that this approach is superior to content-based filtering. Although they have conducted experiments with real users, it consisted of a small scale experiment where only 90 users over 15 weeks were considered. Moreover, it was required explicit feedback from the users. Our work focus on an offline large scale analysis and experimentation on data collected from a popular EBSN.

A seminal and closely related work to ours is the one introduced in [3] where the authors analyze real data collected from Meetup all over USA and investigate EBSN properties, such as heavy-tailed degree distributions, strong geographic dependence of social interactions, and the interplay between online and offline interactions of users. They also propose a recommendation model of users in EBSN.

In [5] it is proposed a content-based recommender where cultural events metadata are enriched with open linked data available on the web. While this approach might work well for small scope event domains, it may find problems to cover the multitude of event types of EBSN. Another work from Pessemier et al. [1] presented a smartphone application, Outlife, to recommend events for users and users to invite for an event based on the users Facebook profiles. The event recommendation is addressed by selecting the most appropriate algorithm for each situation (with a decision tree) out of a set of recommender algorithms. If no ratings are available a content-based algorithm is used.

A recent work by Khrouf et al. [2] propose a hybrid event recommender that combines linked open data, social information and content features. While the authors focus their experiments on a small set of Last.fm users and events and a small set of event types (i.e. mostly concerts and festivals), we investigate large scale data on a multitude of event types. Furthermore, wile the authors of this work focus on the denser portions of the data, we investigate the performance of several recommenders under the true level of sparsity found on EBSN.

Thus, our work is complementary to the aforementioned works, where we investigate previously unexplored features of EBSN and how they can affect the performance of event recommendation algorithms.

3. DATA ANALYSIS

Meetup is one the world's largest EBSN nowadays[4]. It provides an on-line environment where people can meet both on-line and face-to-face. Events of all kinds are published all the time, ranging from simple get togethers to large concerts and conferences. Moreover, large portions of data are offered through the site on-line API[5], which turns Meetup into a good test bed for investigating new event recommendation approaches.

3.1 Data Collection

The cities chosen for our experiments were Phoenix, Chicago and San Jose, all from USA. These cities were selected because they (i) are among the top cities in number of users and events in Meetup and (ii) are located in different states, which represent eventual cultural differences and thus contribute to form a rich and diverse sample to work with.

Meetup is organized in on-line groups, where every group has a physical location. To collect the data, we passed the city names as seeds and retrieved all the groups located in a radius of 100 miles from a city location returned by Meetup. Then every user, event and RSVP (i.e. user-event pairs) of those groups were retrieved. The data collected comprise the period from January, 2010 to December, 2011. Table 1 presents the characteristics of the data collected. It is worth noting the extreme sparsity of RSVPs in all cities considered.

Table 1: Data Statistics

City	Users	Events	RSVPs	Sparsity
Phoenix	589,808	215,338	1,557,161	99.998%
Chicago	719,011	220,076	1,353,795	99.999%
San Jose	281,547	242,216	1,717,792	99.997%

In the following we investigate some characteristics of this data with respect to RSVPs, event life time, co-participation of users in events and the distances between the users home and event locations.

3.2 RSVP Analysis

When an event is created, users can provide RSVPs to it, i.e., provide (Yes or No) responses. We consider that a user who respond with "Yes" has a higher probability to attend the event than the user who answer "No" or provide no answer. Hence, we use this response as a proxy value to the event attendance rate, as the real attendance count is not available in Meetup.

Figure 1 shows the distribution of positive RSVPs per event for all cities. The numbers show that more than 45% of the events have at most 1 RSVP. Approximately 90% of the events have at most 10 RSVPs in all cities. The logarithm scale in the x-axis emphasizes the high skewness of the distribution leading one to conclude that the large majority of events, in all cities investigated, have low attendance.

[4] http://www.meetup.com/about/

[5] www.meetup.com/meetup_api/

This represents a major problem for most of the collaborative filtering-based recommendation algorithms which are well known to deteriorate under severe levels of sparsity.

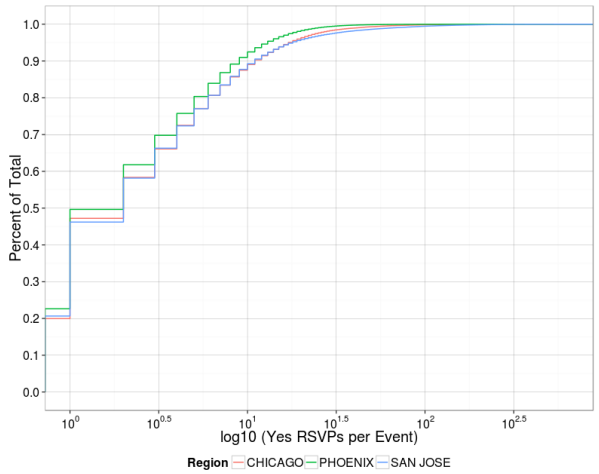


Figure 1: Cumulative Distribution of RSVPs per Event

3.3 Event Life Time

We consider the life time of an event as the period between its creation in Meetup and its occurrence. In Figure 2 we can see that most of the events have a life time ranging from 5 to 100 days. This means that while a small percentage of events have a very short life time (1 day), most of the events are active long enough to be discovered by the users or brought to their attention.

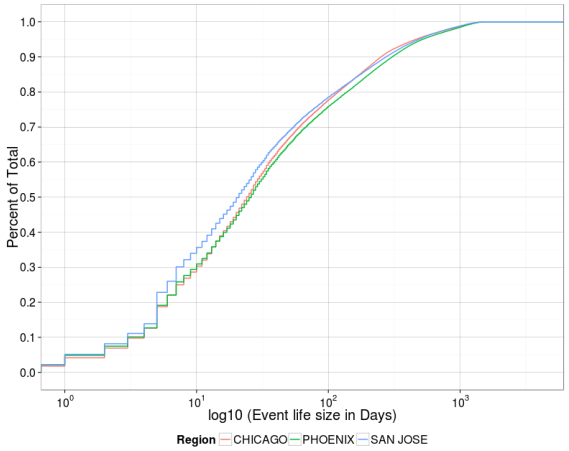


Figure 2: Cumulative Distribution of Event Life

3.4 When do RSVPs occur?

Here we investigate when exactly the positive RSVPs occur during the life time of events. Figure 3 shows in the x-axis the 21 first positive RSVPs[6], in chronological order, regarding all the events of the three cities considered. The y-axis ranges from 0, when the event is created, to 1, when it happens. Notice that the more positive RSVPs events receive, the closer to the events occurrence the RSVPs are given. Although the cities investigated present small variations in this respect, they follow the same overall pattern, i.e., most of the RSVPs are provided close to the occurrence of the event. This is even more visible in the events with a life time greater than 100 days, for example, which we noticed to receive more than 80% of all positive RSVPs (among the 21 considered) in the last 20% of the their life times.

[6]Note that approximately 95% of all events have 21 or less RSVPs

This observation bears several implications to the design of effective event recommenders. For example, after the creation of the event there will be scarce collaborative (in terms of RSVPs) information to be used, leaving room to content-based approaches. As the occurrence of the event approaches, more RSVPs are provided which favours collaborative filtering-based methods and hybrid approaches.

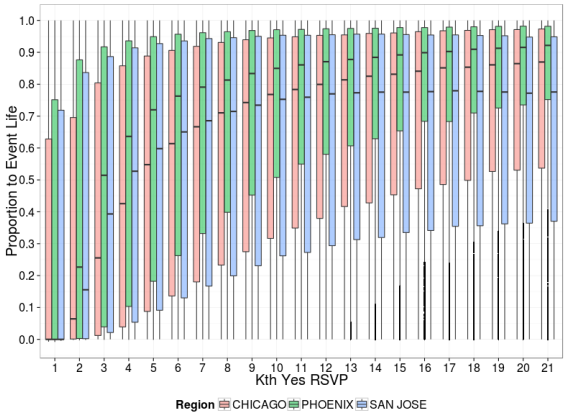


Figure 3: Cumulative Distribution of the time to the k-th "Yes" RSVP relative to the event life time

3.5 Collaborative Analysis

The collaborative aspect of the data was investigated by the distribution of events co-participation by two different users (in terms of positive RSVPs). Our analysis suggests that approximately only 30% of the users co-participated in two or more events in all cities considered. This observation represents an empirical bound to the effectiveness of collaborative filtering-based recommenders.

3.6 Distance Analysis

Figure 4 depicts the distance distribution between the users home and events locations, also investigated by other works [2, 6]. We can see that around 50% for the users provided positive RSVPs to events within 10 Km from their homes, while users do not provide RSVPs to events farther then 100 Km to their homes. A recommendation algorithm could use this observation to weigh events nearby the users home higher than farther events.

4. EVALUATION

In this section we compare some well known top-n recommendation algorithms for the event recommendation task. We also evaluate the algorithms in different levels of sparsity in order to investigate their limitations.

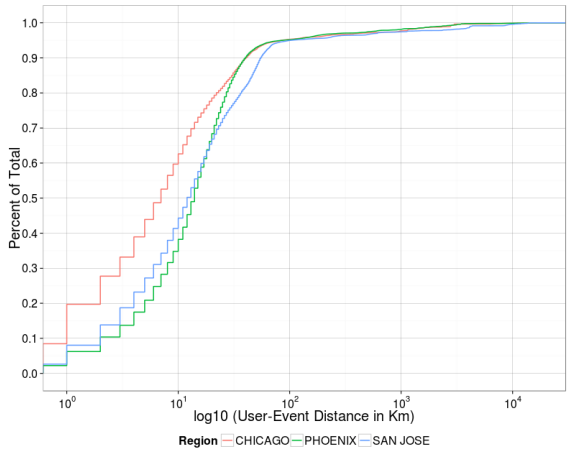


Figure 4: Cumulative Distribution of the Distance between the User and Event Location

4.1 Data Preparation

The data sets of each city were time split in order to resemble a real world setting. We selected 6 time stamps, equally spaced in time, for splitting training and test. For each partition time stamp, we used the previous 6 months for training and the events created during these 6 months but occurring after the partition time stamp for test. The average number of users, events and user-events pairs (RSVPs) after these partitions are displayed in Table 2 .

Table 2: Average number of susers, events and user-events pairs after paritions

City	# Users	# Events	# User-Events
Phoenix	2,176.8	4,483.3	9,870
Chicago	2,814.3	2,955.7	8,703.7
San Jose	3694.7	3,052.2	11,025.5

4.2 Sparsity Analysis

Here we investigate the sparsity of the recommendable events, in all partitions, in the following levels:

$$\{0, 1, 2, 3, 4, 5, 6-10, 11-20, >20\}$$

where each level denotes the number of positive RSVPs received per event. Figure 5 shows the event sparsity level plot. The y-axis counts the number of events in the test set that has the given sparsity level in the train. This plot tell us that regardless of when we partition the data set, there will be always a large number of events with no RSVPs. Therefore, cold-start appear as an inherent problem of the event recommendation domain.

4.3 Evaluation Metric

In this paper we are considering top-n item recommendations, which are usually related to the generation of a personalized ranking recommendation list. In our case, the task of the recommender is to correctly predict which events a given test user will provide positive RSVPs in the future (test set). We have used the well known Normalized Discounted Cumulative Gain (NDCG) metric truncated to 20

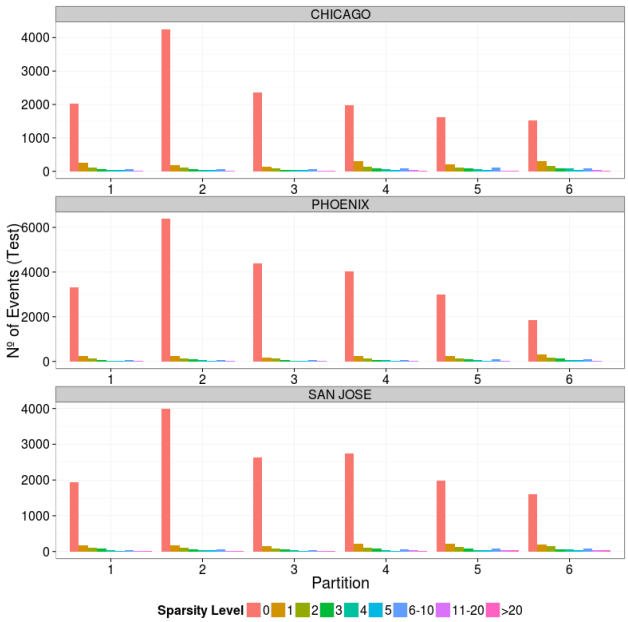


Figure 5: Event Sparsity Level per Partition

recommendations. So, the $NDCG@20$ for a given user u is defined as follows.

$$DCG@20 := \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{log_2(i+1)} \quad (1)$$

$$NDCG@20 := \frac{DCG@20(u)}{IDCG@20(u)} \quad (2)$$

In Equation 1 above, rel_i is 1 or 0 if the event at position i is relevant or not respectively, and the function $IDCG_p(u)$ returns the perfect ranking value, acting as a normalization term.

4.4 Experimental Results

In this section we compare the following well know top-n item recommendation algorithms from the literature:

- *Random*: The recommendation list is randomly generated.
- *Most-Popular*: The candidate events are ranked in descending order of popularity. We define popularity of an event as the number of positive RSVPs received.
- *Location-Aware*: This algorithm ranks the events based on their distances to the users home, assuming that nearby events are more likely to be attended by the user. This algorithm does not rely on RSVP data.
- *BPR-MF*: The Bayesian Personalized Ranking [7] is a state-of-the-art matrix factorization-based algorithm for top-n item recommendation. Its hyper-parameters were defined by grid-search where the best results were achieved with 50 latent factors, 0.1 for the gradient descent learning rate and 500 iterations.

- *User-KNN* and *Item-KNN*: Correspond to the classic k-nearest neighbor collaborative filtering based on users or items. The Collaborative Analysis of Section 3.5 have an important role in these algorithms. After a grid-search, the neighborhood size was set to 100 for both algorithms.

- *Logistic-Regression*: We also tested an hybrid algorithm where the event scores of all aforementioned algorithms (except the *Random*) are fed into a logistic regression model.

Figure 6 displays the recommendation performances of each algorithm in each city considered. In spite of the high sparsity levels, the KNN based algorithms attain the best performances in comparison to the other individual algorithms. One possible explanation to this result is that, in many cases, users who will attend the same event are already friends or acquaintances and therefore may have mutual influence on the selection of future events. The *Location-Aware* algorithm is comparable to the *Most-Popular*, leading one to conclude that the geographic distance, although carrying some signal, is not among the main reasons affecting the decision of a user in attending or not an event. Another potential reason for this result is the inaccuracy of the users home position that is approximated from its IP address. Nonetheless, since this algorithm does not rely on RSVP data, it represents a good alternative for full cold-start scenarios. Although BPR-MF is usually better than simpler KNN based recommenders in other domains, this is not the case here. This might be related to the extreme level of sparsity of EBSN, which is not observed in other papers that concentrate their experiments on denser regions of collaborative data.

The *Logistic-Regression* approach is at least as good as the *Item-KNN*, attaining slightly better results in San Jose. Nonetheless, it is worth noticing that the overall $NDCG@20$ values are very low, achieving at most 0.3 in the best cases.

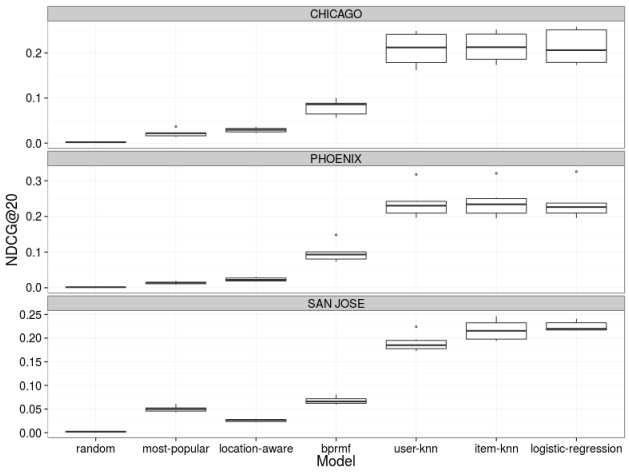


Figure 6: $NDCG@20$ results per algorithm for all cities

The algorithms were also evaluated in terms of the event sparsity level. Here we want to investigate which events are more likely to be correctly recommended according to their sparsity levels. We want to answer questions like: events having 20 or more positive RSVPs are more likely to be correctly recommended than events having 10 or less RSVPs? Figure 7 shows the results of this analysis for all algorithms in each sparsity level considered. The x-axis encode the algorithms and the colors encode the sparsity levels.

As expected, the more positive RSVPs an event has, the more likely it is to be correctly recommended by all recommendation algorithms, except the *Location-Aware* that does not use RSVP information, the *Item-KNN* and the *Logistic-Regression* that seems to deteriorate in Phoenix with the decrease of sparsity.

5. CONCLUSIONS AND OUTLOOK

In this paper we approached the problem of event recommendations in EBSN. We showed that this task is more challenging than typical recommendation domains investigated by the literature since EBSN data is inherently cold-start. One alternative to alleviate this problem is to use RSVP data, although this data is still very sparse.

We analysed important features of EBSN that can affect the design of effective event recommenders and compared well known algorithms on real data collected from the popular EBSN Meetup. Our main findings are summarized below:

- RSVPs tend to be given close to the occurrence of the event.
- The largest majority of events are cold-start.
- Despite the high sparsity of RSVP data, KNN-based algorithms appear as the best single alternative.
- Matrix-factorization does not perform as well in this domain as it does in other more typical domains.

In future work we intend to investigate the influence of group membership on event attendance and more sophisticated context-aware models to exploit the contextual data of events, such as time, tags and events descriptions.

6. REFERENCES

[1] T. De Pessemier, J. Minnaert, K. Vanhecke, S. Dooms, and L. Martens. Social recommendations for events. In *CEUR workshop proceedings*, volume 1066, page 4, 2013.

[2] H. Khrouf and R. Troncy. Hybrid event recommendation using linked data and user diversity. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 185–192, New York, NY, USA, 2013.

[3] X. Liu, Q. He, Y. Tian, W.-C. Lee, J. McPherson, and J. Han. Event-based social networks: linking the online and offline social worlds. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 1032–1040, New York, NY, USA, 2012. ACM.

[4] E. Minkov, B. Charrow, J. Ledlie, S. Teller, and T. Jaakkola. Collaborative future event recommendation. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 819–828, New York,

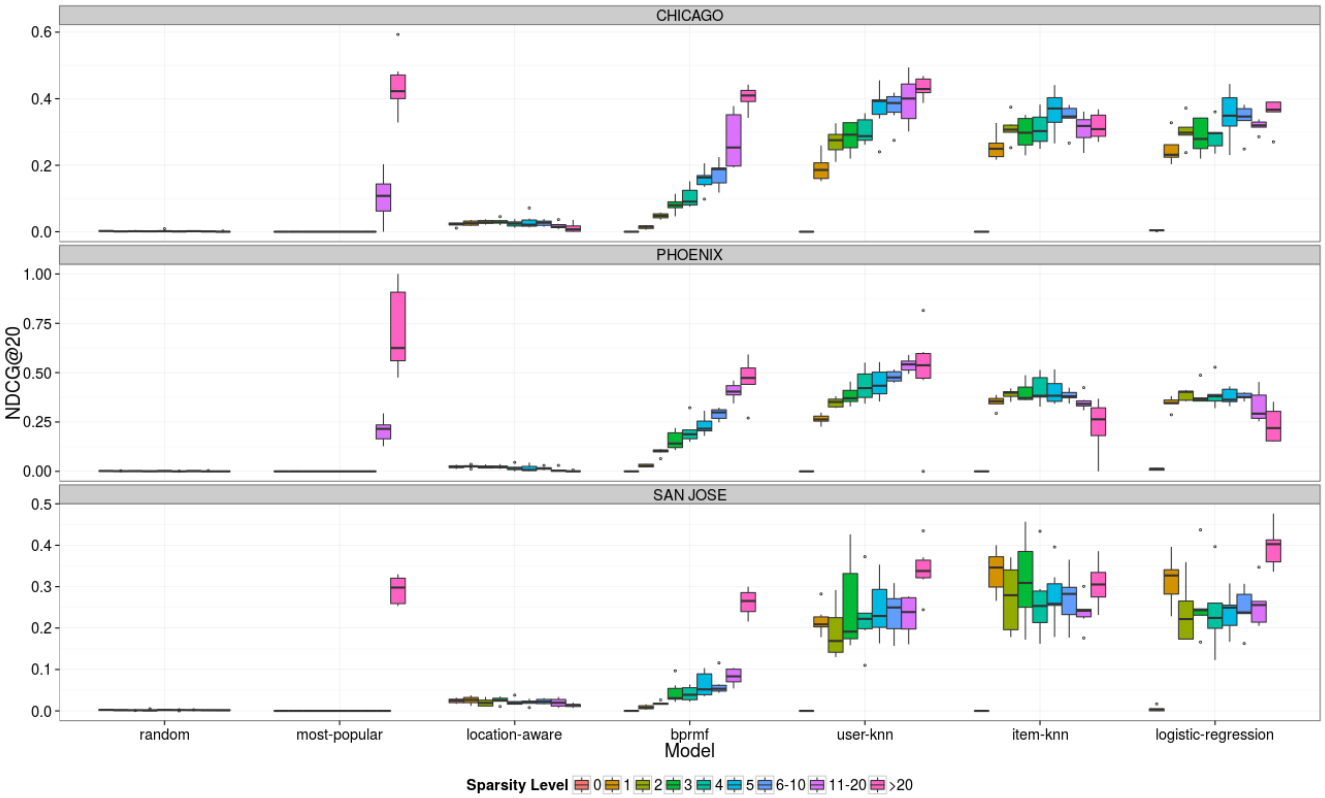


Figure 7: *NDCG*@20 results per algorithms and event sparsity level for all cities

NY, USA, 2010. ACM.

[5] T. D. Pessemier, S. Coppens, E. Mannens, S. Dooms, L. Martens, and K. Geebelen. An event distribution platform for recommending cultural activities. In *WEBIST*, pages 231–236, 2011.

[6] D. Quercia, N. Lathia, F. Calabrese, G. Di Lorenzo, and J. Crowcroft. Recommending social events from mobile phone location data. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM '10, pages 971–976, Washington, DC, USA, 2010. IEEE Computer Society.

[7] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 452–461, Arlington, Virginia, United States, 2009. AUAI Press.

Sentiment Visualisation Widgets for Exploratory Search

Eduardo Graells-Garrido
Universitat Pompeu Fabra
Barcelona, Spain
eduard.graells@upf.edu

Mounia Lalmas
Yahoo Labs
London, UK
mounia@acm.org

Ricardo Baeza-Yates
Yahoo Labs
Barcelona, Spain
ricardo.baeza@barcelonamedia.org

ABSTRACT

This paper proposes the usage of *visualisation widgets* for exploratory search with *sentiment* as a facet. Starting from specific design goals for depiction of ambivalence in sentiment, two visualization widgets were implemented: *scatter plot* and *parallel coordinates*. Those widgets were evaluated against a text baseline in a small-scale usability study with exploratory tasks using Wikipedia as dataset. The study results indicate that users spend more time browsing with scatter plots in a positive way. A post-hoc analysis of individual differences in behavior revealed that when considering two types of users, *explorers* and *achievers*, engagement with scatter plots is positive and significantly greater *when users are explorers*. We discuss the implications of these findings for sentiment-based exploratory search and personalised user interfaces.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Search process*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Interaction styles*

Keywords

Visualisation Widgets; Sentiment Analysis; Exploratory Search; Wikipedia; Individual Differences

1. INTRODUCTION

Search is a common activity on the web today, performed by almost everyone. Even though search engines have been present for many years on the web, today most of them still have the initial text-based interface, which is shown to all users, in spite of the emergence of several paradigms in information seeking and user modeling that could be used to personalise it.

One of those paradigms in information seeking is *Exploratory Search* [20], where a concrete information need is not always present and information seekers usually engage in learning and investigation strategies instead of plain lookup of documents. One way to support exploratory search is by using faceted search interfaces [15], where information seekers have access to several orthogonal dimensions of the information space even when there is no explicit information need. This approach allows information seekers to explore the information space without writing a query. However, its implementation requires a structure in the underlying data that is not always available. A solution to this is to extract meta-data from the information space to provide the needed structure. In this paper we adopt this approach to build a facet for an unstructured information space, by using attributes annotated in text documents calculated through *sentiment analysis* [23].

Users are getting used to see and understand emotional annotations in text, as popular websites such as news outlets and e-commerce sites have user ratings and reviews, which are inherently emotional. However, to the extent of our knowledge, this emotionality inherent in the text has not been exploited to encourage exploratory search. This is somewhat surprising as it is not uncommon for information seekers to have sentiment in mind when performing some tasks, for instance, when browsing user reviews to find restaurants, movies, places, or other things where the emotional or affective responses of other users are important. When sentiment is actually depicted in these scenarios, its depiction is usually focused on a single variable that goes from negativity to positivity, and often this variable is discrete, as in the case of a simple text classification of *negative, neutral or positive*, or a *n-star ratings*. Using only a single variable hides the richness of the various sources of sentiment and their distribution. For instance, in review sites, the only way to find the sentiment diversity is by manually browsing the list of reviews, as a *n-star* rating simply displays an average.

Most sentiment depictions do not consider the ambivalence present in text, which means that a document may have both positive and negative content at the same time. In our approach we build *visualisation widgets* [11] where the widget visualises ambivalent sentiment as a facet for search results. Although this may be feasible using the typical text widgets used in faceted interfaces, our work focuses on visualisation to provide an exploratory experience that is engaging. In this regard, our research question is: **do visual approaches foster exploration in a sentiment-based exploratory search setting?** To answer this question, we defined a set of design goals for visualisation widgets in our setting. We fulfilled those goals with two interactive visuali-

sations based on known paradigms: *scatter plots* and *parallel coordinates*, and tested these visual approaches against a text-based baseline. We performed quantitative and qualitative analysis to analyse the results and see if exploration using sentiment-based visualisation widgets is fostered from a user engagement perspective.

As information space for a case study we chose Wikipedia, an open encyclopedia where anyone can contribute and edit articles. Wikipedia is a prominent social media platform, which contains articles with inherent sentimental content [21]. In addition, its users, both readers and editors, search more on average than those never or hardly using Wikipedia [27]. This prominence of search in Wikipedia, its publicly available content and the existence of sentiment in it, made it a good candidate to use as basis to evaluate our visualisation widgets.

This paper contributes a user evaluation of exploratory behavior in the presence of sentiment in both user intent and information space. Based on the study results, we show that users are spend more time performing tasks when using scatter plots. This additional time is explained by positive engagement *when users are explorers*, based on qualitative feedback and the analysis of individual differences [6]. The analysis of individual differences was based on how users interact with search interfaces: we identified two types of users, *explorers* and *achievers* [4]. Our results suggest that scatter plots are more suitable for explorers, as they significantly increase engagement, opening a path to research which visualisations or interface elements are more suited for achievers, for whom we did not find a particular visualization that increased engagement.

2. RELATED WORK

Although bar and pie charts are common depictions to visualise sentiment, there are other approaches to visualise it. In [14], affect in document collections is visualised with *wind rose charts*. *Heatmaps* are used in [10] to encode the average sentiment of a period of time. In the context sentiment in reviews, [1] used *histograms* and [5] used *treemaps*. *Scatter plots* are used in [24] to visualise ambivalence in public opinions. This is the most similar work to ours from a visualisation perspective, as other previous work focused on unidimensional color-coding of sentiment. We also use *parallel coordinates* [17], which have not been used before in this context to the extent of our knowledge.

We Feel Fine [18] is a search engine where information seekers can answer questions with an explicit sentiment component such as *"how did the U.S. feel when Obama was elected?"* and obtain a visualisation of search results. The purpose of the visual depiction is artistic, and results can be filtered through facets of meta-data such as gender, age and mood. With regard to *visualisation widgets*, [11] depicts facets such as time, geo-location and topics. In [7], treemaps are used to depict a hierarchical facet. It was found that the usage of visualisation had positive impact on perceived task difficulty, repository understanding and enjoyment. Our work extends [11], as we present widgets for a specific facet that could be used among other widgets.

In many search scenarios the information seeker is not an expert who has to perform a concrete, specialised task. Hence, non-experts have a diversity of expertise, knowledge and experience with computer systems. Because not even two persons are equal, the study of *individual differences* [6] proves to be useful, as it allows to find which factors, from demographic, cultural and behavioral, have impact on user modeling and user generated content. In informational contexts, personality traits have been considered to define a user taxonomy of *fast surfers, broad scanners and deep divers* [16]. In virtual worlds, a popular taxonomy is based on how people interact with the world: *achievers* and *explorers* [4]. We consider the latter taxonomy as a first step towards more complex ones.

3. SENTIMENT VISUALISATION

We start from a scenario where the information seeker already has a query, but one that is not necessarily final. We consider learning and investigation activities [20] as focus for design goals. Our design goals are:

Depict ambivalence. Typical sentiment depictions only show one sentiment attribute, often as a mixture of both positivity and negativity to find out which one is prevalent. However, ambivalence is present in many categories and genres of textual content, including public discourse, fiction and news articles. Information seekers should be able to see the duality of sentiment in text, depicted in terms of *positivity* and *negativity*, or ambivalence directly as in [24].

Show sentiment distribution. Following the scenario presented by [18], questions such as *"How did the U.S. feel when Obama was elected?"* have an implicit request for seeing distribution and an explicit request for seeing sentiment.

Allow sentiment filtering. The interface of [18] uses sentiment keywords such as *mood*, *sad*, *happy*, *depressed*, to filter results according to emotion. In text query interfaces, information seekers depend on the context at hand, and a keyword search may exclude the desired sentimentality because the information seeker did not use "matching" keywords. Visual filtering would remove the burden of writing the correct keywords from users and provide a more flexible tool for filtering according to emotion.

A visualisation widget that conforms to these design goals will allow information seekers to understand how sentiment is distributed in an information space, to see the ambivalence present in it and to filter documents in order to learn and investigate according to their own criteria.

3.1 Visualisation Widgets

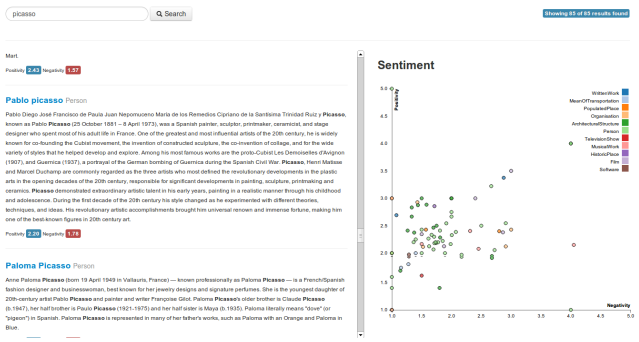


Figure 1: Scatter Plot widget in our prototype search interface.

Following our design goals, we implemented two visualisation widgets: *scatter plots* and *parallel coordinates* [17]. We

Figure 2: Parallel Coordinates widget in our prototype search interface.

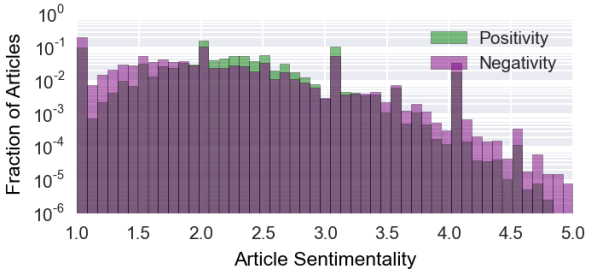


Figure 3: Distributions of Positivity and Negativity in our dataset of Wikipedia articles (using log-scale).

chose two known paradigms because our research question is not about new visualisations, and using only one paradigm might bias the results of our study.

Scatter Plot. Figure 1 shows the scatter plot widget. Each result is a circle whose position is determined by both sentiment attributes: positivity is mapped to the x-axis and negativity is mapped to the y-axis. To filter results, the information seeker can draw a rectangle over the visualisation canvas, selecting only the circles that are positioned inside the rectangle.

Parallel Coordinates. Figure 2 shows the parallel coordinates widget, where each attribute is a different axis: negativity is mapped to the left axis and positivity is mapped to the right axis. Each result is represented as a line that connects the corresponding value of its attributes in each axis. To filter results, the information seeker can draw a rectangle over the axes, selecting only the lines that begin (or end) inside the selected range.

In both widgets we display *positivity* and *negativity* for each item (*depict ambivalence*). We use transparency to showcase density and prevent occlusion (*show sentiment distribution*). We use *brushing and linking* [13] to *allow sentiment filtering*: when the information seeker restricts or widens the ranges of sentiment of interest in the widget, the list of results is updated immediately, and when the information seeker selects a result from the text list, the corresponding element on the visualisation is highlighted. The results filtered out are drawn with more transparency to indicate that they are out of focus. Color coding of points and lines is used to encode item categories if available.

4. SENTIMENT IN WIKIPEDIA

We test our approach on Wikipedia[1] – a multilingual, web-based, free-content encyclopedia, written collaboratively by a large number of volunteers. Although Wikipedia has a *neutral point of view* policy [28], neutral is not equal to emotionless. It is possible to find sentiment in content in Wikipedia, as it contains biographies, disasters, awards, celebrations and summaries of fiction, among other categories.

Dataset. We use a dataset of 737,863 english articles from Wikipedia with annotated sentiment [21]. Each article is annotated with two scores: *positivity* (from 1 to 5) and *negativity* (from 1 to 5). Note that positivity does not imply that negativity is absent, and vice-versa: ambivalence is almost always present in text. The sentiment values of an article are calculated based on the content of the article itself, and that of other articles linking to it. In other words, each article becomes annotated with the sentiment scores of its own content, plus that of the associated articles. Figure 3 shows the distributions of both scores in our dataset.

Although the distributions are skewed towards lower values of sentimentality (the average positivity is 2.17 ± 0.64 and the average negativity is 1.92 ± 0.78), there are articles with high values of sentiment attributes. The distributions confirm that there is sentiment in Wikipedia, creating the opportunity to use our visual approach to search and explore it.

5. USER EVALUATION

We performed a small-scale usability study in a lab-setting with 13 participants (5 male and 8 female; 5 aged 20–29, 6 aged 30–39, 1 aged 40–49, and 1 unknown), who scored their knowledge in visual web search as 3.46 ± 1.13 in average (using a Likert scale from 1 to 5). Participants were recruited from open calls in social networks and did not receive compensation for participating in the study.

Apparatus. We built a prototype search engine that indexed extended abstracts[2] of the 737,863 articles in the dataset. The user interface contained the following elements: query box, the number of results, the list of results with each article's title, extended abstract and sentiment values in text form, and the visualisation widgets. Given a query, the search engine returned a list of articles (maximum count: 200) ranked using the BM25 scoring algorithm [3]. All participants used the same computer, a notebook of 15 inches screen with resolution of 1440×900 pixels. In the experimental prototype, categorical color coding was based on the DBPedia ontology class of each article [2]. This ontology is shallow, and we restricted the depth of ontologies associated to search results to be able to create a color mapping understandable for users.

Design and Procedure. The study used a within-subjects design. Each participant tested three treatments: baseline (*BA*, a text-based widget of buttons to filter the results, shown in Figure 4), scatter plot (*SC*, shown in Figure 1) and parallel coordinates (*PC*, shown in Figure 2). The order of pairs *(task, treatment)* was randomised for all participants to avoid positional bias.

[1] `http://wikipedia.org`

[2] Defined as the first section of each Wikipedia article.

After performing each task, participants were asked to answer five questions about aesthetic value of the interface[3]. A Likert scale from 1 (*strongly disagree*) to 5 (*strongly agree*) was used for this purpose. In addition, participants were asked to write a small summary of the results they found, and were asked about their *perceived time* [22] of task completion. After performing all tasks, participants filled a feedback questionnaire about their thoughts on the interfaces, how they would describe the different widgets and if they had any comments and suggestions. Finally, we logged each query and calculated the actual time of completion for each task in order to estimate the difference between perceived time and real task completion time. This metric is called *subjective duration assessment* [8] and has been interpreted before as *cognitive engagement* [9]: lesser perceived time than task completion indicates positive engagement.

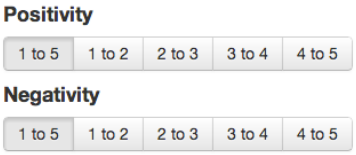


Figure 4: Text-based widget used in the baseline approach of the user study.

Tasks. Participants were asked to perform three exploratory search tasks, one task per treatment. One task was personalised in terms of what they had to search for, while the remaining tasks were based on the definitions in [19]:

> *"Think about a topic you like, and find five articles with a highly negative connotation. Then think about a topic you do not like, and find five articles with a highly positive connotation".*

> *"Imagine you are taking a class called 'Art in Europe'. For this class you need to write a research paper on some aspect of an art movement, but have yet to decide on a movement you will focus. Use the system to find three artists within that movement: one artist with a positive connotation on Wikipedia (but slightly negative), one with a negative connotation (but slightly positive), and one with high emotionality (by being highly positive and negative at the same time), so that you might make a decision as to which movement you will write about."*

> *"Your professor wants you to write a paper comparing the consequences of war in three countries. Use the system to find three countries which have highly emotional (high positivity and negativity) events or works as consequences of the war. Find three events or works for each country."*

5.1 Results

To answer our research question, **do visual approaches foster exploration in a sentiment-based exploratory search setting?**, we tested the following hypothesis: *in exploration on sentiment-based scenarios, participants perform more queries and spend more time when using visualisation widgets*, by evaluating the two visualisation widgets against a text-based baseline. Post-hoc differences in means were tested using *Wilcoxon's Ranked Sums* (Bonferroni corrected) after performing *Kruskal-Wallis* analysis of variance on the three groups.

Results in Table 1 partially support our hypothesis. There is a significant group difference in task time ($p < 0.05$),

[3]Example: *The search system was aesthetically appealing.*

	BA	PC	SC	K	p
Query Count	7.38	14.15	19.31	5.18	0.07
Task Time (s)	463.00	745.23	1035.92*	6.83	0.03
Perceived Time	507.69	770.77	761.54	3.73	0.15
Cognitive Engagement	−44.69	−25.54	274.38	3.53	0.17
Aesthetics	13.54	15.77	17.08	2.06	0.36

Table 1: Experimental results for all participants ($N = 13$). p values correspond to Kruskal-Wallis analysis of variance (K). *: post-hoc comparison between SC and BA significant at Bonferroni corrected $p < 0.017$.

	Achievers ($N = 6$)	Explorers ($N = 7$)	p
Total Queries	25.33	54.14	0.02
Total Time	1372.50	2991.26	0.00
Queries BA	6.67	8.00	0.83
Queries SC	8.83	28.29	0.02
Queries PC	9.83	17.86	0.13
Task Time BA	326.67	579.86	0.20
Task Time SC	566.67	1438.14	0.02
Task Time PC	479.17	973.29	0.03
Perceived Time BA	550.00	471.43	0.78
Perceived Time SC	600.00	900.00	0.09
Perceived Time PC	620.00	900.00	0.07
C. Engagement BA	−223.33	108.43	0.02
C. Engagement SC	−33.33	538.14	0.03
C. Engagement PC	−140.83	73.29	0.31

Table 2: Post-hoc comparison with *Wilcoxon's Ranked Sums* of results for explorers and achievers.

and post-hoc testing revealed that SC task time is larger ($p < 0.017$, Bonferroni corrected) than the other two groups. In terms of query count, no significant effect was found, although there is a trend towards greater amount of queries in visual approaches ($p < 0.1$). Hence, using SC, users spend more time exploring, but do not necessarily perform more queries. No significant differences were found in aesthetic perception, perceived time and cognitive engagement.

To explain quantitatively the differences in task time, we considered the following user taxonomy: *achievers* (those who *"are interested in doing things to the game, i.e. in ACTING on the WORLD"*), and *explorers* (those who *"are interested in having the game surprise them, i.e. in INTERACTING with the WORLD"*) [4]. We define achievers ($N = 6$) as those users who are in the bottom 50% w.r.t. the geometric mean of total task time and total queries issued; and explorers as the rest ($N = 7$), that is, those in the upper 50%. In this way, achievers want to finish the task fast and quickly, while explorers are interested to see how the system can surprise them. Table 2 reports differences in means for both groups in all approaches. There are significant differences (measured with *Wilcoxon's Ranked Sums*) on total queries and total time, which were expected as they are consequence of the user taxonomy. However, other significant differences emerge: *1)* explorers issue more queries than achievers using SC ($p < 0.05$), but not in BA and PC; *2)* explorers spend

more time when using SC and PC ($p < 0.05$) but not when using the baseline; *3)* explorers have greater positive cognitive engagement than achievers when using SC and BA (and SC's engagement is almost 5 times BA).

Qualitative Feedback. We included open-feedback questions in order to understand and explain the quantitative results. We use [P*i*] to refer to participant *i*.

The baseline (BA) was characterised by participants as *"boring"* [P8] but *"the easiest for me to find results"* [P4]. It was perceived as a tool for *"discriminating"* [P11] and *"filtering"* [P10]. As expected, the *"filters were really easy to use"* [P3], as participants are used to this kind of interface. However, most of the positive feedback for BA was related to the act of performing the task, and not on how the actual users felt about the text-based widget: *"I think the most useful one is the buttons one because it has more precise information reflected on it."* [P10], although not everyone felt comfortable with it: *"The one with the numbers was misleading for me"* [P6].

Regarding the visual approaches, the scatter plot (SC) was described as *"attractive"* [P8], *"like a classifier"* [P4], as well as a *"spectrum"* [P10] or a *"map"* [P11], perhaps referring to how a scatter plot allows to classify elements according to their position on the screen. We expected that users would have been familiar with SC, as in: *"[BA] and [SC] are easy to use. They are helpful and easy to understand"* [P3]. However, it also *"needs more concentration"* [P6]. Some users were more vocal in their enthusiasm for this approach: *"this is the task that I enjoy the most! I liked pretty much the graphics"* [P8], *"this is the approach I liked the most, it was easier to filter the results"* [P9], indicating that scatter plots not only are familiar, but also they generate a more positive, emotional reaction. Parallel Coordinates (PC) produced an ambivalent reaction. On one hand, it was described as *"interesting"* [P8],*"much more cooler then the other one"* [P1], *"the high-low thing helps me to know if it is positive or negative faster. I really like how [PC] worked"* [P6], and *"the sentiment indicator [PC] helps in the task"* [P9]. On the other hand, users claimed that *"[PC] was not appealing nor easy to understand or use"* [P3] and that *"it's confusing"* [P11].

In addition to visualisation feedback, participants suggested some features that could improve our system prototype: *"drawing the box around the numbers in each axis is too complex, I would have preferred to have another way of controlling the sentiment in the results. Maybe even a simple slider"* [P9, referring to PC], *"a grid in the circles system would help to have more exact information about the scores at a glance"* [P11, referring to SC]. With respect to the search results, some users expressed they were not satisfied with their quality: *"the search engine does not work properly, distracting myself from the task"* [P3], *"the search was very frustrating, as the searches often did not yield many results"* [P7]. Some users thought about whether they would use a system like this in the future: *"the system was useful but I don't search using sentiments frequently... Maybe when searching for the politic situation of a country I would use it"* [P9].

6. DISCUSSION AND IMPLICATIONS

User Engagement and Visualisation Widgets. In the experiment the SC group spent more time performing the exploratory tasks than BA and PC. Whether this is a good scenario, if users spend more time because they are engaged, or if they spend more time because the visualisation is impeding the task at hand, is something that needs to be determined and explained. We attributed part of the longer task time of SC in Table 1 to a positive user experience *when users are explorers*, as explorers performed more queries and spent more time, while at the same time they showed a significantly greater positive cognitive engagement. Moreover, the qualitative feedback received by SC was positive, indicating that it is unlikely a negative experience when using that treatment to perform the task. This positive engagement result is consistent with previous work [7], where users expressed more enjoyment when using visualisation techniques in the search interface. Since not all visualisations are perceived equally, it makes sense that some visualisations engage users and some do not, as well that a visualisation might engage one kind of users only. In this aspect, our results are limited to explorers only, because no significant patterns were found for achievers, although some users explicitly favored the parallel coordinates widget as attractive. As there might not be a globally *better* visualisation for all users, it remains to be seen which visualisation is more likely to engage *achievers*.

Personalisation of User Interfaces. Individual differences based on exploratory behavior provide a base for a contextual personalisation of user interfaces, as a to complement to content personalisation based on user generated content. When considering individual differences, we restricted the definition of exploration as the geometric mean of task time and number of queries, which allows to implement the *explorers and achievers* taxonomy based on: *1) previous activity on the search system*, making possible to provide this type of widget-based personalisation when query logs and interaction data are available; *2) granularity of a query*, in the sense of how *"good latin restaurant in Born neighborhood"* indicates something one wants to achieve, while *"restaurants in Barcelona"* indicates something one wants to explore. Considering availability of this user taxonomy, user interfaces can be personalised to increase engagement in users performing learning and investigation tasks by using scatter plots instead of text widgets.

Limitations. In terms of implementation, participants in our experiment expected better results than those provided by our prototype implementation. The effect of those unfulfilled expectations over the obtained results is unknown and should be considered in future experiments. In addition, trending differences in behavior surfaced on quantitative results, perhaps a limitation of the small-scale of the user study. We believe these limitations can be fully addressed in a larger-scale experiment using an improved search engine and following the TREC interaction track guidelines [12].

7. CONCLUSIONS

This paper presented results on our research of sentiment visualisation widgets for exploratory search. We defined design goals in this scenario, and implemented two visualisations based on known techniques: scatter plots and parallel coordinates. Both approaches were evaluated against a baseline of text-based links for exploring search results. Even though the scale of our study is small, we found statistical evidence of users spending more time performing tasks when using scatter plots. Through analysis of qualitative feedback and individual differences, we explained that time difference

as positive engagement with the visualisation widget. In particular, the individual differences analysis focused on a user taxonomy that defines *explorers* and *achievers*: those who *interact* in the world and those who *act* in it, respectively. Our results indicate that scatter plots are suitable for explorers, as they are more engaged in a positive way when using that visualisation paradigm in comparison to a text baseline and the parallel coordinates visualisation. Hence, in the presence of explorers, we suggest search and exploratory systems to personalise the user interface with scatter plots to browse sentiment, to increase user engagement and foster exploration.

Future Work. Our approach assumes the presence of sentiment meta-data, which may be added algorithmically to any text collection. The usage of Wikipedia proves to be useful as there is a varying degree of sentimentality across the subset we studied. As future work we will consider other scenarios, such as reviews, media and social networks, where the amount and variation of sentiment will likely be greater. In addition, we will consider more complex behavioral taxonomies based on personality traits [16], as personality traits in social networks can be predicted in social media [25, 26]. Finally, we will explore the possibilities of our approach in other bivariate related contexts such as political leaning.

Acknowledgments. This work was partially funded by Grant TIN2012-38741 (Understanding Social Media: An Integrated Data Mining Approach) of the Ministry of Economy and Competitiveness of Spain.

8. REFERENCES

[1] B. Alper, H. Yang, E. Haber, and E. Kandogan. Opinionblocks: Visualizing consumer reviews. *IEEE VisWeek Workshop on Interactive Visual Text Analytics for Decision Making*, 2011.

[2] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. Dbpedia: A nucleus for a web of open data. *The Semantic Web*, pages 722–735, 2007.

[3] R. Baeza-Yates and B. Ribeiro-Neto. *Modern information retrieval: the concepts and technology behind search, Second edition*. Addison-Wesley, Pearson, 2011.

[4] Richard Bartle. Hearts, clubs, diamonds, spades: Players who suit muds. *Journal of MUD research*, 1(1):19, 1996.

[5] Giuseppe Carenini, Raymond T Ng, and Adam Pauls. Interactive multimedia summaries of evaluative text. In *Proceedings of the 11th international conference on Intelligent user interfaces*, pages 124–131. ACM, 2006.

[6] Chaomei Chen, Mary Czerwinski, and Robert Macredie. Individual differences in virtual environments–introduction and overview. *Journal of the American Society for Information Science*, 51(6):499–507, 2000.

[7] Edward Clarkson, Krishna Desai, and James Foley. Resultmaps: Visualization for search interfaces. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):1057–1064, 2009.

[8] Mary Czerwinski, Eric Horvitz, and Edward Cutrell. Subjective duration assessment: An implicit probe for software usability. In *Proceedings of IHM-HCI 2001, vol. 2*, pages 167–170, 2001.

[9] Munmun De Choudhury, Scott Counts, and Mary Czerwinski. Identifying relevant social media content: leveraging information diversity and user cognition. In *Proceedings of the 22nd ACM conference on Hypertext and hypermedia*, pages 161–170. ACM, 2011.

[10] Nicholas Diakopoulos, Mor Naaman, and Funda Kivran-Swaine. Diamonds in the rough: Social media visual analytics for journalistic inquiry. In *Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on*, pages 115–122. IEEE, 2010.

[11] M. Dörk, S. Carpendale, C. Collins, and C. Williamson. Visgets: Coordinated visualizations for web-based information exploration and discovery. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1205–1212, 2008.

[12] Susan T Dumais and Nicholas J Belkin. The trec interactive tracks: Putting the user into search. *TREC: Experiment and evaluation in information retrieval*, pages 123–152, 2005.

[13] S.G. Eick and G.J. Wills. High interaction graphics. *European Journal of Operational Research*, 81(3):445–459, 1995.

[14] M.L. Gregory, N. Chinchor, P. Whitney, R. Carter, E. Hetzler, and A. Turner. User-directed sentiment analysis: Visualizing the affective content of documents. In *Proceedings of the Workshop on Sentiment and Subjectivity in Text*, pages 23–30. Association for Computational Linguistics, 2006.

[15] M.A. Hearst. *Search user interfaces*. Cambridge University Press, 2009.

[16] Jannica Heinström. *Fast surfers, broad scanners and deep divers*. Åbo Akademis förlag, 2002.

[17] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(2):69–91, 1985.

[18] S.D. Kamvar and J. Harris. We feel fine and searching the emotional web. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 117–126. ACM, 2011.

[19] Bill Kules and Robert Capra. Creating exploratory tasks for a faceted search interface. In *Proceed ings of 2nd Workshop on Human–Computer Interaction*, pages 18–21. Citeseer, 2008.

[20] G. Marchionini. Exploratory search: from finding to understanding. *Communications of the ACM*, 49(4):41–46, 2006.

[21] Yelena Mejova, Ilaria Bordino, Mounia Lalmas, and Aristides Gionis. Searching for interestingness in wikipedia and yahoo! answers. *International World-Wide Web Conference (WWW)*, 2013.

[22] Heather L O'Brien and Elaine G Toms. The development and evaluation of a survey to measure user engagement. *Journal of the American Society for Information Science and Technology*, 61(1):50–69, 2009.

[23] B. Pang and L. Lee. *Opinion mining and sentiment analysis*. Now Pub, 2008.

[24] Galen Panger, Bryan Rea, and Steven Weber. Visualizing ambivalence: showing what mixed feelings look like. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, pages 1029–1034. ACM, 2013.

[25] Daniele Quercia, Michal Kosinski, David Stillwell, and Jon Crowcroft. Our twitter profiles, our selves: Predicting personality with twitter. In *Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom)*, pages 180–185. IEEE, 2011.

[26] Daniele Quercia, Renaud Lambiotte, David Stillwell, Michal Kosinski, and Jon Crowcroft. The personality of popular facebook users. In *Proceedings of the ACM 2012 conference on computer supported cooperative work*, pages 955–964. ACM, 2012.

[27] Robert West, Ingmar Weber, and Carlos Castillo. A data-driven sketch of wikipedia editors. In *Proceedings of the 21st international conference companion on World Wide Web*, pages 631–632. ACM, 2012.

[28] Wikipedia. Wikipedia:neutral point of view — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/wiki/Wikipedia:Neutral_point_of_view`, 2013.

Personal Life Event Detection from Social Media

Smitashree Choudhury
Knowledge Media Institute
The Open University
United Kingdom
smitashree.choudhury@open.ac.uk

Harith Alani
Knowledge Media Institute
The Open University
United Kingdom
h.alani@open.ac.uk

ABSTRACT

Creating video clips out of personal content from social media is on the rise. MuseumOfMe, Facebook Lookback, and Google Awesome are some popular examples. One core challenge to the creation of such life summaries is the identification of personal events, and their time frame. Such videos can greatly benefit from automatically distinguishing between social media content that is about someone's own wedding from that week, to an old wedding, or to that of a friend. In this paper, we describe our approach for identifying a number of common personal life events from social media content (in this paper we have used Twitter for our test), using multiple feature-based classifiers. Results show that combination of linguistic and social interaction features increases overall classification accuracy of most of the events while some events are relatively more difficult than others (e.g. new born with mean precision of .6 from all three models).

Keywords

Social Web, social media, event detection, personal life events

1. INTRODUCTION

With the wide spread of social media sites (e.g. Twitter, Facebook, YouTube), millions of of people use them on daily basis to communicate and share information on a wide variety of events, ranging from world events (e.g. World Cup), to personal events (e.g., Wedding, Graduation). Use of these systems serves the multitude of purposes of knowledge sharing, information communication, event organisation, professional collaboration, political expression, as well as socialisation. To put in perspective, more than 500 million of tweets generated in a day[1], millions of photos are uploaded to Facebook every day. There may be differences in terms of content volume created on different platforms depending on the personal preferences and the perceived purpose of the tool, nonetheless most popular online systems are carrying huge amount of data created by individual users in the form of texts, videos, and photos. While technology for data creation and storage has significantly matured and efficiently managed, accessing, managing and processing of such data is still a challenge and can be done by fews experts. Due to the lack of efficient data access mechanism available to normal users, most of the historical data tend to be forgotten or will remain unused.

Access and reuse of such information trove will provide greater insight about the individual user, their preferences, and situational dynamics and result in many useful applications e.g. personalised healthcare, customised training and education, social and community engagement application and life stories. To this end, mining and analysing such content could help identifying one's life milestones and salient events. Identifying interesting and important moments in one's timeline on social media is valuable to services such as Facebook Lookback and Google Awesome, which generates short video clips for users to summarise and visualise their timelines.

In realisation of the importance of events on social media, Facebook [2] has recently generated millions of 1 minute lookback videos of content from users' timelines. Over 270 million video rendered and over 200 million users watched their look back movie in the first two days and more than 50% shared their movie. A project like Intel's Museum of Me[3] follows a similar line to collect data from user's Facebook profile and generate a short video. Purpose of our work (personal life event detection) is a sub-objective of the broader research objective in similar direction i,e, automatic creation of digital documentaries from social media content including interesting and relevant life moments and events.

Event detection from social media content has so far been focused on detecting world events such as earthquakes [Chile, japan], political protests, elections (US, Germany, UK) and planned public events such as entertainment award functions (Oscar, Golden Globe), academic events (conferences), sports event (Olympic). However, detection of personal life events have been mostly overlooked, and only mildly investigated for content recommendation [cite]. Objective of this piece of is to automatically identify interesting and impor-

[1]https://blog.twitter.com/2013/new-tweets-per-second-record-and-how

[2]https://code.facebook.com/posts/236248456565933/looking-back-on-look-back-videos

[3]http://www.intel.com/museumofme/r/index.htm

tant life events of individual users from their social media content, which can be part of their personal digital storybook or memory archive. In this work, we have taken Twitter as the test platform and will extend our research to other systems such as Facebook, Instagram, Pininterest in our future work.

Detecting personal events is non-trivial and may require a combination of multiple approaches for a robust detection result. Unlike public events or events concerning celebrities and well-known personalities, personal events may not be characterised by high activity volume and additional sources of information e.g. blogs or Wikipedia. These events are limited to the concerned person and to her immediate social network (friends and family). In addition to the above problems, microblog sites like Twitter bring its own complexities with short, informal and noisy content. Any meaning-making task on these content has to deal with these idiosyncrasies. Next, we will briefly delve into the concept of a personal event before going into the details of the experimental work.

1.1 Personal Life Events

Personal life events range from recurring events such as birthdays and anniversaries, to very occasional and uncommon events, such as work promotions, and relocation. Events can also be further categorised on an affective scale, from highly positive and pleasant events to to unpleasant events, such as illnesses or accidents and deaths of loved ones. In this paper, we focus on 5 life events (4 positive and 1 negative) i.e. graduation, marriage/engagement, new job, birth of child, and surgery. Our motivation to start with these events inspired by a study [**?**] which lists 6 important memorable life events are "Beginning school", "first full time job", "Falling in love", "Marriage", "Having children;", "Parent's death".

The main contributions of this paper are not on algorithm and its efficiency, but rather on presenting evidence that with effective combination of existing methods and social media data, we can analyse and detects important and critical moments of individuals life., hence the contributions are:

- a thorough study of five personal life events and their idiosyncrasies as reported in social media especially in Twitter .
- detection of life events using both content and interaction features.

This paper is organised as follows: In section 2 we review related work in the field of event detection in social media and in section three, we briefly describe how personal life events are reported on twitter and their characterisation. Section 4 describes our approach which includes feature selection a nd model construction followed by discussion and conclusion in section 5.

2. RELATED WORK

Event detection is now a new research subject, and has been part of studies on topic detection in news stories and other text documents [**?**]. Social media bought multi modal content created by both professional and amateurs leading to a resurgence of interest in detecting social topics and events in this new domain[**?**]. We have been motivated by the need to identify personal life events, which have a great personal value when aggregated over time and location. One of the prerequisites of such a system is the identification of content reporting a real event. Events can be planned events such as cultural events, tech conferences, music award functions, elections or sports event or unplanned events for example, natural disasters, earthquack [**?**] and even generic events such as breaking news events are subject of few studies [**?**][**?**]. Existing studies cover both planned and unplanned events with varying degrees using both machine learning and text analysis techniques. Benson et.al.[**?**] reported detecting concert events from social media stream using city calendar as a target list. Agarwal et. al.[**?**] detected events such as factory fire, labor strike from Twitter stream using a combination of local sensitive hashing and location dictionary. Weng and Lee[**?**] proposed event detection with clustering of word bursts from tweets. Authors in [**?**] proposed a natural disaster alert system using Twitter users as virtual sensors. In their work, they were able to calculate the epicentre of an earthquake by analyzing the delays of the first messages reporting the shock. Social media centric event detection also covers non textual data such as photos and videos, Chen et al.[**?**] discovered social event from Flickr photos by using both user tags and other metadata including time and location (latitude and longintude). Firan et.al[**?**] explored tags, title and description to classify pictures into event categories. Some of the popular approaches used for event detection are spatio-temporal segmentation[**?**], burst analysis in word signals, clustering as well as topic detection techniques.

To the best of our knowledge, we found no prior studies on personal life event detection from social media except one reported in [**?**] where authors tried to detect two life events "marriage" and "employment" and bears some similarity to our work. Our focus is on user level event detection that can be used to build individual digital storyboards form historical data.

3. PERSONAL EVENTS ON TWITTER

We now define the concept of personal life event in the context of Twitter message stream and provide a definition of the problem that we address in this work.

Definition of term "event" differs from domain to domain ranging from Philosophy to cognitive psychology to computing. Despite a lack of uniform definition of the term it embeds a few generic characterstics such as time, participating objects and a location. In this context, we define an event as a real world occurrence with an associated time period and one or more participating objects/agents at a certain location which may or may not be explicitly apparent in tweet messages. According to this definition a tweet needs to reflect a time interval when the event has occurred involving either the user or someone connecting to the user as the participating agent. Based on this abstract notion, we looked into the real data to confirm or re-arrange the definition and devise a strategy for detecting personal events.

3.1 Dataset

As a first step, we collected tweets using Twitter streaming API[4] which allows to crawl some portion of public tweets as and when it comes. We restricted tweets to English language only and crawled for 3-4 hours per day for three weeks. The entire dataset contained around 4 million tweets. Ratio of event tweets to non-event tweets is expected to be extremely skewed as the targeted events are very specific and user centric. So the next logical step is to use a filter mechanism to segregate the event related tweets from the rest and process further. For this initial segregation, we extended the event query with synonyms and related terms and phrases (shown in Table 1). These related terms are mainly synonyms and terms commonly known and used to describe the event of interest. Use of related terms with the main event terms were intended to widen the coverage where users might not be using the exact terms to describe the main events. After filtering we got 9168 tweets for marriage event, 2570 tweets for graduation, 3192 tweets for surgery, 3661 for new job and 2954 tweets for new born. A question may arise about those tweets where the event term may be absent yet the implicit semantics reflects a real event for example. "Welcome to the new member of our family". However, we agree such kind of possible omissions with the present approach and intend to capture them with contextual and historical information as part of our future work. The resulting filtered datasets still contain many irrelevant tweets. For example, *"family have brought a 2nd lawsuit against her, this time to try to annul her marriage"* is not about a marriage event though it contains the keyword. Our task is identify such tweets from genuine event tweets by means of binary classification.

Table 1: Events are their related words.

Event terms	Related Terms
Marriage	"Wedding","Tied the knot","married"
Graduation	"Convocation","commencement "
New Job	" new position","first day at work","job offer"
New Born	"Baby boy","baby girl", "new born"
Surgery	"Operation"

Manual inspection of these tweets revealed that event reporting tends to happen at three time spans; part, present, and future. We also noticed three categories of participating agents (self, others individual and general public). Examples of such diversities are shown in table 2.

In light of these findings, defining a personal event seems to be more tricky and imprecise. Two pertinent questions here are how to resolve the time reference associated with the event and how to associate the right subject (participating agent) with the event. In this study we are only focusing on the events where the time reference can be resolved to a specific time point within a month time interval by automatic means. One such example is "I graduated yesterday", " 26 days to graduation". In both cases, the time of the event can be resolved with help from the timestamp attached to the message. However, ambiguous time references such as "graduation is so close yet so far", "marriage in few weeks time" are ignored.

[4]https://dev.twitter.com/docs/api/streaming

The second dimension where the event reporting differs is on participating agent or affected subject. Event tweets are either about the user who created the tweet or about someone else known to the user and in some cases, about an undefined group of people e.g. group of students. Since our focus is on personal events, ideally we should target self-reported tweets and ignore the rest. But resolving an event to a participating agent needs advanced semantic role labelling which will be our next step of this ongoing work. For this paper, we restricted our attention to generic event detection, hence included all the tweets irrespective of who the affected subject is.

Based on this generic definition, we proceed with our actual experiment task that starts with feature extraction.

4. FEATURE EXTRACTION

After filtering event related tweets from the non-event tweets, we extracted different types of features [**?**] to be used for building event classifiers. We examined several feature categories describing different aspects of tweets and users. Specifically we considered lexical, sentimental and social interaction features.

4.1 Textual Features

Event term: The basic lexical feature of an event is the event term itself and most closely related terms or its synonym "#graduation, convocation" for the event graduation. The synonyms are extracted from Wordnet[5]

Co-occurring textual Features are the features of a term that co-occur significantly along with the event term for example, "cap", "dress", "present", "prom","party" are some of the frequently occurred terms for graduation, while "prayer", "hospital" for surgery. Presence of these terms along with the main event term is expected to boost the detection process. Co-occuring terms were extracted from various tag based social media sites such as Flickr, instagram where terms are described with highly related terms. are These features are event specific and treated as binary values i.e. 1 for presence otherwise 0.

Temporal terms: This feature reflects the presence of time terms in a tweet. Since the content are about an event, it is intuitive to assume that some reference to time is natural and required by definition. For this feature, we used LIWC's time category which includes 68 time terms.

Person reference terms: Since these events are about personal life event one or more reference terms reflecting social relation is expected when the event is about somebody other than the poster, or self reference if the event is about the user.

Sentiment: personal events are expressed with rich emotions both for pleasant or unpleasant events. Sentiments are detected by Sentistrength [**?**] library and proved to be good for social media sentiment detection. Value of this feature ranges from -5(negative) to +5(positive) while +1 to -1 considered as neutral.

[5]http://wordnet.princeton.edu/

Table 2: Events and their examples from Twitter.

Event	Examples
Marriage	Kansas City here we come! It's happening! My sister's marriage this weekend!! :) 8 years ago this day , married to the most loving man on this earth. Congratulations to my beautiful friend, @SheridanMillls, who tied the knot today! ???
Graduation	Happy graduation day, bebe! Congrats cutie pie! http://t.co/YqgNgK9WMw Graduation is just around the corner. Time to start planning programs and certificates. Talk to our print consultants today! 3 sets of graduation picture next week! Hahaha. At last! :)
New Job	First day of a new job.... Kind of dreading it. #officeassistant Starting my new position today. Ayy lmao. Shout out to my cuz Quincy Johnson aka Q. On his new Executive Chef position! ???
New Born	My baby girl is here! Introducing: Halen born naturally May 3rd @ 4:43 pm. Exactly 3 weeks till my babyshower & almost 7 weeks till my baby boy Is born ?
Surgery	Good luck on your surgery today @chloebieber ear surgery ??it went well Everyone please continue to pray for Karlie these next 5 hours. She just went back for her brain surgery. #PrayersForKarlie

Non-Textual and punctuation Features relating to punctuation and emoticons such as presence of "!/?" are expected to add the discriminating qualities of a learning model.

4.2 Interaction and Social Feature

Unigram is a basic model for classification and the result shows a reasonable accuracy including a poor performance for the *new born* event. This motivated us to further explore the feature space and extract more defining attributes of an event in terms of activity and interactions based on the simple logic that important events are bound to generate more attention and activity within the immediate personal network of an individual. Accordingly, we computed the following Twitter specific features concerning to a tweet and the user. These features can be broadly classified into two categories: **1) Activity and 2) Attention.** Activity features (first four in the list below) are based on userÕs activity (tweets, re-tweet and replies) while attention features are the measures of engagement between the user and his/her network (last four features in the list below)

1. Tweets per day: Number of tweets per day a user posts
2. Re-tweets per day: Number of tweets per day a user posts.
3. Replies per day: Number of replies given by the user to other users.
4. Unique mentions per day: Number of unique mention (users addressed) in a day by the user.
5. Number of times the user is mentioned in a day
6. Number of times a user is replied to, by other users
7. Number of times a tweet is re-tweeted by other users **
8. Number of times a tweet is marked as "favourite" by other users.**

In this work, we have used the last two interaction features only for comparison study, while other features are part of an extension work primarily focusing on iteration specific models in identifying life events.

5. EXPERIMENTAL RESULT

In this step, we analyse the experimental steps and present the results of classifications. We started with the ground-truth annotation process followed by classification steps and their results.

5.1 Ground Truth Annotation

In the absence of any benchmark data for personal event detection prepared a gold standard dataset with manual annotation of 2 users with computing background . Annotators were given 1000 tweets per event for annotation. These 1000 tweets are randomly selected from the filtered dataset. Instruction for annotation was to annotate a tweet as event positive (presence of event) if they consider the tweet describes an event happening (present e.g. today) or about to happen with certainty (e.g. 4 days to graduation) within a month's time window. It is difficult to precisely define an event as most of the tweets are not reported exactly during the event but pre and post event. Since our objective is to identify the event from userÕs timeline with definitive time stamp attached to the event, we opted for a 1 month time interval. We retained those tweets (304) as event positive tweets whenever both the annotators agreed on the label. It is imperative to mention that event negative tweets are simply those where annotators felt that a particular event is not occurring despite the presence of event related keyword.

5.2 Event Detection: Unigram Model(UNI)

Our first model is the simplest bag-of-word model where word frequencies are used as features for document classification. In our case, each tweet is considered 1 document. We first applied a String to word vector filter that coverts the strings into numerical features. Then we trained our model with 10-fold cross validation using four different types of classifiers: Naive Bayes (NB), Multinomial Naive Bayes (MNB), Support Vector Machine (SVM) and Decision Tree

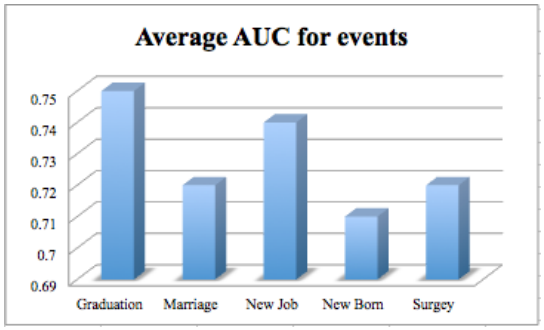


Figure 1: AUC curve for different events.

(J48) implemented in machine learning library Weka [**?**]. We evaluated our model on the test set (100 from each event) and performance of these classifiers reported in terms of Recall (is the number of correct results divided by the number of results that should have been returned) Precision (is the number of correct results divided by the number of all returned results) and F-score (harmonic mean). Table 3 (fig. 2) shows the average precision, recall and F score for all the events. However SVM performed best in 4 out of 5 followed by Naive Bayes. Graduation (.8) has highest precision score whereas "New job" has the highest recall (.95) score. The most difficult event is the "New born" across all the classifiers with lowest precision score (.55).

Examining the ROC curves which plots the true positives (TP) vs false positives(FP) and indicates the area under curve (figure 1) (AUC: probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative example) ranges from .71 to .75 giving a reasonable quality of the learners. NB performs better than SVM with an average of .77 against .72 across all events.

Table 3: Average precision, recall and f-Measure from all classifiers based on unigram model.

Event	Precision	Recall	F-Measure
Graduation	0.80	0.80	0.73
Marriage	0.75	0.87	0.79
New Job	0.78	0.95	0.80
New Born	0.55	0.92	0.68
Surgery	0.72	0.87	0.76

Analysis of error classification mainly showed the diversity of language constructs among the misclassified tweets. Since the model is purely content based, any variation not captured by the model are missed from the result.

5.3 Event Detection: Model with Contextual Lexical Patterns (UNI+META)

Bag-of-words or unigram model is the basic approach yet proved to have reasonable accuracy though with lots of false positives. This led us to refine the model with more lexical features and features such as sentiment. We considered features (described in sec. 4) such as co-occurring terms (e.g. prayers, hospital for surgery), POS tagging, presence of social relation terms(my friend, sister etc.), temporal terms (today, week, morning etc.), sentiment strength of a tweet. POS tagging was done using Stanford tagger[6] and sentiment was derived using the Sentistrength java library[**?**].

Recognizing Temporal Expression: Temporal features tend to be implicit, diverse, and informal (e.g. last week, hourly, around the corner). Identifying these references within the vicinity of an event term occurrence increases the likelihood of accurate detection. Moreover, we need to resolve the tense of the verb as well to know weather the tweet is about some future event, or past. In this paper, we are using the time terms of LIWC dictionary which has 68 time inducing terms (e.g. forever,week,until etc.). This feature also used as a binary feature in the second classification model.

Average accuracy of the second model showed an average improvement of 4-5 % in precision score over the initial model for all the events, showing that simple lexical features are able to capture some of the diversity. For brevity purpose we are only showing the results of the top classifier (SVM).

Table 4: Precision, Recall and F-measure for (UNI+META) Model (SVM).

Event	Precision	Recall	F-Measure
Graduation	0.83	0.81	0.819
Marriage	0.77	0.83	0.798
New Job	0.818	0.93	0.865
New Born	0.61	0.92	0.733
Surgery	0.77	0.87	0.816

5.4 Event Detection: Model with Interaction Features (UNI+META+INT)

Inherent in social media and social networks, it is intuitive to hypothesise that interesting events will stimulate interesting and increased interaction among the friend circle of the user in the form of replies and sharing. The third and the final model takes advantage of these interaction features embedded in microblogging sites through mechanisms like retweet and favourites. Each tweet is now represented with two more features besides the above lexical features for classification. We used only SVM as the classifier because of its superior performance in previous two occasions. Results of the final model (table 5) are reported by means of precision score per event. A final comparison of four models (UNI, UNI+META, UNI+META+INT and INT) is shown in figure 3. The result shows that, although the hybrid model performed better than the unigram-based one (UNI), the improvement was marginal. On the other hand, the model based only on interaction features (INT) performed worst, where accuracy dropped to 53-61%. .

6. CONCLUSION

This paper describes event detection from personal timeline of a user in Twitter. Existing detection tasks predominantly focused on public events and events concerning celebrities both from news articles and social media whereas personal life events are mostly overlooked. We started with 5 life

[6]http://nlp.stanford.edu/software/tagger.shtml

Table 5: Precision, Recall and F-measure for (UNI+META+INT) Model (SVM).

Event	Precision	Recall	F-Measure
Graduation	0.85	0.83	0.839
Marriage	0.79	0.83	0.809
New Job	0.82	0.91	0.862
New Born	0.64	0.92	0.754
Surgery	0.78	0.87	0.822

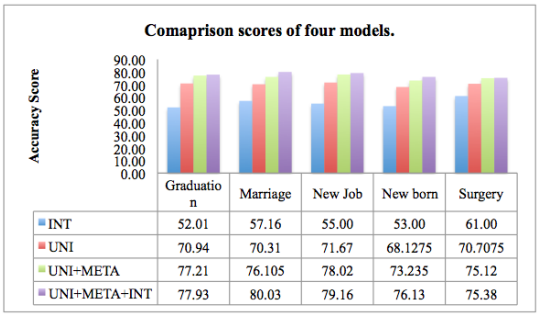


Figure 2: A comparative performance of four different models.

events and trained 5 different binary classifiers based on bag-of-word features which gave 55 to 80% precision on a test dataset with an average AUC of 77%. The learning models were further streamlined with meta features such as sentiment, temporal, social relation terms, emoticons and punctuations features, which improved the classification performance by 4-5%, however addition of interaction feature in the third classifier did not yield substantial improvement contrary to the expectation. This final result is a stronger motivation for an in-depth analysis of these features in our future work. We also aimed to adopt an unsupervised approach to detect life events as there may be many more unexpected events happening in one's life bearing substantial influence in life and eligible to be included .

7. ACKNOWLEDGMENT

This work was supported by EPSRC project ReelLives (EP/L004062/1).

8. REFERENCES

[1] P. Agarwal, R. Vaithiyanathan, S. Sharma, and G. Shroff. Catching the Long-Tail : Extracting Local News Events from Twitter. In *book1*, pages 379–382, 2012.

[2] E. Benson, A. Haghighi, and R. Barzilay. Event Discovery in Social Media Feeds. In *book1*, volume 3, pages 389–398. Association for Computational Linguistics, 2011.

[3] L. Chen and A. Roy. Event detection from flickr data through wavelet-based spatial analysis. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 523–532, New York, NY, USA, 2009. ACM.

[4] B. D. Eugenio, N. Green, and R. Subba. Detecting Life Events in Feeds from Twitter. pages 274–277. Ieee, 2013.

[5] C. S. Firan, M. Georgescu, W. Nejdl, and R. Paiu. Bringing order to your photos: Event-driven classification of flickr images based on social knowledge. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 189–198, New York, NY, USA, 2010. ACM.

[6] J. Gl§ck and S. Bluck. *Looking back across the life span: A life story account of the reminiscence bump*. Springer, 2007.

[7] Q. He, K. Chang, and E.-P. Lim. Analyzing feature trajectories for event detection. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 207–214, New York, NY, USA, 2007. ACM.

[8] A. Jackoway, H. Samet, and J. Sankaranarayanan. Identification of live news events using Twitter. In *book1*, page 1, New York, New York, USA, 2011. ACM Press.

[9] A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: Understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis*, WebKDD/SNA-KDD '07, pages 56–65, New York, NY, USA, 2007. ACM.

[10] S. Papadopoulos, C. Zigkolis, Y. Kompatsiaris, and A. Vakali. Cluster-based landmark and event detection for tagged photo collections. In *book1*, volume 18, pages 52–63, Los Alamitos, CA, USA, Jan. 2011. IEEE Computer Society Press.

[11] S. Phuvipadawat and T. Murata. Breaking news detection and tracking in twitter. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 03*, WI-IAT '10, pages 120–123, Washington, DC, USA, 2010. IEEE Computer Society.

[12] T. Sakaki. Earthquake shakes twitter users : Real-time event detection by social sensors. In *Proceedings of the 19th International Conference on World Wide Web*, 2009.

[13] M. Thelwall, K. Buckley, G. Paltoglou, and D. Cai. Sentiment strength detection in short informal text, 2010.

[14] C. L. Wayne. Topic detection tracking (tdt). In *In Proceedings DARPA Broadcast News Transcription and Understanding Workshop*, page 98, 1998.

[15] J. Weng, Y. Yao, E. Leonardi, F. Lee, and B.-s. Lee. Event detection in twitter. In *book1*, pages 401–408. Ieee, 2011.

Improving Sparsity Problem in Group Recommendation

Sarik Ghazarian[‡], Nafiseh Shabib[‡†], Mohammad Ali Nematbakhsh[‡]
[‡]University of Isfahan, [‡†]Norwegian University of Science and Technology
sarikghazarian@yahoo.com,
shabib@idi.ntnu.no,
mnematbakhsh@eng.ui.ir

ABSTRACT

Group recommendation systems can be very challenging when the datasets are sparse and there are not many available ratings for items. In this paper, by enhancing basic memory-based techniques we resolve the data sparsity problem for users in the group. The results have shown that by conducting our techniques for the users in the group we have a higher group satisfaction and lower group dissatisfaction.

Keywords

sparsity, group recommendation, collaborative filtering

1. INTRODUCTION

Recommendation systems (RSs) are tools and techniques, which provide suggestions for items to be used by users. They generally directed towards helping users for finding items that are likely interested in the overwhelming number of items and they try to predict the most suitable products or services, based on the users' preferences and constraints. However, even active users have rated just a few items of the total number of available items in a database and respectively, even popular items have been rated by only a few number of total available users in the database. This problem, commonly referred as a sparsity problem [17]. Different approaches have been proposed in the research literature focusing on Sparsity problem for single user recommendations [21, 24]. However, as far as we know, this is the first work presenting a complete model for group recommendations, which resolving sparsity problem for a group. In general, sparsity has a major negative impact on the effectiveness of a collaborative filtering approach and especially on group recommendation. The main challenge behind group scenarios has been that of computing recommendations from a potentially diverse set of group members' ratings in a sparse situations. In this work, we studied sparsity problem in the group recommendation. First, we formalize the problem of *sparsity* in the group recommendation and use our model for aggregating user rating in a group. Second, we run an extensive set of experiments with different group sizes and different group cohesiveness on Millions of Song data set. Our experiments exhibit that in the most cases the group satisfaction in our proposed model is higher and the group dissatisfaction is lower than the previous models, which does not take into account sparsity.

The rest of paper is organized as follows: Section 2 describes the sparsity problem for a group and we propose a complete model for sparsity in the group recommendation. Experiments are presented in section 3. Section 4 provides some background and formalism. We conclude in section 5.

2. DATA MODEL AND RECOMMENDATION ALGORITHM

We assume a set of users $\mathcal{U} = \{u_1, \ldots, u_n\}$ out of which any ad-hoc group $\mathcal{G} \subseteq \mathcal{U}$ can be built. We consider a set $\mathcal{I} = \{i_1, i_2, \ldots, i_m\}$ with m items.

2.1 Item-Item Similarity

The basic component of proposed method is a machine learning regression method called Support Vector Machine (SVM) which is used for calculating similarities between items [26]. SVM is a supervised learning technique, which learns the function that is produced from input data in the best manner. It uses the built-in function to give appropriate output for an input data [26]. The input data pairs are as follows: $(x_1, y_1), ..., (x_i, y_i)$. The x_i is a record in d dimensional space and y_i is a real value. SVM tries to find $f(x)$ function which approximates the relations between data points [20]. The target function has two types: *linear* and *nonlinear*. In linear regression the relationships between input and output data points are linear and their relationships can be approximated by a straight line. The linear function is computed as equation 1.

$$f(x) = w.x + b \tag{1}$$

, where $w \in X$, X is the input space and b is a real value [20].

In nonlinear case SVM preprocesses input data. It uses nonlinear mapping function ($\varphi \rightarrow \rho$) which maps data from input space to the new feature space ρ. After this mapping action, the standard linear SVM regression algorithm is applied in the new higher feature space. The dot product between data points in higher dimensional feature is called kernel function [23]. Equation 2 shows this function.

$$K(x, x') = \varphi(x).\varphi(x') \tag{2}$$

There are different kernel functions like linear, polynomial, radial basis function (RBF), and Pearson VII Universal Kernel (PUK) [23]. In our proposed method PUK function has been used for modeling the similarities between items, because it had higher accuracy than other functions.

$$PUK : k(x, x') = \frac{1}{\left[1 + \left(\frac{\sqrt[2]{\|x-x'\|^2}\ \sqrt[2]{2^{(\frac{1}{\omega})}-1}}{\sigma}\right)^2\right]^{\omega}} \tag{3}$$

2.2 Listen Count

The algorithms in our work are based on explicit feedback from users; subsequently there is a need to normalize the listening counts to a predefined scale so that the algorithms can work optimally. In the [11], they modified basic latent factor model to convert implicit ratings to the explicit ones. Similarly to the approach taken [11], a boolean variable (p_{ui}) shows the user's interest on an item (equation 4). If a user has listened to a song (l_{ui}), its boolean variable's value is 1 otherwise it is 0. Thus, implicit data do not indicate users' preferences, rather they show confidence (c_{ui}) about users' preferences and there is a direct relationship between confidence value and the number of times that each user has listened to a song (equation 5). The relationship is controlled by constant α.

$$p_{ui} = \begin{cases} 1 & \text{if } l_{ui} > 0 \\ 0 & \text{if } l_{ui} = 0 \end{cases} \tag{4}$$

$$c_{ui} = 1 + \alpha l_{ui} \tag{5}$$

By these alternations, the equation of latent factor model modified as equation 6. This equation is a least square optimization process by considering user factors (p_u) or item factors (q_i) to be fix in each step. After finding user factors and item factors, their dot products show the users' explicit ratings on items.

$$\min_{q^*,p^*} = \sum_{r_{ui} \text{is known}} c_{ui}(r_{ui} - q_i^T p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2) \tag{6}$$

2.3 Sparsity Calculation

The sparsity value was computed as follows:
The ratio of specified ratings of items in the initial user-item matrix to the whole specified and not specified items' ratings.

$$SparsityValue = \frac{\text{Num.of specified ratings}}{\text{Num.of all possible ratings}} \tag{7}$$

2.4 Group Modeling

We define the following hypothesis: *The relevance between a group and an item i is only dependent on the relevance of i to individual members of the group.* Using this hypothesis, we derive the following definition that not only includes the preferences of individual users but also integrates the users preferences when they are in a group while recommending a set of items.

2.4.1 User-User Similarity

The major goal of this component is to overcome the weakness of Pearson's correlation method in the sparsity situation. The Pearson's correlation is limited to the joint items in both users' preference lists. In a random group setting, the collections of common items between users are very small, so comparing users based on very few items leads to lower accuracy [8, 19]. To solving this problem, the idea of proposed method is to compare all items rated by one user with all items in another user in the group, one by one. In other words, our method involves all possible combination of items in preference lists of both users. Equation 8 demonstrates the idea. The basic part of this equation is based on our conception of similar and dissimilar users:
Two users are considered similar, if they have close ratings for similar items.
Two users are dissimilar, if they have rated two dissimilar items.
Given a group $\mathcal{G}$, the similarity of each user $u \in \mathcal{G}$ is denoted as:

$$UserSim_{uv} = \frac{\Sigma_{\forall i \in R_u \bigcap \forall j \in R_v}(1 - \frac{|r_{ui} - r_{vj}|}{r_{max} - r_{min}}) \times ItemSim_{ij}}{\Sigma_{\forall i \in R_u \bigcap \forall j \in R_v}|ItemSim_{ij}|} \tag{8}$$

, $R_u = \{i | r_{ui} \neq 0\}, R_v = \{j | r_{vj} \neq 0\}$, and r_{max} and r_{min} are maximum and minimum possible values of the ratings. Note that, $ItemSim_{ij}$ is equal to similarity values between items i and j which is calculated by the SVM regression model that has been explained in the 2.1.

2.4.2 User-Item Relevance

Given a group $\mathcal{G}$, the relevance of a user $u \in \mathcal{G}$ for an item $i \in \mathcal{I}$ is denoted as:

$$Rel_{ui} = \bar{r}_u + \frac{\Sigma_{v \in U}(r_{vi'} - \bar{r}_v) \times UserSim_{uv}}{\Sigma_{v \in U}|UserSim_{uv}|} \tag{9}$$

, where i' is the most similar item to i that user v has rated. Thus, by considering i' in the relevance function, it is not required to take into account just the users who have rated the same item, but it considers all ratings given by users, and we can use ratings of other most similar items to the target item to fill in the sparseness.

2.4.3 Group Relevance

The preference of an item i by a group $\mathcal{G}$, denoted as $Grel(\mathcal{G}, i)$, is an aggregation over the preferences of each group member for that item. We consider two main aggregation strategies:
Average

$$Grel(\mathcal{G}, i) = \frac{\Sigma_{u \in \mathcal{G}} Rel_{ui}}{|\mathcal{G}|} \tag{10}$$

Least Misery

$$Grel(\mathcal{G}, i) = min_{u \in \mathcal{G}}(Rel_{ui}) \tag{11}$$

2.5 Group Satisfaction

To evaluate our methods accuracy in group recommendation process, we used group satisfaction metric [5].
This metric is the average of all group members' satisfaction for recommended items

$$Gsat = \frac{\Sigma_{u \in U} Usat}{|\mathcal{G}|} \tag{12}$$

User's satisfaction is shown as $Usat(u)$ which is calculated:

$$Usat = \frac{\Sigma_{i=1}^{k} Rel_{ui}}{k * Max(Rel_{ui})} \tag{13}$$

, where Rel_{ui} is user preference on item, k is the number of items, and $Max(Rel_{ui})$ is maximum preference value of user u for all items.

2.6 Group DisSatisfaction

To evaluate our methods in group recommendation process, we also used group dissatisfaction metric [13]. This metric is the fraction of dissatisfied users whose satisfaction measures were less than a threshold. In our case we consider

the threshold equals to 0.6.

$$GdisSat = \frac{|\mathcal{U}|}{|\mathcal{G}|} \quad (14)$$

, where $u|Usat < 0.6$ (equation 13)

3. EXPERIMENTS

We have shown after solving sparsity problem for each single user in the group, we have a higher group satisfaction and lower group dissatisfaction.

Dataset description: In this section, we evaluate our method with Million Song Dataset (MSD)[1], in the music recommendation scope. The Million Song Dataset (MSD) is a collection of music audio features and metadata that has created to support research into industrial-scale music information retrieval. It is freely-available collection of meta data for one million of contemporary songs such as song title, artist, publication year, audio features, and much more [14]. In addition, The MSD is a cluster of complementary datasets contributed by the community: SecondHandSongs dataset for cover songs, musiXmatch dataset for lyrics, Last.fm dataset[2] for song-level tags and similarity, and Taste Profile subset for user listening history data. Comprising several complementary datasets that are linked to the same set of songs, the MSD contains extensive meta-data, audio features, song-level tags, lyrics, cover songs, similar artists, and similar songs. In this work, we have used information about song's features such as *title*, *release*, *artist*, *duration*, *year*, *song-hotness*, *songs similarity*, *users listening history*, and *song's tags*. In addition to this information, we have information about song tags and its degrees in Last.fm dataset, which the tag's degree shows how much the song is associated to a particular tag. In our work, for each song we consider three main tags.

We implemented our prototype system using Java and for computing SVM model's accuracy we used WEKA[3].

3.1 Item-Item Similarity

In order to use similarity data between songs and create SVM regression model, we needed to prepare suitable data, *preprocessing*, for training process as follows: *song*, *release*, *artist*, *term1*, *term2*, *term3*, *song-hotness*, *duration*, *year*, *similarity-degree*.

In MSD, about half of songs have at least one tag. In this research for each song, its three most relevant tags were considered. If a song didn't have three relevant tags, remaining tags were filled with the highest one. Similarity-degree is an integer attribute in [0, 1] interval. 1 shows the most similar songs and conversely 0 is used for dissimilar songs. In SVM model each record should be represented as a point in input space. To achieve this purpose similarity based functions have been used [10]. For computing similarity between string attributes, *Jaro-Winkler* method has been used, which gives 1 to most similar items and 0 to dissimilar ones. For terms, we used similarity function of nominal attributes. After computing similarity between corresponding pairs of attributes, each record came in form: *title-dif*, *release-dif*, *artist-dif*, *term-dif*, *song-hotness-dif*, *duration-dif*, *year-dif*, *similarity-degree* The "dif" suffix stands for the differences. Then we used these new records to create SVM model for predicting similarities between songs.

3.1.1 Item-Item similarity results

For computing SVM model's accuracy, mean absolute error (MAE) [25] values of different regression models were compared by using Waikato Environment for Knowledge Analysis (WEKA) software tool. All parameters in different methods were tested. In all SVM methods with different kernel functions like PUK, RBF, $normalizedPolyKernel$, and $polyKernel$, the PUK kernel function with $\sigma = 1$ and $\omega = 1$ had the minimum and best MAE value. Figure 1 illustrates different MAE values for different regression methods in WEKA. Therefore, in our work PUK function has been used for modeling the similarities between items.

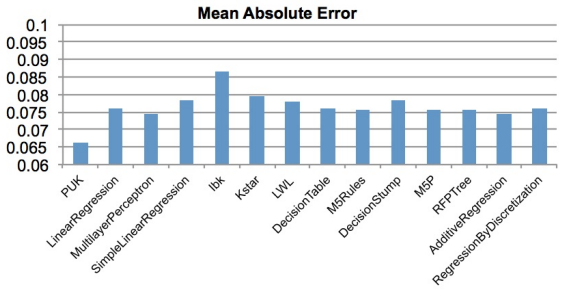


Figure 1: MAE value for different regression methods

3.2 User Collection Phase

We selected subset of users to provide their music preferences. Later, those users are used to form different groups and perform judgments on group recommendations. For this aim, we selected those users who have at least listened to fifteen songs in our dataset. As mentioned in previous section, MSD contains listening history of users, which shows the number of times each user has listened to a particular song. Thus, preferences have been expressed in implicit format. This format is not equivalent to explicit one, which shows the exact preferences of users. Since, the user-based and item-based collaborative filtering (CF) approaches have been designed for explicit ratings, conversion of implicit feedbacks to explicit ones was essential. In order to achieve explicit one, we have used latent factor model with some alternations as proposed in the Listen Count in the previous part.

3.3 Group Formation

We considered two main factors in forming user groups *i.e.* *group size*, *group cohesiveness* [2]. We hypothesized that varying group sizes will impact to the group satisfaction. We chose three group sizes, 3, 5, and 10, representing small, medium, and large groups, respectively. Similarly, we assumed that group cohesiveness (i.e., how similar are group members in their music tastes) is also a significant factor in their satisfaction with the group recommendation. As a result, we chose to form three kinds of groups: *similar*, *dissimilar*, and *random*.

3.4 Result Interpretation

After predicting unknown items' score in all users' preference lists, it is essential to aggregate users' preferences to

[1]http://labrosa.ee.columbia.edu/millionsong
[2]http://last.fm
[3]http://www.cs.waikato.ac.nz/ml/weka

make recommendation for a group. For this purpose, we used basic methods (average and least misery) and recommended k items with highest values. To evaluate our method in the group recommendation process, we used group satisfaction and dissatisfaction metrics. The reason that we used group dissatisfaction metric is observing how the algorithm performs when we have dissatisfied members in the group. Note that, the sparsity value for each group is the following numbers.
Similar group : G3=0.31 G5=0.55 G10=0.77
Dissimilar group : G3=0.52 G5=0.68 G10=0.80
Random group : G3=0.58 G5=0.72 G10=0.84

3.4.1 Varying Group size

We examined the effect of different group sizes on group satisfaction/dissatisfaction in Figure 2. The number of recommended items is fixed 10 and the group sizes varies between 3, 5, and 10 members. As we can see in Figure 2, in the similar groups, the group satisfaction remains the same even though the number of people in each group is increasing. In addition, in most of cases our algorithm has higher group satisfaction in both average and least misery methods in compare of CF method, which does not take into account sparsity. Additionally, with increasing the group sizes the sparsity value is increasing, but our algorithm performs fairly constant. Moreover, the result shows that in the dissimilar and random groups we have lower dissatisfaction.

3.4.2 Varying Top-k

We examined the effect of different recommendation items (Top-k= 5,10,15, and 20) on group satisfaction/dissatisfaction in Figure 3. The group size is fixed 10. The result shows that with increasing the number of items, the group satisfaction is decreasing in all the groups but it decreases more in the similar and dissimilar groups than random groups. In general, our method has a higher group satisfaction in compare of CF method. Also, the result shows that, we have less dissatisfaction when we applied Average as an aggregation method and we have less dissatisfaction in our method.

3.4.3 Varying Group Cohesiveness

We examined the effect of different group cohesiveness on group satisfaction/dissatisfaction in Figure 4. Group cohesiveness varies between similar group (similarity between members >0.5), dissimilar (similarity between members <0.5) and random members. The number of recommended item is fixed 10. Our observation showed that for small groups, group satisfaction is very close to each other in different techniques, but in the random groups we can see noticeably change in the group satisfaction between CF and our proposed method that takes into account sparsity. In addition, the result shows that in the dissimilar and random group our method has a lower dissatisfaction.

4. RELATED WORK

Research on recommendations is extensive. Typically, recommendation approaches are distinguished between: content-based, collaborative filtering, and hybrid [1]. Recently, there are also approaches focusing on group recommendations. Group recommendation aims to identify items that are suitable for the whole group instead of individual group members. Group recommendation has been designed for various domains such as news pages [18], tourism [9], music [6], and TV programs [27]. Group is defined as two or more individuals who are connected to one another. A group can range in size from two members to thousands of members. A group may be formed at any time by a random number of people with different interests, a number of persons who explicitly choose to be part of a group, or by computing similarities between users with respect to some similarity functions and then cluster similar users together [15, 2]. There are two dominant strategies for groups: (1) aggregation of individual preferences into a single recommendation list or (2) aggregation of individual recommendation lists to the group recommendation list [2, 3]. In other words, the first one creates a pseudo user for a group based on its group members and then makes recommendations based on the pseudo user, while the second strategy computes a recommendation list for each single user in the group and then combines the results into the group recommendation list.
However, in the both approaches we may faced the sparsity problem. Sparsity is one of the major problems in memory-based CF approaches [22]. In sparseness conditions most cells of user-item matrix are not rated. The reason is that users may not willing to provide their opinions and preferences and they do this only when it is necessary [7]. In these type of matrices, the accuracy of calculated predictions by applying memory-based CF approaches is low, since there are not enough information about user ratings [12]. Lately, Ntoutsi applied user-based CF approach in order to predict unknown ratings [16]. For this, they partitioned users in to clusters. Then for predicting a particular item's rating for a user, they considered just the ones in the cluster of target user instead of all users in dataset. They calculated the relevancy of an item to a user based on the relevancy of that item to similar users in the target user's cluster. Moreover, they involved a support score in prediction process to be shown how many users in the cluster have rated that item. Because of using memory-based approaches as basis, this approach also cannot be used in sparse data situations. Chen et al. proposed a method which predicts each item's group rating by considering its similar items that have been rated by whole group or by most subgroups [4]. For this aim, first they applied collaborative filtering technique and find each user's preferences on that item and then used genetic algorithm according to subgroups' ratings to achieve the item's overall score. However, our main focus in this research is on sparsity problem in users' preference lists, Chen et al. worked on sparsity problem in groups' ratings, for this reason they could use collaborative filtering in their calculations.

5. CONCLUSION

We formalize the problem of sparsity in the group recommendation and use our model for aggregating user rating for the group. In this work, we proposed a new method that overcomes the weakness of basic memory-based approaches in sparsity. We evaluated our method in sparse cases and compared it with prior methods. The results show that in sparse matrices our proposed method has better group satisfaction and lower group dissatisfaction than basic CF. In addition, in conditions where user-based approach can be run, our proposed method performs better. In the future, we plan to peruse the accuracy of our proposed method in other less been paid fields like TV programs, books and images, and we want to investigate our research in the big

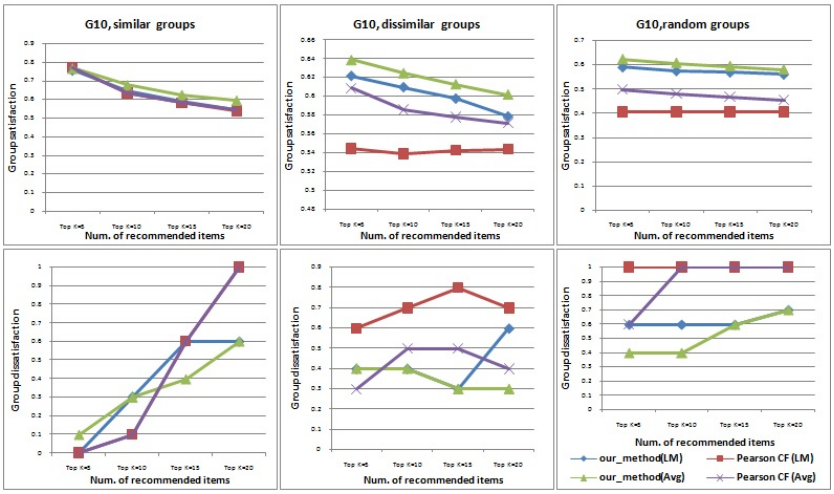


Figure 2: Comparison of group satisfaction and group dissatisfaction with varying group size

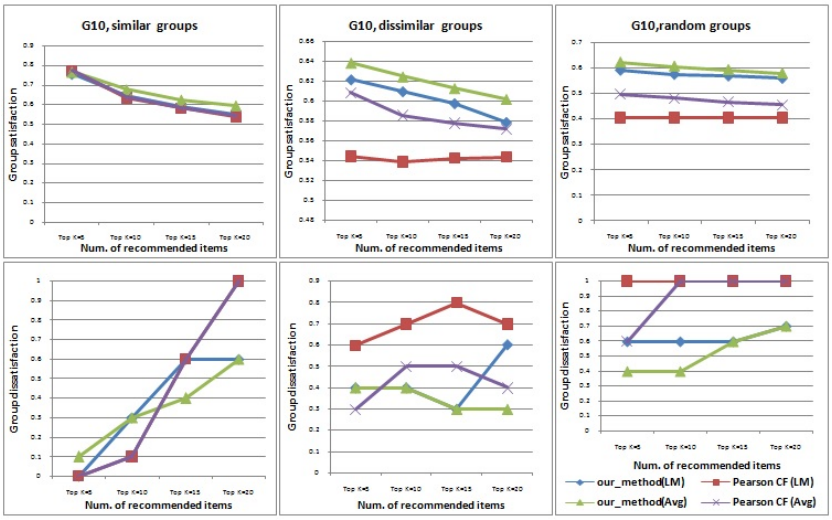


Figure 3: Comparison of group satisfaction and group dissatisfaction with varying Top-k

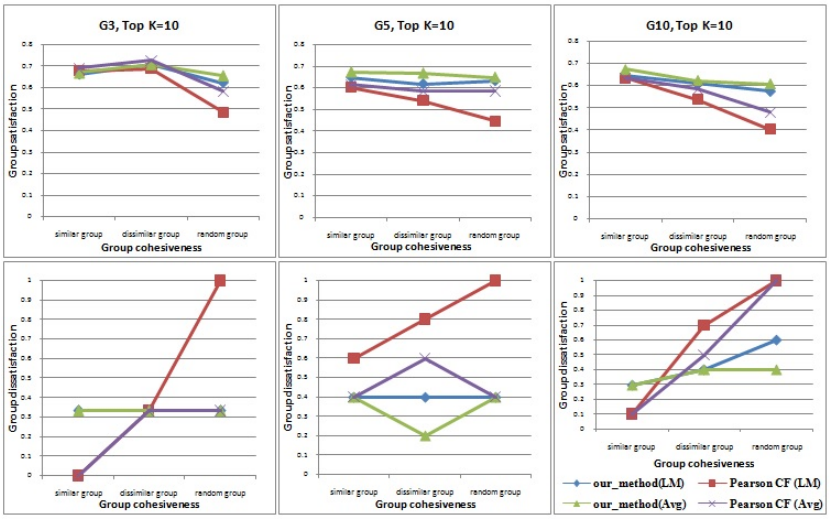


Figure 4: Comparison of group satisfaction and group dissatisfaction with varying group cohesiveness

groups.

6. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[2] S. Amer-Yahia, S. B. Roy, A. Chawlat, G. Das, and C. Yu. Group recommendation: semantics and efficiency. *Proc. VLDB Endow.*, pages 754–765, 2009.

[3] S. Berkovsky and J. Freyne. Group-based recipe recommendations: Analysis of data aggregation strategies. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, pages 111–118, 2010.

[4] Y.-L. Chen, L.-C. Cheng, and C.-N. Chuang. A group recommendation system with consideration of interactions among group members. *Expert Syst. Appl.*, 34(3):2082–2090, 2008.

[5] I. A. Christensen and S. N. Schiaffino. Entertainment recommender systems for group of users. *Expert Syst. Appl.*, 38(11):14127–14135, 2011.

[6] A. Crossen, J. Budzik, and K. J. Hammond. Flytrap: Intelligent group music recommendation. In *Proceedings of the 7th International Conference on Intelligent User Interfaces*, IUI '02, pages 184–185, New York, NY, USA, 2002. ACM.

[7] L. N. Dery. Iterative voting under uncertainty for group recommender systems (research abstract). In J. Hoffmann and B. Selman, editors, *AAAI*. AAAI Press, 2012.

[8] A. Eckhardt. Similarity of users' (content-based) preference models for collaborative filtering in few ratings scenario. *Expert Syst. Appl.*, 39(14):11511–11516, Oct. 2012.

[9] I. Garcia, L. Sebastia, and E. Onaindia. On the design of individual and group recommender systems for tourism. *Expert Syst. Appl.*, 38(6):7683–7692, June 2011.

[10] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2011.

[11] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, pages 263–272, Washington, DC, USA, 2008. IEEE.

[12] Z. Huang, H. Chen, and D. D. Zeng. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Trans. Inf. Syst.*, 22(1):116–142, 2004.

[13] J. K. Kim, H. K. Kim, H. Y. Oh, and Y. U. Ryu. A group recommendation system for online communities. *Int. J. Inf. Manag.*, 30(3):212–219, June 2010.

[14] B. McFee, T. Bertin-Mahieux, D. P. Ellis, and G. R. Lanckriet. The million song dataset challenge. In *Proceedings of the 21st International Conference Companion on World Wide Web*, WWW '12 Companion, pages 909–916. ACM, 2012.

[15] E. Ntoutsi, K. Stefanidis, K. Nørvåg, and H.-P. Kriegel. Fast group recommendations by applying user clustering. In *Proceedings of the 31st International Conference on Conceptual Modeling*, pages 126–140. Springer-Verlag, 2012.

[16] E. Ntoutsi, K. Stefanidis, K. Nørvåg, and H.-P. Kriegel. Fast group recommendations by applying user clustering. In *ER*, pages 126–140, 2012.

[17] M. Papagelis, D. Plexousakis, and T. Kutsuras. Alleviating the sparsity problem of collaborative filtering using trust inferences. In *Proceedings of the Third International Conference on Trust Management*, iTrust'05, pages 224–239, Berlin, Heidelberg, 2005. Springer-Verlag.

[18] S. Pizzutilo, B. De Carolis, G. Cozzolongo, and F. Ambruoso. Group modeling in a public space: Methods, techniques, experiences. In *Proceedings of the 5th WSEAS International Conference on Applied Informatics and Communications*, AIC'05, pages 175–180, Stevens Point, Wisconsin, USA, 2005. World Scientific and Engineering Academy and Society (WSEAS).

[19] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. T. Riedl. Application of dimensionality reduction in recommender systems: A case study. In *WebKDD Workshop at the ACM SIGKKD*, 2000.

[20] A. J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, Aug. 2004.

[21] X. Su, T. M. Khoshgoftaar, X. Zhu, and R. Greiner. Imputation-boosted collaborative filtering using machine learning classifiers. In *Proceedings of the 2008 ACM Symposium on Applied Computing*, SAC '08, pages 949–950, New York, NY, USA, 2008. ACM.

[22] X. Su, T. M. Khoshgoftaar, X. Zhu, and R. Greiner. Imputation-boosted collaborative filtering using machine learning classifiers. In *SAC*, pages 949–950, 2008.

[23] B. Ustun, W. Melssen, and L. Buydens. Facilitating the application of support vector regression by using a universal pearson vii function based kernel. 2006.

[24] J. Wang, A. P. de Vries, and M. J. T. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In E. N. Efthimiadis, S. T. Dumais, D. Hawking, and K. Järvelin, editors, *SIGIR*. ACM, 2006.

[25] G.-R. Xue, C. Lin, Q. Yang, W. Xi, H.-J. Zeng, Y. Yu, and Z. Chen. Scalable collaborative filtering using cluster-based smoothing. In R. A. Baeza-Yates, N. Ziviani, G. Marchionini, A. Moffat, and J. Tait, editors, *SIGIR*, pages 114–121. ACM, 2005.

[26] H. Yu and S. Kim. Svm tutorial — classification, regression and ranking. In G. Rozenberg, T. Bäck, and J. Kok, editors, *Handbook of Natural Computing*, pages 479–506. Springer Berlin Heidelberg, 2012.

[27] Z. Yu, X. Zhou, Y. Hao, and J. Gu. Tv program recommendation for multiple viewers based on user profile merging. *User Model. User-Adapt. Interact.*, 16(1):63–82, 2006.

Towards Personalized Offers by Means of Life Event Detection on Social Media and Entity Matching

Paulo Cavalin
IBM Research - Brazil
pcavalin@br.ibm.com

Maíra Gatti
IBM Research - Brazil
mairacg@br.ibm.com

Claudio Pinhanez
IBM Research - Brazil
csantosp@br.ibm.com

ABSTRACT

In this paper we present a system for personalized offers based on two main components: a) a hybrid method, combining rules and machine learning, to find users that post life events on social media networks; and b) an entity matching algorithm to find out possible relation between the detected social media users and current clients. The main assumption is that, if one can detect the life events of these users, a personalized offer can be made to them even before they look for a product or service. This proposed solution was implemented on the IBM InfoSphere BigInsights platform to take advantage of the MapReduce programming framework for large scale capability, and was tested on a dataset containing 9 million posts from Twitter. In this set, 42K life event posts sent by 19K different users were detected, with an overall accuracy of 89% e precision of about 65% to detect life events. The entity matching of these 19K social media users against an internal database of 1.6M users returned 983 users, with accuracy of about 90%.

Keywords

Social Media Networks, Life Event Detection, Natural Language Processing, Machine Learning, Entity Matching

1. INTRODUCTION

Social Media Networks (SMN), such as Twitter and Facebook, engage thousands of people that post, on a daily basis, a huge amount of content represented by texts, images, videos, etc [5, 10]. Often the content can be intimately related to the person the publishes it, in such a way that is can expose behavioral traits or events that are happening in the individual's life. As a consequence, the proper exploration of this type of content not only can be a way to better understand the users on SMNs, but also can leverage many applications that require adequate user profiling, for instance credit risk analysis, marketing campaigns, and personalized product and/or service offers.

One way to find potential customers for services or products is by detecting life events from public user activities on SMNs, in special microbloggings. Generally, a life event can be defined as something important that happened, is happening, or will be happening, in a particular individual's life, such as getting married, get graduated, having a baby, buying a house, and thus forth. That is, if a life event is properly detected, a product or service can be offered to someone even before she looks for it, anticipating her needs. For instance, if a person posts on the SMN that her marriage will be happening in a few days (or weeks or months), a loan or an insurance (for the honey moon trip for example) can be offered to her in advance. Furthermore, as state in [6], marketers know that people mostly shop based on habits, but that among the most likely times to break those habits is when a major life event happens.

For this reason, this work focuses on presenting a system that can detect life events from textual posts on SMNs, and can match the corresponding users with an existing database, i.e. entity matching with current clients, using basic information such as the name and the location available on the SMN. Entity matching is important to understand whether a given user of a SMN is already a customer or not, and adapt the way the person can be approached.

Both life event detection and entity matching are complex tasks which are subject of various research in fields such as artificial intelligence, machine learning [6], natural language processing and large scale analysis of unstructured data (popularly known as *Big Data*) [12]. Performing natural language processing on microbloggings' posts presents several challenges, such as dealing with the short and asynchronous nature of the messages, making it difficult to extract contextual information, and dealing with a very unnormalized vocabulary due to the frequent use of slangs, acronyms, abbreviations, and informal language often with misspelling errors [1, 7, 13]. Nonetheless, one study that supports the possibility of detecting life events from textual posts has been presented in [4]. In that work, the author conducted a study on the behavior of mothers during pregnancy, and they observed that these mothers can be distinguished by linguistic changes captured by shifts in a relatively small number of words in their social media posts.

In the light of this, in this work we describe and evaluate our proposed solution to tackle the life event detection problem and the entity matching. For the first task, we propose a

hybrid system combining rules and machine learning (ML). In contrast to the system specifically focused on life event detection presented in [6] (the only one for this problem to the best of our knowledge), which uses only ML, our system allows for dealing with the life event classes independently. The rule-based phase acts as a mechanism to filter most posts that do not contain life events, since all those posts not matching the desirable rules are eliminated. Then, binary classifiers (one for each type of life event) are applied to validate the possible life events. Greater detail is provided in Section 3.1. For entity matching, a combination of string distance functions is used to compare the names and locations of the users. This method is better described in Section 3.2.

The entire system has been implemented on the IBM InfoSphere BigInsights platform [9], to take advantage of the MapReduce programming paradigm for large scale data processing. A dataset containing 9 million posts in portuguese, extracted from Twitter, has been used to evaluated the system. To evaluate the entity matching, a database with 1.6 million users has been constructed. More details about the experiments are present in Section 4.

2. BACKGROUND AND RELATED WORK

Since the work proposed in this paper is a hybrid solution on which we integrate a ML-based classifier with an Entity Matching solution, the background and related work is presented separated for both as follows:

Life Event Detection: as already mentioned, a life event can be defined as something important regarding the users' lives in SMNs. It is important to differentiate it from some related work which uses the *event detection* expression to refer to the problem of detecting unexpected event exposed by several users in SMNs like a rumor, a trend, or emergent topic. In the case of the work proposed in this paper, detection means to classify a short post, like Twitter's or Facebook's status messages in one of the life event categories, which could be considered, for instance, topics. Therefore, as related work, any approach of topic classification of short messages could be considered like [6], which is the most related to our work. Regarding ML-based solutions, other supervised or unsupervised methods for topic classification are also related, although not yet used for short messages but long documents. And regarding semantic-rule-based solutions, AQL rules combined with dictionaries are known approaches for topic classification with the usage of templates. Ontologies have also been applied for long documents.

Entity Matching: in SMNs there are two problems one can find Entity Matching solutions for. One is, given a set containing user features on SMNs, like user information and activities, and another set containing real people information, the goal is to try to match the users within both sets. The second problem is, given two sets containing user features on two different SMNs, the goal is to try finding corresponding users, i.e., the biggest possible number of social profiles that refer to the same person between both social networks. The latter can also be called Entity Resolution (ER) problem, and in the past few years some work has been proposed to solve this problem. For instance, [14] proposed supervised learning techniques and extracted features to build different classifiers, which were then trained and used to rank the probability that two user profiles from two different OSNs belong to the same individual.

The former problem can be considered a subset of the latter if we ignore the fact that the second set contains real people information rather than SMN's profiles. And generally, as summarized by [15], there are two approaches for handling this: (i) syntactic-based similarity approaches: providing exact or approximate lexicographical matching of two values; and (ii) semantic-based similarity approaches: used to measure how two values, lexicographically different, are semantically similar. For instance, Foaf-o-matic[1] and OKKAM[2] projects aim at social profiles integration by means of formal FOAF (Friend-of-a-friend) semantics.

Regarding, syntactic-based similarity approach, we summarize here the ones most used for URI, numeric-based attributes and, in the context of SNMs, two users' full names. *Levenshtein* or *Edit Distance* [11] is defined to be the smallest number of edit operations, inserts, deletes, and substitutions required to change one string into another. In addition, Jaro is an algorithm commonly used for name matching in data linkage systems. A similarity measure is calculated using the number of common characters (i.e., same characters that are within half the length of the longer string) and the number of transpositions. Winkler (or Jaro-Winkler) improves upon Jaro's algorithm by applying ideas based on empirical studies which found that fewer errors typically occur at the beginning of names [3][2].

Another approach is the N-Gram name similarity, on which N-grams are sub-strings of length n and an n-gram similarity between two strings is calculated by counting the number of n-grams in common (i.e., n-grams contained in both strings) and dividing by either the number of n-grams in the shorter string (called Overlap coefficient), or the number of n-grams in the longer string (called Jaccard similarity), or the average number of n-grams in both strings. 2-grams and 3-grams have been used to calculate the similarity between the two users' full names. Finally, the VMN name similarity approach proposed by [18] was designed for full and partial matches of names consisting of one or more words. VMN supports the case of swapped names and the cases of partial matches.

In this paper, we use two versions of ED preceded by Jaro's similarity as described in the next section.

3. METHODOLOGY

In this section we describe in detail both systems for life event detection system and entity matching.

3.1 Hybrid Life Event Detection System

Given a social media network, the life event detection system has as main goal to return a list of users that posted life events within a given time window. This task involves a crawler to gather data, and a system to search for life events on the data. Note that not only accuracy is important in this case, to find the largest list of users with a high precision, but also performance is important since the system is likely

[1]http://www.foaf-o-matic.org/
[2]http://www.okkam.org/

to face a large amount of data. In addition, on a production environment, the system must allow for easy fine-tuning, addition and removal of life events classes.

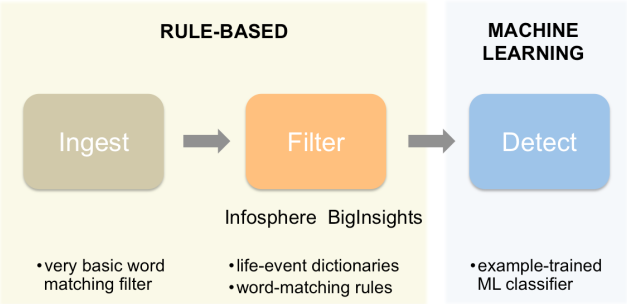


Figure 1: Hybrid Life Event Detection System.

To cope with the aforementioned issues, we propose a hybrid life event detection approach, combining both rules and machine learning (ML). Such a system, depicted in Figure 1, is basically composed of three subsequent phases or modules, namely *Ingest*, *Filter*, and *Detect*. The first phase, i.e. *Ingest*, captures a database of posts to be used for the search for life events. This is done by considering a set of words that can possibly relate to all life events of the system. We assume that the larger this dataset, the larger the set of users that will be returned. Once the set of posts has been totally crawled, the *Filter* module selects the set of posts that is more likely to contain life events. That is, by considering a set of simple rules such as words and combinations of words (but more elaborated rules than those of *Ingest*), but in this case a set of rules for each type of life event, the posts that match these rules are marked with the corresponding possible life events.

Despite these rules can indicate a possible life event, a large portion of these message can be false candidates. For this reason, the *Detected* phase is then carried out to validate the possible life events with their corresponding probability. For each post found in the *Filter* phase, we apply the ML classifier of the corresponding possible life events and compute the probability of that the post contains the given life events. With this information, all posts with life event probability above the threshold θ are selected and users of the corresponding posts are generate as the output of the system.

It is worth noting that currently ML is well-known to produce the best solutions to deal with ambiguous and noisy texts such as microbloggings' posts. However, the proposed hybrid solution takes advantage of the rule-based filtering to reduces the search space for the ML classifier, which can reduce both the number of errors and processing time. Moreover, by treating types of life events independently it makes it easy for fine-tuning, addition and removal of life event classes. For instance, to add a new type of life events, one need to append the corresponding keywords for the *Ingestion* phase, the rules for *Filter*, and a binary classifier in the *Detect* phase. This can be done with no impact on the accuracy of existing life events.

3.2 Entity Matching System

Given the output of the life event detection system, i.e. users (aka entities) that posted life events on social media, the main goal of the entity matching system is to find corresponding people in a database of real names. For achieving this task accurately, the system must use as much information as possible to decrease the level of uncertainty.

Dealing with users found on SMNs, though, is very challenging. First of all, on most SMNs the basic information about the user (e.g. name, location, age) is very limited (on Twitter only the name and location of the user are available). In addition, such personal information may be lacking or not relevant since filling them may be not mandatory, and the content filed is not verified. Besides that, when the information is seriously provided by the user, other difficulty factors can appear, such as the use of simplified names (*Claudio Pinhanez* instead of *Claudio Santos Pinhanez*), the use of social media pen-names (*@cinhanez* instead of *Claudio Santos Pinhanez*), or the use of nickname (*Darth Vader* instead of *Claudio Santos Pinhanez*.

To deal with some of the aforementioned difficulty factors, for this work we have developed a system to match names and locations of users using three different string distance functions:

1. *Exact matching (EM)*: a match is found if all the names of an SMN user are identical to those of a client
2. *Entity Distance 1 (ED1)*: designed to consider misspellings and transpositions between adjacent characters as a match. For instance, the user "Jooa Paulo" matches the client "Joao Paulo", and the user "Carolina" matches "Carolina". In this case, the threshold σ_1 is used to define a match only if the similarity value is above this threshold.
3. *Entity Distance 2 (ED2)*: designed to match abbreviations and some nicknames. For example, the user "Joseph S." matches the client "Joseph Salem"; the user "Fabinho" matches the client "Fábio", and "Mari" matches "Mariana". Similarly to ED1, the threshold σ_2 is used to define a match.

The execution of three aforementioned matching algorithms results in three distinct sets of users, denoted Ω_{EM}, Ω_{ED1} and Ω_{ED2}. The resulting set of users Ω_{All} corresponds to the union of those individual sets. That is, $\Omega_{All} = \Omega_{EM} \cup \Omega_{ED1} \cup \Omega_{ED2}$, where $\Omega_{EM} \cap \Omega_{ED1} \cap \Omega_{ED2} \neq \emptyset$ or $\Omega_{EM} \cap \Omega_{ED1} \cap \Omega_{ED2} = \emptyset$, depending on the data.

It is worth mentioning that the Jaro Winkler similarity filtering [20] is used prior to calling ED1 and ED2, to eliminate weak matches such as 'Maria' and 'Maria das Graças Silva'. Furthermore, ED1 and ED2 may return more than one match for the same user, whenever the result is above the given threshold. In this work, only the matching with the highest value is considered.

4. EXPERIMENTS

In this section we present the results of applying the proposed system on a dataset containing 9 millions of posts

from Twitter, which have been produced by about 1.4 million users. This data has been gathered by means of the GNIP social media data provider [8].

Mainly, these experiments have two different purposes. First we aim at evaluating the numbers related to applying the system on this 9 million dataset, i.e. how many posts and users are returned by using the system. And second, we focus on a quality analysis to validate those numbers by means of a manual inspection of samplings of this dataset.

The life event detection system has been implemented for six types of life events: Marriage, Graduation, Travel, Birthday, Birth,and Death. For each one, a training dataset of about 2 thousand samples has been manually labeled as either life event or non life event, and a distinct classifier has been trained. The training data has been obtained with the Twitter Search API [17]. For this work we make use of Naive Bayes classifiers using bag-of-words features [19]. The main parameters, i.e. θ, σ_1 and σ_2, have been set to 0.5, 0.95 and 0.95, respectively.

4.1 Quantitative Results

As we mentioned, the first experiment has as main purpose to evaluate how many posts and users are returned after carrying out each phase of the proposed system. The results of applying the implemented life event detection system on 9-million-tweet dataset is summarized in Figure 2. In this case, the Filter phase has returned 347 thousand posts from about 220 thousand users. Then, after going through the Detect module, 42 thousand posts, from about 19 thousand users, have been detected as life events. It is worth noting the large difference in terms of proportion from one phase to another. The Ingest phase captures a very large dataset, i.e. 9 million posts. Then, Filter finds out that only 3.7% of these posts can be of interest. However, the Detect phase shows that from these 347K% of posts, only 42 thousand (0.45% of 9M or 12% of 347K) are really those that the application is looking for. Considering that many of the current search system are rule-based, these results indicate that our proposed system can avoid a useless search on about 88% of the posts returned, 307 thousand posts in this case.

In Table 1 we present the results of the experiment above for each type of life event. We can observe that about 12% of the posts filtered have been generally confirmed as life events, but this proportion can vary according to the type of life event. For instance, for the Marriage class, from the 182,096 posts that the filter considered as possible life event, the machine learning algorithm detected 19,475 (10.6%) as being actually life events, which is close to the average. The Graduation type, on the other hand, presented a much larger proportion (43.21%), while Death and Travel smaller ones (5.47% and 8.26% respectively). We believe that this difference can happen either due to the period of the year in which the data is gathered (Graduation supposedly has more posts in certain periods of the year), or even due to the type of life event that may contain more non life events (Travel for example, which may present many posts from marketing agencies) or even less life event posts (for instance Death, whereas people might to be more introspective).

To evaluate the entity matching, we have a built a dataset

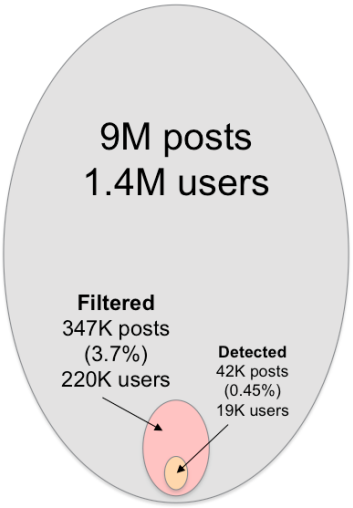


Figure 2: Results on the 9M dataset.

Table 1: Detailed results on the 9 million dataset.

Life event	Filter (% of 350k)	Detect (% of 42k)
Marriage	182,096 (52.4)	19,457 (46.5)
Graduation	25,676 (7.4)	11,097 (26.5)
Travel	22,596 (6.5)	1,868 (4.5)
Birthday	33,305 (9.6)	3,604 (8.6)
Birth	48,687 (14.0)	3,881 (9.3)
Death	35,242 (10.1)	1,929 (4.6)
Total (% of 9M)	347,602 (3.7)	41,836 (0.45)

containing 1.6 million users using publicly-available data. The users on this dataset have been matched against the 19 thousand users that have been detected as the ones that posted life events in the 9M dataset. The results and this process are illustrated in Figure 3. Note that we have conducted two different experiments. The first one matches these users by taking into account only their names, since we consider this as the minimum information we will be able to obtain from the SMN. In this case, 983 users have been found as probable matches. In the second experiment, where both names and locations are considered, only 5 users have been found. This shows that the precision of entity matching can be increased considering more for this process. On the other hand, this will also reduce the size of the resulting matching set.

In order to validate the above results, we performed a random sampling of 23 thousand posts (from the 9 million set) focusing on quality analysis. The number of posts filtered and detected are shown in Figure 4. The total of posts filtered is 1,008, from which 105 have been detected as life events. Similar to the results on the 9 million set, only about 10% of the filtered posts have been detected as life events. Detailed numbers, for each type of life event, are presented in the columns Filtered and Detected in Table 2.

Those 1,008 posts resulting from the Filter module have

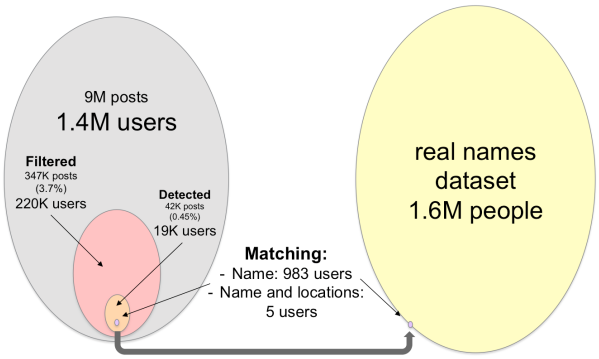


Figure 3: Entity matching results.

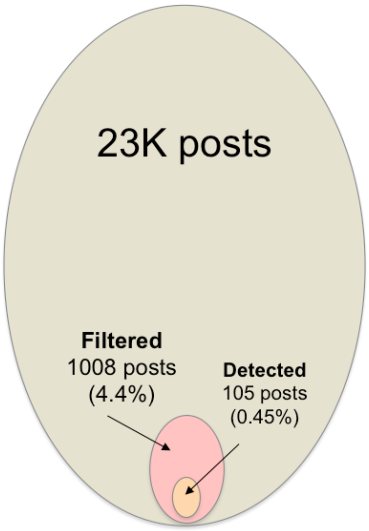


Figure 4: Results on the 23K sampling.

Table 2: Number of posts returned per life event type on the 23K sampling.

Life event	Filtered (% of 1008)	Detected (% of 105)	Ground-Truth (% of 142)
Marriage	162 (16.2)	8 (7.6)	7 (4.9)
Graduation	70 (7.0)	26 (24.7)	15 (10.6)
Travel	474 (47.4)	55 (52.4)	99 (69.7)
Birthday	102 (10.2)	11 (10.5)	12 (8.5)
Birth	107 (10.7)	4 (3.8)	7 (4.9)
Death	93 (9.3)	1 (9.5)	2 (1.4)
Total (% of 23K)	1008 (4.4)	105 (0.45)	142 (0.6)

been then manually inspected in order to verify whether the Detect phase has assigned the correct probability or not. The total of posts for each type of life event are listed in the Ground-Truth column in Table 2. It can be observed that our system presents numbers that are close to what was found by the manual inspection. By comparing the manual inspection with the results of the system, we have been able to compute the confusion matrix presented in Table 3, which contains the total number of true positives, true negative, false positives and false negatives. This has allowed us to compute the values for accuracy, precision and recall [16], which were at about 89%, 65% and 48%, respectively. In this case, a true positive consists of a posts that contains a life event (according to the manual inspection) and is correctly detected by the system, a true negative is not a life event and is correctly ignored by the system, a false positive is not a life events but is detected by the system, and a false negative is a life event but is not detected by the system. As a consequence, the precision represents the proportion of detected posts that contain life events, and the recall the proportion of life events that have been found by the system. It is worth noting that there is a trade-off between precision and recall that is set according to the value of θ, where lower values can increase recall and large values increase the precision (see Figure 5).

Table 3: Confusion matrix of the 1008 filtered posts found on the 23K sampling.

		Manual labeling	
		Life Events	**Non Life Events**
Life Event Detection System	**Positive**	68 (6.7%)	37 (3.7%)
	Negative	74 (7.3%)	829 (82.4%)

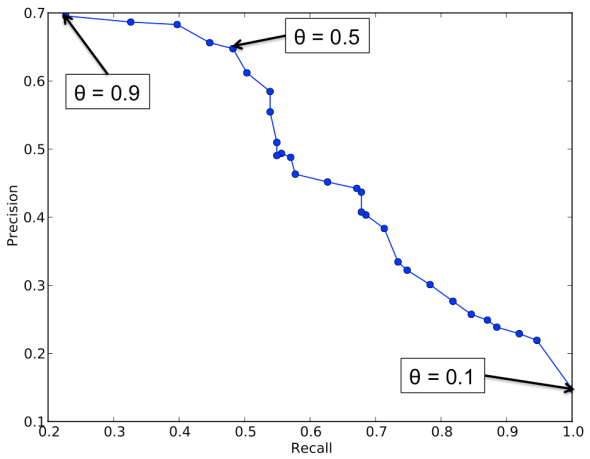


Figure 5: Precision/Recall trade-off by varying θ from 0.1 to 0.9.

Similarly, to validate the quality of the entity matching algorithm we have done a random sampling of 500 users and manually inspected the correctness of the matchings found. In this case, the entity matching algorithm returned 72 users, being 43 found by EM, 13 by ED1 and 16 by ED2. But, as we mentioned, both ED1 and ED2 can return more than one matching per user if the matching algorithm returns a value above the threshold σ_1 and σ_2. For a better analysis of the algorithm, in Table 4 and Table 5 we present the confusion matrices of both ED1 and ED2 considering all matches. The former has found a total of 476 matches, with an accuracy of about 91%, precision of 10.4% and recall of 71.4%, while the latter has returned a total of 452 matches, 94% of accuracy, precision of 50% and recall of 94%.

Table 4: Confusion matrix for ED1 on 500 users.

		Manual labeling	
		Match	**Non Match**
Entity Matching System	**Positive**	5 (1.10%)	43 (9.0%)
	Negative	2 (0.4%)	426 (89.5%)

Table 5: Confusion matrix for ED2 on 500 users.

		Manual labeling	
		Match	**Non Match**
Entity Matching System	**Positive**	17 (3.7%)	17 (3.7%)
	Negative	1 (3.9%)	427 (96.1%)

5. CONCLUSIONS

In this work we presented a system for personalized offer based on life event detection. Once the system detects users posting life events on a social media network, these users are matched against an internal database of clients to decide what is the best approach to offer them a service or product. We described a way to implement the entire system, and presented the results of applying the system on a dataset of 9 million posts. From this set, a total of 42 thousands life events have been found, with a projected accuracy of 88.90% and precision of 65%. This indicates that, in a normal day of 20 million posts published by Brazilian users, for instance, the system presents the ability to detect around 91 thousand posts a day, being about 60 thousand of them correct. Besides that, it is worth mentioning that the system is scalable since it has been implement with the MapReduce programming paradigm.

Future work can follow many different and complementary paths. Accuracy is important and could be improved by evaluating other types of classifiers and features, as well as increasing training data. The addition and evaluation of other types of life events could be important to better understand the way people behave on the SMNs. Furthermore, the adaptation to a real-time streaming platform such as the IBM InfoSphere Streams would allow the system react very quickly (near to real-time) once the users post life events.

6. REFERENCES

[1] Atefeh, F., and Khreich, W. A survey of techniques for event detection in twitter. *Computational Intelligence* (2013), n/a–n/a.

[2] Bilenko, M., Mooney, R., Cohen, W., Ravikumar, P., and Fienberg, S. Adaptive name matching in information integration. *IEEE Intelligent Systems 18*, 5 (Sept. 2003), 16–23.

[3] Cohen, W. W., Ravikumar, P., and Fienberg, S. E. A comparison of string distance metrics for name-matching tasks. pp. 73–78.

[4] De Choudhury, M., Counts, S., and Horvitz, E. Major life changes and behavioral markers in social media: Case of childbirth. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work* (New York, NY, USA, 2013), CSCW '13, ACM, pp. 1431–1442.

[5] Ehrlich, K., and Shami, N. S. Microblogging inside and outside the workplace. In *ICWSM* (2010).

[6] Eugenio, B. D., Green, N., and Subba, R. Detecting life events in feeds from twitter. *2012 IEEE Sixth International Conference on Semantic Computing 0* (2013), 274–277.

[7] Felt, A. P., and Wagner, D. Phishing on mobile devices. In *In W2SP* (2011).

[8] GNIP. GNIP, 2014. [Online; accessed 28-May-2014].

[9] IBM. IBM InfoSphere BigInsights, 2014. [Online; accessed 28-May-2014].

[10] Kwak, H., Lee, C., Park, H., and Moon, S. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web* (New York, NY, USA, 2010), WWW '10, ACM, pp. 591–600.

[11] Levenshtein, V. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady 10* (1966), 707.

[12] Lin, J., and Dyer, C. *Data-Intensive Text Processing with MapReduce*. Claypool Publishers, 2010.

[13] Liu, F., Weng, F., and Jiang, X. A broad-coverage normalization system for social media language. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1* (Stroudsburg, PA, USA, 2012), ACL '12, Association for Computational Linguistics, pp. 1035–1044.

[14] Peled, O., Fire, M., Rokach, L., and Elovici, Y. Entity matching in online social networks. In *Social Computing (SocialCom), 2013 International Conference on* (Sept 2013), pp. 339–344.

[15] Raad, E., Chbeir, R., and Dipanda, A. User profile matching in social networks. In *Network-Based Information Systems (NBiS), 2010 13th International Conference on* (Sept 2010), pp. 297–304.

[16] Sokolova, M., and Lapalme, G. A systematic analysis of performance measures for classification tasks. *Information Processing and management*, 45 (2009), 427–437.

[17] Twitter. Using the Twitter Search API, 2014. [Online; accessed 28-May-2014].

[18] Vosecky, J., Hong, D., and Shen, V. User identification across multiple social networks. In *Networked Digital Technologies, 2009. NDT '09. First International Conference on* (July 2009), pp. 360–365.

[19] Weiss, S. M., Indurkhya, N., and Zhang, T. *Fundamentals of Predictive Text Mining*. Springer London, 2010.

[20] Winkler, W. E. String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. In *Proceedings of the Section on Survey Research Methods (American Statistical Association* (1990), pp. 354–359.

A User-Study on Context-aware Group Recommendation for Concerts

Simen Fivelstad Smaaberg, Nafiseh Shabib, John Krogstie
Norwegian University of Science and Technology
Trondheim, Norway
smaaberg@stud.ntnu.no, {shabib, krogstie}@idi.ntnu.no

ABSTRACT

In this paper, we present a prototype of a group recommendation system for concerts. The prototype is context sensitive taking the user's location and time into account when giving recommendations. The prototype implements three algorithms to recommend concerts by taking advantage of what users have listened to before: a collaborative filtering algorithm (K-Nearest Neighbor), a Matrix Factorization algorithm, and a Hybrid approach combining these two. The usability of the prototype was evaluated using the System Usability Scale and a user centered evaluation was performed to evaluate the quality of recommendations. The results from the usability evaluation shows that users generally were satisfied with the usability of the prototype. The results from the Quality Evaluation shows that the K-Nearest Neighbor and Hybrid approach produces satisfactory results whereas the Matrix Factorization implementation was experienced to be a bit poorer. The users testing the prototype were generally satisfied with the quality of recommendations.

Keywords

collaborative filtering, group recommendation, context-aware

1. INTRODUCTION

Recommendation technology is becoming an increasingly important part of large systems such as Amazon.com and eBay.com, and also in the music industry for example Spotify, iTunes and Last.fm recommendations are used in to an increasing degree. A context-aware group recommendation system is a recommendation system that recommends items for groups of people instead of for a single person in the given context [14]. The group and context part adds additional challenges compared to a normal recommendation system for individual users [14]. A group of people is more dynamic than a single person. You have to consider how the group is formed, how unified recommendations for the whole group can be provided, and the dynamics within the group [6] [11]. Context in addition has many traits, but can be seen as external constraints that affects the recommendation process. This makes the algorithms more complicated. The purpose of this paper is to present a context-aware group recommendation system for concerts that takes the location and time of a user into account when making recommendations. This is done to show that traditional methods for Music Recommendation Systems can also be applied when concerts are recommended and extra context-variables have to be considered. Even though group recommendation systems have been explored, they are not as thoroughly investigated as recommendation systems for individuals. The same can be said for recommendation systems for concerts, and context-aware group recommendation systems, where limited existing research has been found, in particular on the perceived usability and quality of such solutions.
In the next section, we present the main approach, data model and algorithms. In Section 3, the experiments and evaluations is presented. Related work is described in Section 4, before we conclude in Section 5, pointing to future work.

2. DATA MODEL AND RECOMMENDATION ALGORITHM

Illustrating the problem of context-based group recommendation, we take the following scenario as an outset:

A group of friends is traveling to a big city to stay there for a week. Here they wants to attend a concert. Their tastes in music are quite different, so choosing what concerts to attend is a challenge. Moreover, they may not be familiar with all the bands playing and would like to have an application that give them recommendations concerning which concerts to attend based on the type of music they have listened to before and their personal musical preferences.

We consider the following requirements for designing context-aware group recommendation for concerts:

- Recommendations need to be based on the user's listening preferences
- The system should be location-aware (concerts close to a user are preferred)
- The system should be time-aware (not recommend concerts that already have taken place or concerts too far ahead in time)
- Context relaxation should be supported (recommendations for more widespread locations or time-period can be attempted if not enough concerts are found for the given location or time)

2.1 Data model

The main concepts used to support these requirements are depicted in the data-model in Figure. 1. Users have previously listened to music by existing artists. The artists

are in addition tagged (with musical categories), and one have information about the tags/interests of users. Based on listening and tagging history, user-similarity is calculated. Artist play concerts. The concerts take place on venues at a certain time and space (geographically located) in a particular city.

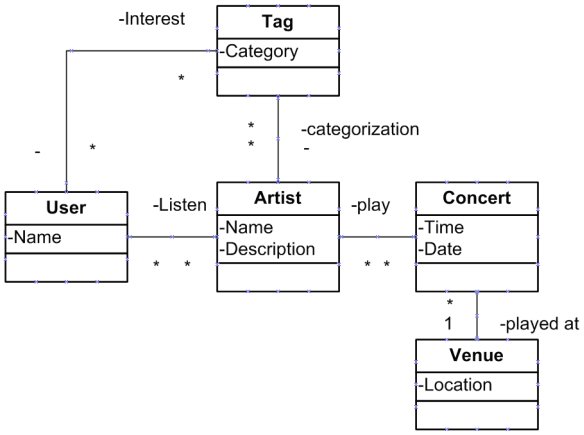


Figure 1: Conceptual Data Model

2.2 Context and Context relaxation

Context captures information that is not part of the database, such as the user location or the current time. A user would probably want to get recommendations for concerts in locations close to where he is located, and not get recommendations for concerts too far ahead in time (unless planning ahead for a later travel of course), or for concerts that already have taken place. Therefore the definition of these context parameters should be a central part of any concert recommendation system (`CRS`). In particular, a context parameter can be relaxed upwards by replacing its value by a more general one, downwards by replacing its value by a set of more specific ones or sideways by replacing its value by sibling values in the a context-hierarchy [13], which in our case would be an adjecent later day or a neighboring city. To enable for relaxation of the location parameter, the 100 concerts closest to the location specified is also fetched. This is done by utilizing the Haversine formula[1]. This formula can be used to estimate the shortest distance between two points on the earth surface[3]. In addition, concerts within 5 days of the date range specified is fetched to support relaxing of the date range parameter.

2.3 Listen Count Normalization

We have retrieved listening history from the Last.fm music discovery service. The algorithms in our work are based on explicit feedback from users, subsequently there is a need to normalize the listening counts to a predefined scale so that the algorithms can work optimally on Last.fm dataset. Similarly to the approach taken by [4], for each user, U, its listening counts for each artist, A, is normalized using the Cumulative Distribution Function (CDF) of the artist listenings for U. The artists with a listening count falling within the first 10% of this distribution is assigned a rating of 10; the artists falling within the first 20% of the distribution is assigned a rating of 9; and so on until the artists within 90 to 100% of the distribution is assigned a rating of 1.

[1]http://en.wikipedia.org/wiki/Haversine_formula

2.4 Neighborhood Model

The K-Nearest Neighbor (KNN) algorithm was one of the first approaches used in user-based recommendation [5]. In our work, the KNN algorithm is split into two phases:

1. Filter dataset
2. Recommend concerts

2.4.1 Filter dataset

In a KNN approach, the K-Nearest Neighbors of u are used as a basis for recommendation. For simplicity reasons we state that a user that have not listened to any of the artists in A' cannot be considered by the algorithm. This can be done because a user that has not listened to any of the artists in A', could only contribute with a listening count of 0 to all of them, and therefore the user might be left out.

Since the set of artists that are considered in the algorithm has been reduced to the set of artists A' playing at one of the concerts in C', implicitly the set of users considered for the algorithm can be reduced to the set of users that have listened to one or more of the artists in A'.

$$U' = \{u' : \forall u' \in U \exists a \in A' listenedTo(u', a)\} \quad (1)$$

2.4.2 Recommend concerts

In a KNN algorithm, the K most similar users to u are found, and their ratings are used as a basis for recommendation. To find these similar users, we applied cosine similarity based on listening count of two users for each artist. So, the user vector $\mathbf{w}_i$ for a user $u_i \in U$ is defined as the vector of the users listening counts to each of the artists in A.

$$\mathbf{w}_i = \{listenCount(i, a) : a \in A\} \quad (2)$$

In a normal K-Nearest Neighbor algorithm the K users with the highest similarity would now be identified and used as a basis for recommendation. For the purpose of a `CRS`, this is not enough. Here, a rating for each of the concerts, c_i, in C' have to be predicted. Therefore, a 3 step process is undertaken for each of the concerts:

1. Find the K users, U'', with the highest similarity to u from the subset of U' that have listened to one or more of the artists performing at that concert.
2. Calculate the predicted rating for each of the artists a playing at the concert. $TotalSimilarity$ is defined as the sum of similarities to u from each of the users in U''. Each of the users u_i in U'' will contribute to the predicted rating with a percentage of $\frac{sim(u_i,u)}{totalSimilarity}$. The actual contribution is influenced by the rating given to a by u_i, so this is multiplied with $rating(u_i, a)$. The predicted rating for an artist i will then be:

$$artistRating_i = \sum_{j=1}^{n} \frac{sim(u_j, u) \times listenCount(u_j, a_i)}{totalSimilarity} \quad (3)$$

3. The overall predicted rating for the concert c_i as a whole for user u is given by the average of the predicted

ratings to each of the m artists performing at c_i.

$$KNNRating^u_{c_i} = \frac{\sum_{k=1}^{m} artistRating_k}{m} \quad (4)$$

2.5 Latent Factor Model

Similarly to [7], the $n \times m$ user-artist matrix M is reduced into a set of user vectors, V, where $V_i \in \mathbb{R}^f$ and artist vectors, B, where $B_i \in \mathbb{R}^f$. f is the number of latent factors to extract (dimensionality of the latent factor space). In this work, the user-artist matrix consists of the normalized listen counts for all of the users in U and the artists in A. To approximate a user u's rating for an artist a, $\hat{r}_{ua}$, the dot product between u's and a's latent factor vectors $V_u B_a$ is performed. As [7] says: this dot product "captures the interaction between user u and item i - the users' s overall interest in the item's characteristics".

$$\hat{r}_{ua} = B_a^T V_u \quad (5)$$

We will refer to this model as $PureSVD$. It uses $f = 64$ features which are optimized by running over 120 iterations. The implementation is based on Timely Developments[2] implementation of the algorithm.

The overall predicted rating for the concert c_i as a whole for user u is given by the average of the predicted ratings to each of the m artists performing at c_i.

$$mfRating^u_{c_i} = \frac{\sum_{a=1}^{m} \hat{r}_{ua}}{m} \quad (6)$$

2.6 Hybrid Model

The predictions given by the algorithms in the previous two sections are in this phase aggregated to produce the final top N concerts to return to the user. For each of the concerts, c_i, in C the final rating for the concert for u, $r_{uc_i} \in R_u$ is given by:

$$\hat{r}_{u_i c} = \frac{mfRating^u_{c_i} + knnRating^u_{c_i}}{2} \quad (7)$$

The N concerts with the highest rating r_{uc_i} in R are selected and returned to the user.

2.7 Aggregation strategy

In this work, an *average* aggregation strategy (which computes the group preference for an item as the average of group members' preferences for that item) is used to aggregate individual ratings into a group rating for a concert. Since, in a music recommendation system we have to utilize implicit feedback, there is no such thing as a negative preference. For example, a listen count of 0 does not necessarily mean that a user does not like the artist, just that the user has not listened to the artist before. The user might like the artist, but he has not discovered it, or he might dislike it. Therefore, it is impossible to know for certain how to interpret a listen count of 0. Similarly, a low listening count may not mean that a user does not like the artist, he might just have discovered the artist or just joined the system. Again, it is impossible to know. Thus, we can safely assume that Least Misery (which computes the group preference for an item as the minimum among all group members' preferences for that item) in an aggregation method would not be applicable thus we used the *average* strategy.

[2]http://www.timelydevelopment.com/demos/NetflixPrize

3. EXPERIMENTS

We evaluate our group recommendation system from two major angles. First, from the *usability* perspective (Section 3.1), and second *quality* perspective (Section 3.2).
We implemented our prototype system using Java and MySQL for the back end. The front end was developed in JavaScript and HTML5, and is based on the Durandal.js[3] Model View Viewmodel framework.

Dataset description: We use the Last.fm dataset for evaluation purposes. Last.fm has become a relevant online service in music based social networking. In our particular `CRS` the data was fetched using Last.fm's publicly available API[4].The dataset as seen in Table 1 consists of 2,891 concerts in Vancouver, New York, London, Oslo, and surrounding areas, between 18. February 2014 and 6. June 2014. The dataset was built by first fetching concerts within a 100km radius from the specified cities. Then, information about the artists performing at those concerts were fetched. Users that have listened to the artists found are then fetched, before the 30 most listened to artists for each user are fetched and saved. In addition to these data, information about the venue that each concert is held at and the most used tags for each artist is stored. When a new user was created where no existing data was present in Last.fm, he would need to rate at least 5 artist that are registered in Last.fm. In the quality experiments below, we have looked upon differences when providing 5 or 10 ratings.

Property	Count
Users	25720
Artists	80877
Concerts	2891
Listening counts	769370
Tags	159348
Tags for artists	1358715
Artist concert participation	6845
User similarities	17025096
Venues	596
User features	17025096
Artist features	5085312

Table 1: Dataset properties

3.1 Usability Experiment

In this work, to evaluate the usability, we recruited 15 participants to use the system and answer three questionnaires, the System Usability Scale (SUS), an Application Specific survey (AS) and a questionnaire to gather Background Information (BI). The result view of the system can be seen in Figure 2 giving an indication on the look and feel of the system.

The System Usability Scale is a "reliable, low-cost usability scale that can be used for global assessments of systems usability" [2]. It gives a global view of subjective assessments that indicates how users agree or disagree with the

[3]http://durandaljs.com/
[4]http://www.last.fm/api

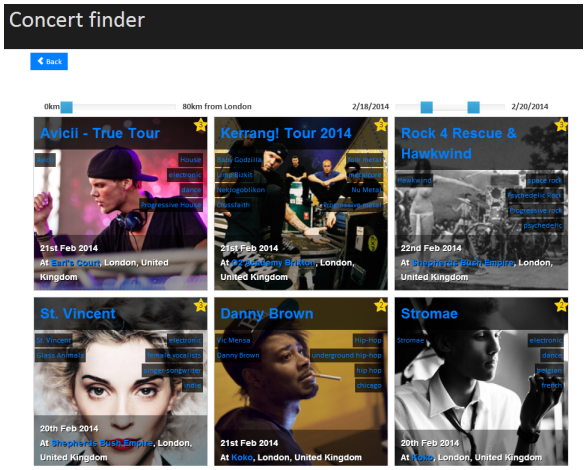


Figure 2: Result view of the prototype

statement. Nielsen suggests that 5 users are enough to find the majority of usability problems of a system, those 5 participants could reveal about 80% of all usability problems [9]. In general, one should run usability tests with as many participants that schedules, budgets, and availability allow. On this basis we are confident that with our 15 users, we have covered the main usability issues of the application.

3.1.1 *Results*

The results from the SUS survey yielded a SUS score of 79.83. [1] proposes an adjective rating scale to help determine what SUS scores actually mean. According to these adjective ratings, a SUS score of 79.83 would fall into somewhere between *Good* and *Excellent*. There is no absolute score when it comes to usability evaluations, but a score of 79.83 is a good indication on that the users found the usability of the prototype satisfactory. The results from the Application Specific survey (AS) showed that 66% of the participants believed that the would use this application in future. 87% of the participants answered either *OK* or *Satisfied* when asked how satisfied they were with the quality of recommendation from these music recommendation services, although the real quality evaluation in a group setting was postpone to the quality evaluation reported below. Concrete improvement proposals gathered were used to develop the second version of the system where more detailed quality experiment was undertaken

3.2 Quality Experiment

To evaluate the quality of recommendations from the improved system, two groups consisting of two and three people respectively were asked to find recommendations both individually and in a group setting, for different dates and places, and to rate how satisfied they were with the given recommendations. For this purpose, we showed three different lists where each list was the result of using the three different algorithms (k-NN algorithm, MF algorithm, the hybrid approach). Each of the lists are given "random" case ids and placed in a random order. The participants were asked to find recommendations individually, in a group of two people, and in a group of three people, for two different timespans (18/02/2014 – 03/03/2014 and 05/03/2014 – 09/07/2014), and two different cities (London and New York). When the second group were asked to find recommendations for a group of 3 people, a user from the first group were added to the recommendation process. For each step, they rated each of the algorithms on how satisfied they were with the recommendations given on a scale from 1-5, where 1 is *Very dissatisfied* and 5 is *Very satisfied*.

Algorithm	Number of selections	Percentage
Matrix Factorization	7	17.5%
k-Nearest Neighbor	16	40.0%
Hybrid approach	17	42.5%

Table 2: Preferred algorithm selection by users

3.2.1 *Results*

As seen in Table 2, the MF algorithm were overall picked as giving the most appealing results 7 out of 40 times, the kNN algorithm 16 out of 40 times, and the hybrid approach in 17 out of 40 cases. Overall, the kNN algorithm received an average rating of 3.72 in the 40 responses, the Hybrid approach 3.62, and the MF algorithm an average of 2.87 as seen in Table 3. Table 4 shows statistics when recommendations were given for groups consisting of 1, 2 and 3 users respectively. From these results there is a clear trend that the kNN and the hybrid approach tend to produce more satisfying recommendations than the MF approach as the average ratings given to the two are generally lower, and they were picked as the favorite algorithms significantly more. An overall average rating of 3.72 and 3.62 out of 5 from the kNN and Hybrid approaches respectively, indicates that the participants were reasonably satisfied with the results given. In general, recommendations given for users created based on 10 of the user's favorite artists, produced more satisfying results than when 5 artists were used in the user creation process. Moreover, by increasing the number of users in a group from two to three users user satisfaction is decreasing.

Algorithm	Average rating	Variance	Standard Deviation
Matrix Factorization	3.13	0.73	0.85
k-Nearest Neighbor	2.28	0.92	0.96
Hybrid approach	2.38	0.75	0.87

Table 3: Overall Average statistics per algorithm

3.3 Insight about Serendipity in Concert Recommendation Systems

Serendipity is concerned with the novelty of recommendations and in how far recommendations may positively surprise the user [12] and it has received increased attention that recommendation system should provide novel and serendipitous recommendations. The emphasis should be put on the lesser known artists, the *long tail* of the listen count curve. However during the development and testing of this prototype it was observed that a full focus on this may not be the best approach for a `CRS`. Our findings show that people tend to prefer to go to concerts with artists they

	Average rating			Variance			Standard Deviation		
Algorithm	1	2	3	1	2	3	1	2	3
Matrix Factorization	2.7	3.3	3.1	0.46	1.12	0.77	0.67	1.06	0.88
k-Nearest Neighbor	4.2	3.7	3.5	0.62	1.12	1.17	0.78	1.06	1.08
Hybrid approach	3.7	3.6	3.4	0.68	0.93	0.93	0.82	0.97	0.97

Table 4: Statistics when recommendations were given for groups consisting of 1, 2 and 3 users respectively

kNN		MF	
Artist	# of listeners	Artist	# of listeners
Avicii	548	Arctic Monkeys	2388
Katy Perry	676	Lorde	554
Arctic Monkeys	2388	Beyoncé	585
Disclosure	535	Metronomy	418
Kanye West	1578	Cut Copy	378
Nine Inch Nails	1270	Alkaline Trio	383
The National	1687	Panic! at the Disco	
Drake	712	Slowdive	308
Interpol	784	Katy Perry	676
Arcade Fire	2165	Pretty Lights	234
Average	**1234**	**Average**	**632**

Table 5: Number of listeners for the top artist playing at the top 10 concerts between 18/02/2014 and 17/07/2014 in London for user *simensma*

already familiar with and the concert scene might not the place were people try to be adventurous and discover new music, it is easier, more convenient, and cheaper to discover and becoming familiar with new artists first, before deciding to attend a concert with them. This might be one of the causes why the kNN and Hybrid approaches received better ratings from the test users when it came to quality of recommendations, as collaborative filtering (CF) approaches tend to have a popularity bias causing the more popular artists to be recommended. An example of this can be seen in Table 5 where the top artist and how many users have listened to them for the 10 top concerts recommended for the user *simensma* in London between 18/02/2014 and 17/07/2014 can be seen. The 5 most frequently used tags to describe *simensma*'s top artists are *electronic*,*house*,*dance*,*indie*, and *electro house*. On average, 1234 users had listened to each of the artists recommended by the kNN algorithm whereas 632 users on average had listened to each of the artists recommended by the MF algorithm.

3.4 Threat to validity

The quality evaluation was performed with only two groups of 2 and 3 people. This low number of participants means that each participant had a very large impact on the results. The statistics produced when a user was created with 5 and 10 favorite artists, were based on n = 10 samplings each; the same was the case with the statistics produced for the results with varying group sizes. By looking at the top tags used for the artists each of the users registered, it is apparent that the users' taste in music are quite different as they share few top tags amongst them. However, because of the low number of users and sample sizes, even with this diversity, it cannot be said that these five users are representative for the whole potential user base, and therefore, further testing should be performed to measure the Quality of Recommendations created by the prototype. Even though more testing is needed, there still is a strong indication that the KNN and Hybrid approaches perform better than the MF approach as suggested with a sample size of n = 40. Similarly, it can be said that the five users testing the prototype were reasonably happy with the resultss.

4. RELATED WORK

Group Recommendation Systems try to provide recommendations to a group of people instead of a single individual. There are two main approaches of accomplishing this: calculating recommendations individually for each of the members of the group, and then aggregating the individual results, or merging the preferences of each of the members of the group, and then providing one set of recommendations based on the merged profile [10][8]. In either of the approaches, there are many ways this merging can be accomplished [8]. This includes least misery, average, and average without misery. The choice of aggregation strategies should be decided based on the problem you are trying to solve, as there is no universal best strategy that works in all cases. As argued above, we choose an average aggregation strategy. Recommendation Systems for Music (MRS) have increasingly become an important part of music services. Services such as iTunes, Spotify, last.fm and Pandora all incorporate music recommendations centrally in their user interface. With an ever growing collection of music, these services compete in finding new and innovative ways on how users can discover new music. Celma [4] identifies three use cases typical for a MRS: neighbor finding, playlist generation and artist recommendation. Neighbor finding consists of finding users with a similar taste in music as you. Playlist generation usually means finding songs to recommend for a user, but instead of just returning the top N songs, songs that go well together are preferred. Artist recommendation usually consists of finding artists based on a user's profile, be it the artist with the highest predicted rating or novel artists. Different services apply a variety of techniques when it comes to the recommendation process [15]. Some of them are acoustic analysis, text analysis, editorial review, and the use of activity data. This diversity indicates that people have different ways to think about music. Enthusiasts and savants might prefer to try out new and little known artists, whereas casual users might prefer well known artists and the latest 'big hits'. With such a diverse set of expectations, creating a music recommendation system that works well for all of them is challenging. In general these techniques are provided for creating recommendations for individual users, little work being done to support groups of users.

5. CONCLUSION

In this paper, a prototype of a context-aware group recommendation system for concerts was presented. The prototype implemented three different algorithms, a Matrix Factorization algorithm, a k-Nearest Neighbor algorithm and a Hybrid approach of the two. The usability of the prototype was evaluated using the System Usability Scale (SUS) and an Application Specific Survey (AS). 15 people were asked to undertake these surveys. In total, the prototype got a SUS score of 79.83 which is a good indication on that the users found the usability of the prototype satisfactory. However, the comments from the free text answers shows that there where room for improvements. The AS mainly focused on the usability of the context relaxation part of the prototype, to find out if it was easy to find concerts close to the parameters specified when it comes to time and location. The results from the AS showed that the users in general were satisfied with how this process worked. The goal for this prototype was to recommend concerts to a user within the location and timespan given that the user could be interested in attending. To evaluate how well this was achieved, a Recommendation Quality Evaluation(QE) was undertaken with two groups consisting of 2 and 3 people respectively. Through a range of scenarios, the groups were told to find recommendations for the dates and location asked about, and for each algorithm, rate how satisfied they were with the results. The results from the QE showed that the users generally were satisfied with the KNN implementation and the Hybrid approach, whereas they were less satisfied with the MF approach. The QE was also undertaken to see how different group sizes affected the quality of recommendations. The results showed that the users became less satisfied when the number of members in the group increased from one to two and three respectively, which is to be expected as different preferences has to be taken into account in larger groups. However, the QE was only performed with five participants, so there is a need for an evaluation with more participants to able to draw any further conclusions. Another way to go from here is to have a look at the context-aware part of the application. Is there any benefit in making relaxation of context an implicit part of the algorithm instead of something performed by the user explicitly? How would other context variables, such as listen recency affect the satisfactions when recommending concerts?

6. REFERENCES

[1] A. Bangor, P. Kortum, and J. Miller. Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of usability studies*, 4(3):114–123, 2009.

[2] J. Brooke. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189:194, 1996.

[3] C. A. Cassa, K. Iancu, K. L. Olson, and K. D. Mandl. A software tool for creating simulated outbreaks to benchmark surveillance systems. *BMC Medical Informatics and Decision Making*, 5(1):22, 2005.

[4] O. Celma. *Music recommendation and discovery: The long tail, long fail, and long play in the digital music space*. Springer, 2010.

[5] M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. *ACM Trans. Inf. Syst.*, 22(1):143–177, Jan. 2004.

[6] D. R. Forsyth. *Group dynamics*. Brooks/Cole, Pacific Grove, Calif., 2. edition, 1990. Donelson R. Forsyth. graph. Darst ; 24 cm. Fräüher u.d.T.: An introduction to group dynamics.

[7] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[8] J. Masthoff. Group modeling: Selecting a sequence of television items to suit a group of viewers. *User Model. User-Adapt. Interact.*, 14(1):37–85, 2004.

[9] J. Nielsen. *Usability engineering*. Elsevier, 1994.

[10] S. B. Roy, S. Amer-Yahia, A. Chawla, G. Das, and C. Yu. Space efficiency in group recommendation. *VLDB J.*, 19(6):877–900, 2010.

[11] N. Shabib, J. A. Gulla, and J. Krogstie. On the intrinsic challenges of group recommendation. In *RSWeb@RecSys*, 2013.

[12] G. Shani and A. Gunawardana. Evaluating recommendation systems. In *Recommender Systems Handbook*, pages 257–297. Springer, 2011.

[13] K. Stefanidis, E. Pitoura, and P. Vassiliadis. On relaxing contextual preference queries. In *MDM*, pages 289–293, 2007.

[14] K. Stefanidis, N. Shabib, K. Nørvåg, and J. Krogstie. Contextual recommendations for groups. In *ER Workshops*, pages 89–97, 2012.

[15] B. Whitman. How music recommendation works and doesn't work. `http://notes.variogr.am/post/37675885491/how-music-recommendation-works-and-doesnt-work`, 2013. Accessed: 2013-04-27.

Voting Based Group Recommendation: How Users Vote

Michal Kompan
Slovak University of Technology
Inst. of Informatics and Software Engineering
Ilkovičova 2, 842 16 Bratislava, Slovakia
name.surname@stuba.sk

Mária Bieliková
Slovak University of Technology
Inst. of Informatics and Software Engineering
Ilkovičova 2, 842 16 Bratislava, Slovakia
name.surname@stuba.sk

ABSTRACT
It has been shown that social information as group structure or personality characteristics improve the group recommendation. Sometimes no such information is available, specifically when ad-hoc groups are constructed. Moreover, often the items' content is not available (or users' preferences are unknown). In this paper we explore the usage of voting based group recommendation and the users preference for such a method settings – we analyze aggregation strategies preferences, sharing preferences and users re-rating consistency.

Categories and Subject Descriptors
H.3.3 [**Information Technology and Systems**]: Information filtering

General Terms
Experimentation, Human Factors

Keywords
Group recommendations, voting, aggregation strategies

1. INTRODUCTION
Group recommendation gets more and more attention in today's adaptive web-based applications [1]. Users' social activity over the Web is increasing and thus new domains and applications as movie, learning or games are available. When recommending to the group of users the social structure and personal characteristics plays important role from the group satisfaction point of view [3]. On the contrary, sometimes there is not possible to obtain these characteristics. When the group is constructed ad hoc – from "random" users it is almost impossible to collect information about the group structure or users characteristics (usually obtained by various questionnaires) [2].

One of the best performing approaches for the group recommendation, which is suitable for active groups is the recommendation based on voting of group members. Group members suggest their preferred items and then the voting is performed by the group. It is clear that the voting process, especially when performed online and when the goal is to reach consensus, can be influenced and enhanced by various aspects (e.g., sharing preferences, aggregation strategies, group size, users' consistency). In order to investigate the influence of these specific aspects we propose a voting mechanism in the domain of movies.

2. VOTING BASED RECOMMENDATION
Proposed approach consist of the construction of user's ratings matrix, which is created based on users' votes (*Items x Votes*). Every user can vote for the items already voted by other users, or the new item can be added as the suggestion to the group. Next, the matrix of normalized ratings is constructed (Min-max normalization) in order to minimize low or hight ratings influence to aggregation strategy. Finally, the total of three representative aggregation strategies (additive, multiplicative and additive with minimal satisfaction) are used in order to construct the group recommendation, which is presented to users:

1. Create user's rating matrix and the normalized rating matrix respectively.
2. Aggregate votes from group members (users rating matrix).
3. Recommend items with highest votes.

Not only the lack of users' preferences knowledge or sufficient group activity indicate to use the voting based group recommendation. Often there is no information about the recommended content available (e.g., movie genre, director), which are used for the standard similarity search. In the voting based approach, this information is processed by the users, thus no content analysis or the lack of new items is required or present.

2.1 Evaluation and Results
Proposed approach was implemented as a simple web-based application MovieRec and available for the free usage within the social network Facebook during the experiment. We expected that – users' ratings are more consistent as when no sharing preferences are presented. We also believe that users' ratings are influenced by the group context – users' re-ratings (rating previously rated item in new event and group) are influenced by the group and event context. The total of 73 real users within 10 days voted for 902 movies (obtained from IMDB database), which were self-divided into the 11 groups and 93 voting events.

The task presented to the users was to create or to join some event and try to reach consensus (based on the voting) on which items should be watched together within the group. For every created event the users voted for their

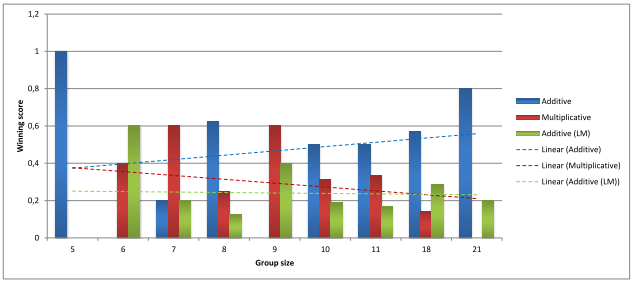


Figure 1: Ratio of winning voting strategies compared to the group size.

candidates to watch. They could create new suggestions until the event deadline. During the experiment we were observing the users' behavior based on the sharing preferences (in the half of events the preferences of other group members were visible), users' consistency and the performance of used aggregation strategy. After the event deadline, three lists of the generated recommendations were presented to every user of the group (additive, multiplicative and the additive with minimal satisfaction consideration strategy). Every user rated for the best recommendation of these three presented lists.

Results – aggregation strategies. Our first question was which strategy is preferred based on the group size. When comparing the winning strategy depending on the group size we discovered that larger groups (more single-users' preferences have to be aggregated) prefer additive strategy, while the decreasing trend can be observed when multiplicative strategy is used (Figure 1). Finally, the additive strategy with least misery performed the worst. This can be explained by the fact that least misery prefers votes from the minority, thus when only one user dislikes an item, this item will not be recommended. With the group size and users' satisfaction, the number of such users is increasing, thus the quality of recommendation is decreasing. Similarly, when the multiplicative strategy is used, low ratings of few users can influence whole recommendation dramatically, obtained results supports this hypothesis – the additive strategy within large groups balances the influence of deviating individuals and the rest of members.

Results – sharing preferences. Next, we focused on influence of sharing preferences. Users' events were divided into the two sets – users who saw preferences of their colleagues, and second set, where no sharing preferences were displayed. We discovered that the sharing preferences do not have (or have very small) influence on the user's ratings. The standard deviation of these two groups differs only 0.0212. Thus, we see that the users in our experiments considered the preference of others minimally, or were very consistent in their similar opinions and thus sharing preferences were redundant.

In general, the winner, in the most of events is the *additive strategy*, followed by the multiplicative and the additive with minimal satisfaction strategy. This is quite surprising result, while the minimal satisfaction seems to be not so desirable (from the majority points of view), especially when a large group is interacting. Obtained results clearly show that when a large group is requesting for the recommendation, the minimal satisfaction from the group point of view decreases the quality of recommendation. This is supported by the standard deviation of obtained votes for particular strategies (Table 1). From the average score point of view, the additive strategy with least misery outperforms the multiplicative, thus the preference diversity was probably small within the group members.

Table 1: Voting strategies comparison.

Strategy	Winning events	SD	Avg. vote
Additive	**184**	0.90	**4.14**
Multiplicative	147	**0.83**	4.08
Additive(LM)	138	0.95	4.12

Results – users' consistency. Finally, we investigated users' consistency over the various voting and events. We focused on movies rated by the user in some event and his/her rating for the same movie in other events. In order to minimize users' effort, if the movie was rated by the user before, we presented this rating as default value (and the user was able to adjust this rating). The total of 462 such "re-ratings" were given by the users, while only in 71 occurrences the users changed the value of previous rating. This is an interesting result, which can be partially caused by the pre-filled ratings. On the other hand, the proportion of users which were consistent (85%) indicates that users adjust their ratings to the actual group context minimally (which is supported by the social psychologist as the tendency to act consistent in various situations.

3. CONCLUSIONS

When there is no additional information about the group available, the voting strategy seems to be the optimal solution. Here, the recommendation task is moved to the group members directly. As we shown the additive and multiplicative strategy are more preferred by small groups, while on the other side for larger groups the additive strategy is preferred. Proposed voting approach revealed that the sharing preferences have no or minimal influence to the group members in adjusting their preferences.

4. ACKNOWLEDGMENTS

The authors wish to thank Ján Trebuľa for helping with implementation of MovieRec. This work was partially supported by the grants No. VG1/0675/11 and APVV-0208-10.

5. REFERENCES

[1] M. Kompan and M. Bielikova. Group Recommendation: Survey and Perspectives. In *Computing and Informatics*, 33(2):446–476.

[2] M. Gartrell, X. Xing, Q. Lv, A. Beach, R. Han, S. Mishra, and K. Seada. Enhancing group recommendation by incorporating social relationship interactions. In *Proc. of the 16th ACM Int. Conf. on Supporting Group Work*, pages 97–106. ACM, 2010.

[3] J. Masthoff and A. Gatt. In pursuit of satisfaction and the prevention of embarrassment: Affective state in group recommender systems. *UMUAI*, 16(3-4):281–319, Sept. 2006.

Semantic Social Recommendations in Knowledge-Based Engineering

Dirk Ahlers Mahsa Mehrpoor
NTNU – Norwegian University of Science and Technology
Trondheim, Norway
dirk.ahlers@idi.ntnu.no mahsa.mehrpoor@ntnu.no

ABSTRACT

We examine the application of semantic context-aware Recommender Systems to improve interaction and navigation in a design-centric engineering domain. The small scale of this specialised environment renders most Web-scale solutions unsuitable, mandating tailored approaches. We report on initial work to identify challenges and promising categories of personalisation and adaptation together with relevant context features taken from the whole environment consisting of users, organisation, and documents to overcome the sparsity issue in professional Information Access.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Storage and Retrieval—*Information Search and Retrieval*; J.6 [**Computer Applications**]: Computer-Aided Engineering

General Terms

Design, Documentation

Keywords

Manufacturing, CAx, Digital factories, Manufacturing design and product lifecycle management, Information Access

1. INTRODUCTION

In this paper, we examine social recommendation from an angle that has not yet received much attention. The angle is that of professional search and recommendation systems. In that case, social does not mean friends, followees and followers, or people who used the same Web site, but the social network of colleagues who work on similar projects or the same discipline. While this may limit the information to be gathered from the social circle, the focus on a specific domain can partly make up for it. For example, an engineer in a product design company can have overlapping social circles and implicit connections. One can be the organisation chart of the hierarchy, setting her up in relation to people within her department, her supervisors, her team, or her staff. Additionally, she will be part of the engineers that work in the company in different departments and different fields, and she may also be part of one or multiple projects. She might additionally be part of a management group or a specific specialization. All these roles and task mean different information access demands for her.

The setting we are examining is that of Knowledge-Based Engineering (KBE), which is an approach used in manufacturing and design engineering to not only capture available process and product knowledge, but to use it systematically in the design process. A focus lies on the reuse of knowledge and knowledge sharing between the involved engineers [7]. Our research is anchored in the LinkedDesign project[1] [2] which aims to provide integrated information and knowledge handling to improve engineering product development.

2. CONTEXT-AWARE APPROACH

The KBE scenario is a good case for personalisation in professional search. The application domain involves knowledge workers and domain experts who need improved Information Access for complex problems and work tasks, based on complex and heterogeneous documents. Even if the knowledge base is a limited in-house system, the documents therein are mission-critical and contain valuable heterogeneous knowledge. Our aim is to enhance the Information Access to provide improved semantic *navigation and interaction in information spaces*.

We follow an exploratory approach to work towards semantic contextual recommendations [1] and structured semantic search [6]. A sole recommendation system with no user refinement can be insufficient. We therefore will adapt it towards an improved navigation which presents suggestions, but is open to user refinement [5]. Furthermore, there are different categories of recommendations based on different information needs. These navigational categories from different perspectives could include:

- hierarchically related documents
- related or similar projects, workflows, and tasks
- project overviews
- documents accessed by colleagues in similar tasks
- workflows used by colleagues
- similar parts of a similar project

[1] `http://www.linkeddesign.eu/`

[2] The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n^{o}284613.

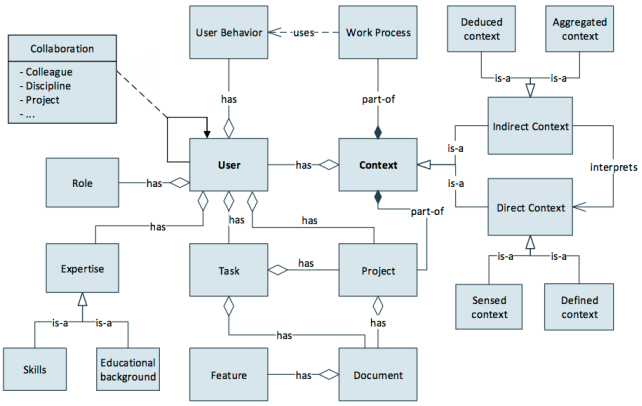


Figure 1: Conceptual model of available context

- related parts in similar projects
- specializations / generalizations
- similar type of document (project documents, module documentation, design drawings, lessons learned, best practices, etc.)
- different organizational perspectives (engineering, management, client relations, controlling, ...)

To understand and retrieve these relations, we need to understand the context and tasks [4]. Some domain-specific context features are already available for the user context, work-task [2], and project context, and context from the knowledge base and the individual document. We can further break these down into links and relations between documents or metadata describing, e.g., conceptual level, level of detail, phase of the lifecycle, project environment, type of document, and the more commonly used metadata such as author, date, topic, etc. Selected context features are shown in Fig. 1. These will be complemented with content-based methods and with data gathered from user behaviour.

Usually, recommendations are generated by inferring relations between documents based on users' interaction with them, with common challenges regarding the level of uncertainty for their results. In our case, a major challenge is estimating which of the mentioned social circles contribute to what extent to the search tasks and how they can be made to work in her favour by supplying the right information by direct recommendation, inform the ranking when she is searching for information, of help her to better filter and manage data, documents, or knowledge objects. Another challenge is that personalisation goals can change a lot during a work task, as she can take on different roles. Finally, there is the problem of sparse and insufficient data, as we have a much smaller number of documents, users, and interactions than in large-scale systems, which can make statistical approaches biased or wholly inapplicable [8]. Using conventional approaches, this would put us into a permanent cold start condition.

We therefore exploit the rich domain-specific context we get from the scenario to better focus the recommendation. In first feasibility analyses, we take hints from the literature [1, 6, 8, 2, 3, 9] and include additional context features lead to the conceptual context model in the engineering environment shown in Fig. 1. The context model is work in progress and will be refined with context features of users and documents as well as document metadata and content taken from a reference ontology developed in parallel.

The personalisation is still informed by the interaction of the users with the document database, but the context is used to offset the sparsity of interaction data. One exemplary task is to retrieve and understand the design decisions leading to a certain set of rules for structural components. The system will enable users to not only search for similar structures or access relevant documentation, but also for cases where similar problems had to be solved, which can give hints towards high-level alternatives. It might also be possible to learn certain workflows or best practices from other engineers. This is complicated by the fact that the more experienced an engineer is, the less they need to access related documents. This is one of the questions we aim to explore in interviews and later from the live system as part of the evaluation.

3. CONCLUSION

We have presented our initial work towards the integration of KBE and Recommendation Systems in a domain-specific and context-rich application scenario. The domain is different from those examined in the literature which means that we will have to heavily adapt and refine existing solutions as well as develop tailored methods. Our goal is to use the described context information of the users and their connections, the organisation, and the documents space to enrich, support, and improve workflows in the manufacturing engineering domain. We will further extend this preliminary work towards a better understanding of the design workflows and information needs [10] and an identification of those parts that would most benefit from personalised recommendations and navigation.

4. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. In *Recommender Systems Handbook*, Springer, 2011.

[2] P. Brusilovsky and D. W. Cooper. Domain, task, and user models for an adaptive hypermedia performance support system. In *IUI '02*, 2002.

[3] T. Gu, X. H. Wang, H. K. Pung, and D. Q. Zhang. An ontology-based context model in intelligent environments. In *CNDS'04*, 2004.

[4] P. Ingwersen and K. Järvelin. *The Turn: Integration of Information Seeking and Retrieval in Context*, 2005.

[5] M. Mehrpoor, A. Gjærde, and O. I. Sivertsen. Intelligent services: A semantic recommender system for knowledge representation in industry. In *ICE 2014*.

[6] P. Petcu and R. Dragusin. Considerations for the Development of Task-Based Search Engines. In *Integrating IR technologies for Professional Search Workshop @ ECIR2013*, 2013.

[7] G. L. Rocca. Knowledge based engineering: Between AI and CAD. *Adv Eng Inform*, 26(2), 2012.

[8] J. Tait. Issues and Non-Issues in Professional Search. In *Integrating IR technologies for Professional Search Workshop @ ECIR2013*, 2013.

[9] M. Wieland, O. Kopp, D. Nicklas, and F. Leymann. Towards context-aware workflows. In *CAiSE*, 2007.

[10] Wilson and T. D. Human information behavior. *Informing Science*, 3(2):49–56, 2000.

Visualizing Student Participation in a Collaborative Learning Environment

Jordan Barría
Universidad Austral de Chile
General Lagos 2086, Valdivia
jordanbarriap@gmail.com

Eliana Scheihing
Universidad Austral de Chile
General Lagos 2086, Valdivia
escheihi@uach.cl

Denis Parra
PUC Chile
Vicuña Mackenna 4860
dparra@ing.puc.cl

ABSTRACT

In the search for techniques that support participation in online communities, in this poster we present a visualization tool for a collaborative learning environment which aims at motivating students to engage in online discussions taking place during learning activities. Grounded on social comparison theory, we propose a graph-based visualization that shows communication patterns between users or teams, in a way that it can increase social awareness and enable social comparison about students' level of contribution. This work is in its design phase, so we present the supporting hypotheses of our proposal, expecting to encourage discussion and user feedback in order to proceed with the coming step of conducting a user study over a period of several months.

Categories and Subject Descriptors

[**Applied Computing**]: Education — *Collaborative Learning*; [**Human-centered Computing**]: Visualization—*Visualization design and evaluation methods*

Keywords

Social visualization, Social comparison, Collaborative learning

1. INTRODUCTION

Promoting participation in online communities is an active research field. From early research [6] to most recent works [8], it has been shown that engaging people to participate is not a simple task, since usually a small proportion contribute actively and the rest of the users, the largest proportion, become *lurkers* who contribute very little or nothing. There are a few existent works on promoting user participation within online learning communities by means of information visualization [9, 3, 5] to reflect students' progress as well as their contribution to the learning activities. This implies that users' participation data are represented in an informative way being accessible to every community member, establishing a social visualization. Social comparison theory [2] is usually cited in support of the successful outcome of these approaches [9, 3], which states that people tend to compare their achievements with people who they think are similar to them, leading to an improved performance. Similarly, in [5] social comparison is mentioned as a result of enabling group awareness, but it focuses on studying the positive influence of raising student participation awareness on their collaborative behavior through a visualization tool. Considering the aforementioned works, we propose the implementation of a social visualization tool focused on increasing awareness on classmates and teachers participating in a b-learning community supported by the Kelluwen platform. It will emphasize graphically aspects that we hope can act as feedback and activate social comparison among users.

In the rest of this document we provide some context by introducing the Kelluwen learning platform where this tool will be implemented, then we provide details of the visualization design, to finally state our conclusions and expectations on the future work.

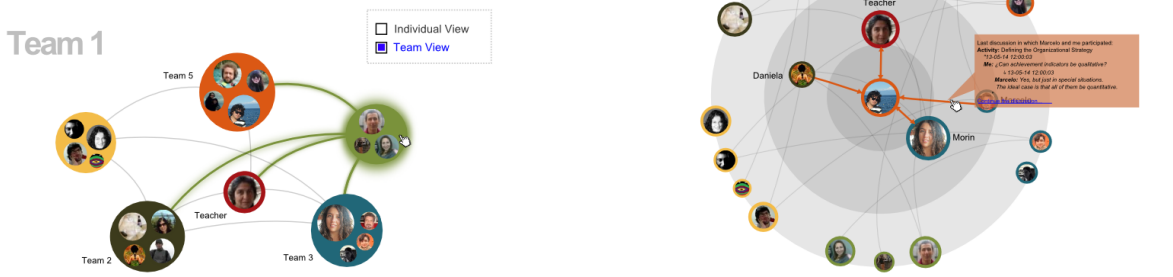
2. KELLUWEN PROJECT

Kelluwen is a community of students, teachers and researchers focused on building, using and sharing collaborative didactic designs that combine traditional classroom activities and the use of social web tools (the Web 2.0) as didactic resources. This project is supported by a Web platform, which has been used by a large a group of vulnerable schools of Southern Chile. The experience we have gained in a couple of years (2010-2012) shows that the pace of students' participation in discussion learning activities decreases remarkably as the weeks go by [1]. This participation is reflected by posting messages, replying those messages and 'liking' them through the Virtual Worklog tool[4]. Another issue identified was the evident lesser rate of interaction established between students belonging to *twin classes* (geographically remote classes who execute the same didactic design at the same time period) compared students in the same class, at the expense of the benefits provided by this source of feedback and opinion sharing. Therefore, we seek for an strategy that increases students' participation in order to support our main project goal: improving socio-communicative skills in students through the use of ICT in their classes [1].

3. VISUALIZATION DESIGN

The proposed visualization (Figure 1) aims to represent the students' participation while discussing through the Virtual Worklog tool. It will be accessible to all users who are part of the same didactic design execution –i.e., students that belong to the same and *twin classes*. Each user is depicted as a circular node with her profile photography inside, where its size is proportional to her level of participation along learning activities. As previously said, it is given by the number of written messages in the Virtual Worklog and the number of 'likes' to peers' messages.

Since learning activities are carried out by means of collaborative group-work, users can identify teams – defined by the teacher who supervises the activities– by the border color of each node since each color represents a specific group. Moreover, users can change the view in which nodes are deployed in order to see them clustered into groups, thus allowing a team view of their participation (Figure 1a) instead of the individual view (Figure 1b). Furthermore, users



(a) *Class perspective* from *team view*

(b) *Self perspective* from *individual view*

Figure 1: The two perspectives (class and self-centered) of visualization from team and individual view respectively. Users can change from team view to individual view (and viceversa) by using the checkbox at the right top corner of the tool (a and b). Users can switch class to self perspective clicking on an specific user node (a), and from self to class perspective clicing outside radial layout displayed (b)

can choose the perspective that summarizes the users' participation: the general *class perspective* (Figure 1a) and the personalized *self perspective* (Figure 1b), which are described next.

3.1 Class Perspective

Through the *class perspective*, users access a holistic view of the participation of all their peers on discussions taken place during learning activities (Figure 1a), providing a high-level representation of the class. Here, user nodes are deployed as a social network, where undirected edges connect pairs of nodes if those users had interacted by either replying a message or 'liking' one. The information presented here is very general, since the layout position of a node within a social network can't describe completely the real closeness between users. For that reason, if a user wants to access to a detailed perspective of her own participation data or one of her peers, she can access to the self-centered perspective.

3.2 Self Perspective

By choosing the *self perspective* (Figure 1b), users can access detailed data about relationships established among a specific user and all of her peers. Here, the selected user node is depicted at the center of a radial layout, while the other nodes are deployed surrounding it. Like in the *class perspective*, interaction between users is represented by a connecting edge, a directed one: unidirectional when the proportion of interaction from user A to B is higher than from B to A (using a certain threshold); or bidirectional in case that both proportions of interaction be equitably distributed. Finally, the distance that separates peripheral peer nodes from the center user node reflects the frequency of interaction both established over learning activities. Therefore, the more a pair of users interact over time, the closer they will be located.

We hypothesize that the implementation of the proposed social visualization tool in Kelluwen Web platform can raise social awareness and perhaps activate social comparison in order to stimulate users to engage in an active reciprocal behavior.

4. CONCLUSIONS AND FUTURE WORK

The contribution of this work is given proposing a visualization supported by a survey of successful experiences about social visualizations within CSCL environments. We will apply our approach in a different cultural context, since existent studies were not applied on teenage students under social risk. Moreover, though visual social network representation has been applied in the analysis of online learning communities [7], we include different perspectives to represent users interaction which can be accessed through the same visualization in a dynamic way. Finally, we will study the effect of this approach on students' participation rather than its usefulness on teachers.

The next phase we pursue is the implementation of this visualization tool on the Kelluwen platform and the design of the experiment that will assess its impact on the overall class behavior. The experiment will consist of incorporating this visualization to certain specific classes, measuring the peer interaction reached throughout the activities in a sample of classrooms having and not having access to the visualization. We also want to explore which view –individual or team– and which perspective –class or self-centered– is perceived as simpler to understand. We are interested in telling whether these perspectives are complementary and enrich social navigation or whether users clearly prefer one over the other to explore their participation.

5. REFERENCES

[1] Luis Cárcamo, Eliana Scheihing, and Camila Cárdenas. *Didáctica 2.0. La Web Social en el aula*. Ediciones Kelluwen, 2013.

[2] Leon Festinger. A Theory of Social Comparison Processes. *Human Relations*, 7(2):117–140, May 1954.

[3] I-Han Hsiao, Julio Guerra, Denis Parra, Fedor Bakalov, Birgitta König-Ries, and Peter Brusilovsky. Comparative social visualization for personalized e-learning. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pages 303–307. ACM, 2012.

[4] Katherine Inalef, Julio Guerra, and Eliana Scheihing. Development and Validation of a Virtual Worklog as a Collaboration Tool for the Kelluwen Learning Community. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*, pages 1936–1941. Ieee, June 2012.

[5] Jeroen Janssen, Gijsbert Erkens, and P.A. Kirschner. Group awareness tools: It's what you do with it that matters. *Computers in Human Behavior*, 27(3):1046–1058, 2011.

[6] Jenny Preece, Blair Nonnecke, and Dorine Andrews. The top five reasons for lurking: improving community experiences for everyone. *Computers in Human Behavior*, 20(2):201–223, March 2004.

[7] Reihaneh Rabbany, Mansoureh Takaffoli, and Osmar Zaïane. Social Network Analysis and Mining to Support the Assessment of On-line Student Participation. *ACM SIGKDD Explorations Newsletter*, (2):20–29.

[8] Julita Vassileva. Motivating participation in social computing applications: a user modeling perspective. *User Modeling and User-Adapted Interaction*, 22(1-2):177–201, March 2012.

[9] Julita Vassileva and Lingling Sun. Using community visualization to stimulate participation in online communities. *E-Service Journal*, pages 1–32, 2007.

Estimating users' areas of research by publications and profiles on social networks

Petr Saloun
VSB-Technical University of Ostrava, 17. listopadu 15
70833 Ostrava
Czech Republic
petr.saloun@vsb.cz

Adam Ondrejka
VSB-Technical University of Ostrava, 17. listopadu 15
70833 Ostrava
Czech Republic
adam.ondrejka.st@vsb.cz

Ivan Zelinka
VSB-Technical University of Ostrava, 17. listopadu 15
70833 Ostrava
Czech Republic
ivan.zelinka@vsb.cz

ABSTRACT

We focus on estimating a research area of a researcher/user by finding a unique identity in digital libraries and social networks and by analyse of public metadata of their publications and published information on social networks profiles. The lack of content of the metadata in some of the publications is solved by the information retrieval using techniques of NLP. We estimate the author's domain by extracting keywords from abstracts as well as by information published on social profiles. The result of this work is a design, an original algorithm and experimental verification of the algorithm.

Categories and Subject Descriptors

H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia—*User issues*; H.3.7 [**Information Interfaces and Presentation**]: Digital Libraries—*User issues*

General Terms

Design

Keywords

digital library, identify user, social media, information retrieval, natural language processing

1. INTRODUCTION

There are situations in life when we need to find works of a specific researcher, for example when we organize a conference. One of the most common way to solve this problem is to search for the information about this researcher, either by looking at the institutions and his publications or by examining the topics he had on various conferences, and then create a profile of the researcher manually. With the boom of social networking people began to publish more openly accessible data than before. Using the data may reveal an interesting complement to the true identity of a person. Unfortunately, the expansion and the emergence of various social networks caused a relatively large fragmentation and users publish specific information about themselves to a social network focusing on the specific topic. The fact that people can have the same name is another obstacle, therefore it is necessary to verify that it is actually a profile of the right person and not of his namesake. The main objective of this work is to identify researchers on social networks and digital libraries. Based on the public information on these sites, we estimate the area of a person's research. The results are keywords that serve both as a description of the person and as an input for further research in finding suitable reviewers of publications presented at conferences and for detecting the violations of a copyright.

2. ESTIMATING AREA OF AUTHOR'S RESEARCH

To find the right profiles we used a technique which compares specific attributes by different weights. Details are described in [2]. We used a modified version shown in the Equation 1 (similar work is mentioned in [3]).

$$sim_{u,p} = \begin{cases} \sum_{i=0}^{n} w_i \cdot sim(a_{i,u}, a_{i,p}) & \text{if } sim(a_{name}) > th_{name} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $sim(a_{name})$ is similarity between author and user profile names, th_{name} is threshold value to decide if names are the same or not, n is count of compared attributes, w_i is weight of compared attributes, a_p is set of user's profile attributes, a_u is set of user's attributes by his publications, $sim(a_{i,u}, a_{i,p})$ is similarity between attributes. The text comparison is done by fuzzy matching to include potential typing errors in attributes.

As shown in the Algorithm 1 the input is the name of the researcher. Then the search requests to all the digital libraries are executed and it downloads the publications. Each publication is then categorized by the defined criteria. Initially we eliminated all the articles that were similar or equal and were occurring in multiple libraries. Then we categorize the publications by affiliations using the text similarity algorithm and also by their co-authors. Now we have groups of possible unique authors. There is an issue now with the author publishing on his own or being active in multiple affiliations, because then the algorithm divides him into more groups. To handle the situation, we included a comparison with user's connections retrieved from social networks and additional information about skills, experiences and so on. After that we categorized the keywords by social profile similarity. We found all the profiles associated with the researcher name. Then we tried to find common connections and affiliations, and if there were at least one in each pair we would assign them together with the compared social profiles. The process was repeated for every found co-authors and referred publications with the input of the previously

found authors, so the results would be more accurate. People with the same name are not merged into one identity, because of the classification by connections and affiliates. It is highly unlikely that these people will have the same co-authors, friends and jobs. More information about a unique user identity is described in [1].

The research domain is obtained by analysing the keywords of all found publications and by extracting the additional information from the social profiles. Because of lacking and incompletely chosen keywords in many publications we had to use our original technique to get additional keywords from abstract. We do not go into detail describing our original technique, because of the page limit of this poster.

```
Data: Author's first name and last name
Result: User's identities
firstName, lastName ← {user raw input};
for searcher in DigitalLibrariesSearchers do
  | publications ←SearchAuthor(firstName, lastName);
end
GroupByPublication(publications);
GroupByAffiliates(publications);
for searcher in SocialNetworkSearchers do
  | publications ←SearchAuthor(firstName, lastName);
end
for group in groups do
  | for publication in publications do
  |   | groupKeywords +=
  |   | AnalyzePublication(publication);
  | end
end
finalGroups = GroupBySocialProfiles(groups);
```

Algorithm 1: Finding unique author identity on digital libraries and social networks

3. EXPERIMENT

From the digital libraries we chose IEEExplorer[1], ACM Digital Library[2] and SpringerLink[3]. In this work, the researchers are found on LinkedIn[4] and Researchgate[5] social networks. In the experiment we check if we can find unique identities and research domains of 180 randomly selected researchers. The search of user identities in digital libraries has been tested by at least 180 researchers, by downloading and analysing about 3100 publications (Table 1). The researchers were chosen randomly and included people of different nationalities. Initially there were users grouped only by co-authors and affiliates. There were 118 authors grouped correctly ("R") with rate 65 %. 3 authors had assigned other author's publications ("POA", 2 % error rate) because of fact that searched author had publications with namesake co-authors and it was poorly evaluated as same person, error rate in this case was 59 authors were not merged correctly ("NA"), there were too many created identities of which should be same one author. This was caused by publications with no or one co-author and different affiliations, it was not possible to find connection between them. Error rate of this category was 33 %.

[1]http://ieeexplorer.ieee.org
[2]http://dl.acm.org
[3]http://www.springerlink.com
[4]http://linkedin.com
[5]http://www.researchgate.com

Table 1: Experiment of finding identities

	R	POA	NA	Precision	Recall
Co-authors	118	3	59	97 %	66 %
C-A + Social	132	3	45	98 %	74 %
C-A + S + K	166	14	0	92 %	100 %

In the next step we included comparisons of authors by data found on their social profiles. 132 users were identified correctly (73 %) and to 3 same authors were again assigned wrong publications due to the same reasons, error rate remained 2 %. The only improvements were made in the case when one author was in two different groups ("NA") and when there were connections found in social profiles between them, so error rate decreased to 25 %.Finally we added comparisons by keywords between publications with a single author and publications with multiple authors. 166 users were identified correctly, correct rate increased to 92 %. There was no situation with a one author in more groups ("NA", error rate of this category decreased to 0 %). Unfortunately 14 users had assigned wrong publications ("POA"), error rate increased to 8 %. It was caused by errors in extracting of keywords and the associated bad detection of a similarity between researchers and publications.

4. CONCLUSION

The goal of our work was to create algorithm to estimate research area of users by finding their identities in digital library and social networks and by analyse found data. As the results from our experiment show, the algorithm for identifying research identities on digital libraries and social networks was successful in 92 % of all the attempts in final. This work was the first step in the research of recommending publications to authors and finding violations of copyrights. We would want to try to add comparing authors' domains detected from publications and information on the Internet to classic full-text search approach. This work is input for further research in finding suitable reviewers of publications presented at conferences and for detecting the violations of a copyright.

5. ACKNOWLEDGMENT

The following grant is acknowledged for the financial support provided for this research: Grant of SGS No. SP2014/42, VSB - Technical University of Ostrava, Czech Republic.

6. REFERENCES

[1] K. Kostkova, M. Barla, and M. Bielikova. Social relationships as a means for identifying an individual in large information spaces. In M. Bramer, editor, *Artificial Intelligence in Theory and Practice III*, volume 331 of *IFIP Advances in Information and Communication Technology*, pages 35–44. Springer Berlin Heidelberg, 2010.

[2] E. Raad, R. Chbeir, and A. Dipanda. User profile matching in social networks. In *Network-Based Information Systems (NBiS), 2010 13th International Conference on*, pages 297–304, Sept 2010.

[3] J. Vosecky, D. Hong, and V. Y. Shen. User identification across social networks using the web profile and friend network. *IJWA*, 2(1):23–34, 2010.

What about Interpreting Features in Matrix Factorization-based Recommender Systems as Users?

Marharyta Aleksandrova
Université de Lorraine - LORIA, France
NTUU "KPI", Ukraine
firstname.lastname@loria.fr

Armelle Brun
Université de Lorraine - LORIA
Campus Scientifique
54506 Vandoeuvre les Nancy, France
firstname.lastname@loria.fr

Anne Boyer
Université de Lorraine - LORIA
Campus Scientifique
54506 Vandoeuvre les Nancy, France
firstname.lastname@loria.fr

Oleg Chertov
NTUU "KPI",
37, Prospect Peremohy,
03056, Kyiv, Ukraine
chertov@i.ua

ABSTRACT

Matrix factorization (MF) is a powerful approach used in recommender systems. One main drawback of MF is the difficulty to interpret the automatically formed features. Following the intuition that the relation between users and items can be expressed through a reduced set of users, referred to as representative users, we propose a simple modification of a traditional MF algorithm, that forms a set of features corresponding to these representative users. On one state of the art dataset, we show that the proposed representative users-based non-negative matrix factorization (RU-NMF) discovers interpretable features, while slightly (in some cases insignificantly) decreasing the accuracy.

Keywords

Recommender systems, matrix factorization, features interpretation.

1. INTRODUCTION, RELATED WORKS

Recommender systems aim to estimate ratings of target users on previously non-seen items. One of the methods used for this task is matrix factorization (MF), which relies on the idea that there is a small number of latent factors (features) that underly the interactions between users and items [1]. Let M be the number of users and N the number of items. The interaction between these entities is usually represented under the form of a matrix R with element r_{mn} corresponding to the rating assigned by the user m to the item n. MF techniques decompose the original rating matrix R into two low-rank matrices U ($\dim(U) = K \times M$) and V ($\dim(V) = K \times N$) in such a way that the product of these matrices approximates the original rating matrix $R \approx R^* = U^T V$. The set of K factors can be seen as a joint latent space on which a mapping of both users and items spaces is performed [1]. Features resulting from factorization usually do not have any physical sense, what makes resulting recommendations unexplainable. Some works [2, 3] made attempts to interpret them by using non-negative matrix factorization with multiplicative update rules (for simplicity, further referred to as NMF). However, the proposed interpretation is not so easy to perform as it has to be discovered manually. Based on the assumption that the preferences between users are correlated, we assume that within the entire set of users, there is a small set of users that have a specific role or have specific preferences. These users can be considered as representative of the entire population and we intend to discover features from MF that are associated with these representative users.

2. THE PROPOSED APPROACH: RU-NMF

Let us consider 2 linear spaces L_1 and L_2 of dimensionality respectively 6 and 3, with basic vectors in canonical form $\{\vec{u}_m\}$, $m \in \overline{1,6}$ and $\{\vec{f}_k\}$, $k \in \overline{1,3}$. Let the transfer matrix from L_1 to L_2 be specified by matrix (1). Then $\vec{u}_5$, $\vec{u}_1$ and $\vec{u}_2$ are direct preimages of $\vec{f}_1$, $\vec{f}_2$ and $\vec{f}_3$ respectively, indeed, $P\vec{u}_5 = \vec{f}_3$. At the same time vectors $\vec{u}_3$, $\vec{u}_4$ and $\vec{u}_6$ will be mapped into linear combinations of basic vectors $\vec{f}_1$, $\vec{f}_2$, $\vec{f}_3$.

$$P = \begin{pmatrix} 0 & 0 & p_{13} & p_{14} & 1 & p_{16} \\ 1 & 0 & p_{23} & p_{24} & 0 & p_{26} \\ 0 & 1 & p_{33} & p_{34} & 0 & p_{36} \end{pmatrix} \quad (1)$$

Matrix U can be considered as a transfer matrix from the space of users to the space of features. Analyzing the example considered above, we can say that if matrix U has a form similar to (1), *i.e.* U has exactly K unitary columns

with one non-zero and equal to 1 element on different positions, then the users corresponding to these columns are direct preimages of the K features. The features can thus be directly interpreted as users. These users will be referred to as representative users. In order to force matrix U to satisfy the imposed conditions we propose the RU-NMF approach, that consists of 6 steps, further detailed below.

Step 1. A traditional matrix factorization is performed. Following [2, 3], NMF is used.

Step 2. A normalization of each of the M column vectors of the matrix U is performed so as to result in unitary columns. The resulting normalized matrix is denoted by U_{norm} and the vector of normalization coefficients by C.

Step 3. This step is dedicated to the identification of the representative users in the U_{norm} matrix. A user u_m is considered as the best preimage candidate (representative user) for the feature f_k if the vector u_m^{norm} is the closest to the corresponding canonical vector (a vector with the only one non-zero and equal to 1 value on the position k). The notion of closeness between vectors is expressed in Euclidean distance. Once all representative users are identified, the matrix U_{norm} is modified so as to obtain a matrix in a form of (1): lines, corresponding to the representative users, are replaced with appropriate canonic vectors. The resulting modified matrix is denoted by U_{norm}^{mod}.

Step 4. Each column of the matrix U_{norm}^{mod} is multiplied by the appropriate normalization coefficient from the set C resulting in matrix U^{mod}. After this, representative users will remain preimages of the features but with scaling factors.

Step 5. In order to obtain the best model we also have to modify the matrix V. The modification of V can be performed using optimization methods with the starting value obtained during the first step. As the objective of this paper is to determine the relevance of finding preimages of the features and to quantify the decrease in the quality of the recommendations, we did not consider this step.

Step 6. The resulting recommendation model is made up of matrices U^{mod} and V ($R^* = \left(U^{mod}\right)^T V$).

3. EXPERIMENTAL RESULTS

Experiments are performed on the 100k MovieLens dataset[1], with 80% of ratings used for learning the model and 20% for testing it. The accuracy is evaluated with two classical measures: mean absolute error (MAE) and root mean square error (RMSE). The goal of the experiments is to compare the accuracies of RU-NMF and NMF. For these reasons we compute the accuracy loss $\rho = \frac{err(RU\text{-}NMF) - err(NMF)}{err(NMF)} 100\%$ for factorizations with 10, 15 and 20 features on 30 different samples. Results are presented in Table 1. A positive loss means that NMF performs better than RU-NMF. In the worst case the accuracy loss equals to 6.64%, for RMSE with 20 features, which is quite small. The lowest average accuracy loss (0.05%) is obtained with 10 features for both errors. When comparing the accuracy loss between test and learning sets, we can note that the average loss is 3 times lower on test than on learning, for both errors and for all the number of features: thus we can say that RU-NMF has a lower relative loss between learn and test compared to NMF. A thorough analysis of the losses obtained on the 30 samples has shown that the accuracy loss on the test set is lower than the one on the learning set in all cases. In some runs, RU-NMF has even a higher accuracy than NMF (Table 1, values in gray shadow).

[1] http://grouplens.org/datasets/movielens/

Table 1: Accuracy loss ρ between RU-NMF and the traditional NMF, for 10, 15 and 20 features.

	Learning set		Test set	
	MAE	RMSE	MAE	RMSE
10 features				
mean	**0.17%**	**0.19%**	**0.05%**	**0.05%**
min	0.03%	0.03%	-0.06%	-0.07%
max	0.38%	0.46%	0.18%	0.20%
15 features				
mean	**0.98%**	**1.29%**	**0.29%**	**0.33%**
min	0.49%	0.61%	-0.06%	-0.04%
max	1.71%	2.38%	0.77%	0.79%
20 features				
mean	**2.78%**	**4.08%**	**0.70%**	**0.82%**
min	1.38%	1.94%	0.13%	0.12%
max	4.27%	6.64%	1.43%	1.53%

4. DISCUSSIONS AND FUTURE WORK

The analysis of the accuracy loss between RU-NMF and traditional NMF has shown that prediction error rises slightly (in some cases insignificantly) with RU-NMF. However the features formed with this approach consistently disturb the accuracy on the test set less than on the learning one. This can be considered as a potential ability of factorization techniques with features related to reality to form better searched predictions. The proposed approach also lets us easily explain the resulting recommendations. Indeed, each user of the population is linearly mapped on the basis related to representative users (through matrix U) and the preferences of the latter ones (expressed by matrix V) are used to estimate the ratings of the whole population. In a future work, we would like to focus first of all on the verifications of the hypothesis that users associated with the features can be really considered as representative ones. We believe that this can be done while solving the new item cold-start problem with ratings of the representative users on new items used to estimate ratings of all the population on these items.

5. REFERENCES

[1] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[2] S. Zhang, W. Wang, J. Ford, and F. Makedon, "Learning from incomplete ratings using non-negative matrix factorization." in *Proc. of the 6th SIAM Conf. on Data Mining*, 2006.

[3] J.-F. Pessiot, V. Truong, N. Usunier, M. Amini, and P. Gallinari, "Factorisation en matrices non-négatives pour le filtrage collaboratif," in *3rd Conf. en Recherche d'Information et Applications*, 2006, pp. 315–326.