Creating Usage Context-Based Object Similarities to Boost Recommender Systems in Technology Enhanced Learning

Katja Niemann and Martin Wolpers

Abstract—In this paper, we introduce a new way of detecting semantic similarities between learning objects by analysing their usage in web portals. Our approach relies on the usage-based relations between the objects themselves rather then on the content of the learning objects or on the relations between users and learning objects. We then take this new similarity measure to enhance existing recommendation approaches for the use in technology enhanced learning.

Index Terms—Recommender systems, web mining, learning web portals, technology enhanced learning, learning analytics

1 Introduction

In recent years, recommender systems have become popular in the technology enhanced learning (TEL) domain to identify suitable learning objects for users that are for example offered by learning web portals. Such recommender systems often require extensive additional information about the learning objects, e.g. the competencies/knowledge they impart and information about the users, e.g. existing knowledge and learning goals [1]. Such additional information is often not available and expensive to create, thus, we need new ways to gather new information about users and items that is implicitly given in their behaviour and usage, respectively.

The special interest group dataTEL[1] (data-driven research and learning analytics) was created to increase research on educational data sets and make educational systems more transparent and predictable. Several educational institutions are already part of the dataTEL group and provided data sets collected in educational settings. In a first study of these data sets, Verbert et al. [2] found that a main challenge to be tackled is the sparsity of the available data. Therefore, they conclude that further research on implicit relevance indicators and similarity measures is required to compensate the lack of data and enable the finding of relevant items and/or users.

In this paper, we present a new way of detecting similarities between learning objects by considering their usage contexts, i.e. the learning objects they most significantly often co-occur with in the same user sessions and evaluate the approach on two data sets submitted to dataTEL, i.e. the usage data collected in the MACE[2] and in the Travel well[3] web portals. We claim that usage context-based similarity gives rise to content similarity and can thus be used for recommendations. This way, we are able to compensate the lack of semantic information for the learning objects as well as sparsity of the rating data.

Other approaches that measure the similarity of data objects based on their usage and, thus, might appear similar at a first glance are item-based collaborative filtering (CF) and association mining. However, item-based collaborative filtering [3] only takes the objects' users and their ratings into account and not the specific sessions in which the objects were used. Furthermore, our approach differs from association mining [4] in that we do not assume two items to be related if they co-occur with each other, but if they significantly often co-occur with the same items. Thus, in contrast to association mining, two items can be highly related even if they were never used together.

The paper is structured as follows: In Section 2, we give an overview of existing recommendation strategies and in Section 3, we describe the principles of the usage context-based approach we propose. In Section 4, we present the data sets gathered from the educational web portals MACE and Travel well that are used in the experimental evaluations presented in Section 5 (usage context-based similarity versus content-based similarity) and Section 6 (usage context-based similarity for enhancing existing recommender systems). In Section 7, we give a conclusion and present ideas for further work.

2 Related Work: Recommendation Strategies

2.1 Content-Based Filtering

Content-based systems use the items' attributes and the users' preferences for recommendations. Item profiles can

1. http://ea-tel.eu/sig-datatel/

2. http://mace-project.eu/
3. http://lreforschools.eun.org/web/guest/travelwell-all

• The authors are with the Fraunhofer Institute for Applied Information Technology FIT, Sankt Augustin, Germany.
E-mail: {katja.niemann, martin.wolpers}@fit.fraunhofer.de.

Manuscript received 21 July 2014; revised 14 Nov. 2014; accepted 25 Nov. 2014. Date of publication 8 Dec. 2014; date of current version 14 Sept. 2015.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TLT.2014.2379261

be created automatically, e.g. through keyword extraction for text documents, or manually, e.g. for restaurants holding attributes like style and location. User profiles can be built explicitly by asking the users about their interests or implicitly by analysing the users' given ratings. For the creation of recommendations, the user profiles are matched against the item profiles and the most suitable unknown items are recommended. Several systems are developed that use content-based filtering (CBF) to help users find information, e.g. PRES (personalised recommender system) [5] and ITR (item recommender) [6]. Since content-based recommender systems only exploit the user profiles of the active users and compare them to the items' profiles to calculate recommendations, it is independent from the completeness of other user profiles. Additionally, when a new item is added to the database, its content is analysed and based thereupon it can directly be recommended to a user and does not need to be rated by other users first. However, recommender systems relying on content-based filtering suffer from problems like the new user problem and overspecialisation. Additionally, it can be time-consuming and expensive to maintain the item profiles.

2.2 Collaborative Filtering

2.2.1 Neighbourhood-Based Filtering

Systems based on collaborative filtering do not consider the items' attributes but make use of the users' ratings on items that can be explicit (e.g. rate a book with three stars) or implicit (e.g. visit a site, listen to a song). The neighbourhood-based CF approaches comprise user-based and item-based techniques. In user-based CF (UBCF), each user is represented by a vector holding her rated items [3]. In order to compare two users, the similarity of their representing vectors is calculated, e.g. by using the cosine similarity. Finally, a user gets recommendations based on the ratings of the users that are most similar to her. A prominent example for such a system is MovieLens.[4] Item-based CF (IBCF) approaches do not compare users but calculate the similarity of items by comparing their users' implicit and explicit ratings [7]. A system relying on this approach is Amazon.com in which products often bought by the same users get a higher similarity value than products that do not share so many users [8]. Advantages of neighbourhood-based CF approaches are that no semantic information is needed to create item profiles, cross-genre niches can be identified, and the subjectively felt quality of the items is incorporated in the evaluation process. However, user and item profiles first need to evolve before sufficient recommendations can be produced.

2.2.2 Matrix Factorisation

Matrix factorisation models map both users and items to a joint latent factor space, i.e., each item is associated with a vector q that measures the extent to which the item bears the factors of the space and each user is associated with a vector p that measures the extent of interest the user has in items that are high on the corresponding factors. The resulting dot product of p and q represents the user's overall interest in the item's characteristics [9]. Matrix factorisation methods have become popular since they combine a high predictive accuracy with good scalability. In recent years, several approaches have been created to deal with the major challenge of matrix factorisation, which is computing the mapping of each item and user to factor vectors (see [10], [11]). Similarly to the neighbourhood-based CF, the more ratings are given for users and items, the more accurate are the predicted ratings.

4. http://movielens.umn.edu

2.3 Hybrid Systems

Hybrid systems are implemented to exert the advantages from more than one technique while the drawbacks of single techniques can be compensated. Burke [12] describes several approaches to combine recommenders, e.g. Ranked Hybrids that combine the ratings from different recommenders using a weighting scheme (see [13], [14], [15]), Switching Hybrids that select the rating prediction from the recommender with the highest confidence value (see [16], [17]), and Feature Augmentation Hybrids in which each contributing recommender adds features to the items' descriptions, so that the actual recommender has a better base (see [18], [19]).

2.4 Recommender Systems in TEL

The creation of recommendations in a TEL scenario partly differs from the typical item recommendation task in that for example the existing knowledge of the user and her goals as well as the knowledge the items aim to impart ideally are taken into account [20]. Hence, a number of recommender systems that focus on TEL scenarios have been introduced in the last decade [21], [22], [23], [24], [25].

A promising approach is for example the use of multi-criteria input where learners and teachers can rate learning objects according to several attributes (e.g. their level of complexity and their curriculum alignment) [20]. Another possibility is the inclusion of the user's context (e.g. location or mood) in the recommendation process [26].

However, in most scenarios as for example in the learning platforms MACE and Travel well considered in this paper, such detailed data is not available. Thus, the approach presented in this paper focusses on utilising the users' knowledge and context, which is inherent in their activities, in order to reveal item relations without forcing the users to explicitly share their knowledge.

3 Usage Context-Based Similarity

3.1 Background of the Notion Usage Context

This paper investigates if usage context-based similarity gives a hint at the content similarity of learning object pairs and if this usage context-based similarity can be utilised to enhance recommender systems. The notion of usage context was inspired by the concept of word contexts successfully applied in linguistics. Words stand in linear orders, e.g. in speech or in written texts. The context of a word can thus easily be defined by the words that occur before and after it. If two words have very similar contexts (e.g. words they often co-occur with in sentences), they are said to be paradigmatically related [27]. For example: In many contexts, the word *car* can be replaced by the word *vehicle*, i.e., they share a similar context containing e.g. the words *driver* and

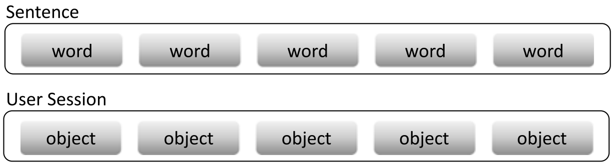

Fig. 1. Analogy of words used in sentences and data objects accessed in user sessions.

highway. Thus, paradigmatic relations lead to semantic relations or, as [28] states, context similarity correlates with content relatedness.

Similar to words used in sentences, data objects are accessed in user sessions, see Fig. 1. Thus, we take up the insight from linguistics and form our hypothesis that the usage context-based similarity of data objects provides an indication for content relatedness. However, the definition of a usage context or user session depends on the conditions. In web mining, a user session comprises all page references made by a user during a single visit to a site [29]. In this paper, the definition of a user session is expanded to hold all events including the event types conducted by a user in one visit. However, a user might leave the site or the portal (which is commonly not logged) and return some minutes or hours later. Thus, it must be defined, when a new visit and, thus, a new user session starts. One possibility is to define a new user session to start after a new log-in of a user or after a pre-defined time in which the user did not conduct any events. The user session definitions applied in this paper are discussed in the sections describing the respective data sets, i.e., Sections 4.1 and 4.2.

Thus, the usage context of an object is the result of the objects accessed before or after it in the same user sessions, i.e., by its (significant) co-occurrences. When comparing two data objects, they are assumed to be semantically related, if their usage contexts' significantly overlap. Please note that two objects can have highly similar usage context, even though they were never used together.

3.2 Defining Significant Co-Occurrences

We define two objects to be co-occurrences if they co-occur in at least one user session. However, not every co-occurrence is statistically significant, rather most co-occurrences are assumed to be coincidental [28]. Basic association measures calculate a significance score by comparing the observed frequency O of a co-occurrence with its expected frequency E, examples are mutual information (MI, equation 1) and z-scores [30]. These simple association measures often give close approximation to the more sophisticated association measures (as described below) and are therefore sufficient for many applications. They also have some limitations as they, for instance, tend to fail when calculating the significance value for an object pair in which one object is very often and the other object is only rarely used [31].

$$\mathrm{MI} = log_2 \frac{O}{E}. \tag{1}$$

In statistical theory, association measures and independence tests are always based on a cross-classification of a set of objects, e.g. using contingency tables. These measures compare the expected and the observed frequencies as well. In contrast to the more simple approaches, they do not only consider the expected co-occurrence frequency of the two objects but compute the expected frequencies for all cells in the contingency table [31]. Table 1 shows the contingency table for the objects i and j which co-occured O_{11} times. Additionally, i was accessed in O_{12} sessions in which j was not accessed, j was accessed in O_{21} sessions in which i was not accessed, and O_{22} session hold neither of these two objects. The expected values for these observed values are E_{11}, E_{12}, E_{21}, and E_{22}, respectively. In order to calculate the value of E_{11} (i.e., the expected number of sessions holding the objects i and j) the number of sessions holding i (i.e., R_1) is multiplied with the number of sessions holding j (i.e., C_1) and then divided by the total number of sessions (i.e., N). The other expected frequencies are calculated analogously.

TABLE 1
Contingency Table

	j	$\neg j$	
i	O_{11}	O_{12}	$=R_1$
$\neg i$	O_{21}	O_{22}	$=R_2$
	$=C_1$	$=C_2$	$=N$

Commonly used association measures that are based on contingency tables are the χ^2-test and log-likelihood (LL) [30]. The χ^2-test adds up the squared z-scores for each cell in the contingency table and puts them in relation to the expected frequencies. Since the normal approximation implicit in the z-scores becomes inaccurate if any of the expected frequencies is small [31], the Yates' continuity correction [32] shown in equation 2 offers a better approximation (corrected χ^2-test or cor-χ^2). Equation 3 shows the log-likelihood measure [33]

$$\text{cor-}\chi^2 = \frac{\left(|O_{11}O_{22} - O_{12}O_{21}| - \frac{N}{2}\right)^2}{R_1 R_2 C_1 C_2} \tag{2}$$

$$\mathrm{LL} = 2\sum_{ij} O_{ij} ln \frac{O_{ij}}{E_{ij}}. \tag{3}$$

After the calculation of the co-occurrences' significance values, the most significant ones must be selected for each object. There are two ways to do so, i.e., by ranking or by using a threshold. Ranking means that the co-occurrences are sorted by their significance values and only the n most significant co-occurrences are selected. When using a threshold, only co-occurrences with a significance value higher than the threshold are selected. However, there is no standard scale of measurement to draw a clear distinction between significant and non-significant co-occurrences [34]. Therefore, the calculation of a suitable n or a suitable threshold (depending on the approach) is an exploratory investigation.

3.3 Object Similarity Calculation

We calculate the similarity for each object pair using the cosine similarity in which each object i is described by a vector V_i that holds the most significant co-occurrences of object i

$$\text{cosine-sim} = \frac{V_i \cdot V_j}{||V_i||\ ||V_j||}. \tag{4}$$

The cosine similarity measures the angle between two vectors, thus, the significance values of the co-occurrences are considered.

3.4 Computational Complexity

The computational complexity of this approach depends mainly on the number of items given by n and the average co-occurrence vector size given by k. First, the co-occurrence vectors must be calculated, i.e., for each co-occurring item pair a significance score must be calculated. This is to say, up to $n * (n-1)/2$ significance scores must be calculated. However, the actual complexity is usually much smaller because the significance values only need to be calculated for those item pairs that were actually used together. Finally, the pair-wise similarities of the n co-occurrence vectors need to be calculated with $n * (n-1)/2$ comparisons each requiring up to k steps. Thus, the overall computational complexity adds up to $O(n^2k)$.

4 Data Sets

4.1 MACE

The MACE[5] (Metadata for Architectural Contents in Europe) project relates digital learning objects about architecture, stored in various repositories, with each other across repository boundaries to enable new ways of finding relevant information [35]. While interacting with the MACE portal, users are monitored and their activities are recorded as CAM (Contextualised Attention Metadata [36]) instances. The event types considered for creating the user sessions are accessing the metadata of a learning object in the MACE portal, e.g. its ratings or user tags and accessing the learning object in its origin repository. All other events that involve learning objects, e.g. tagging and rating, require the access of the learning object's metadata, thus, the learning object is already part of the user session without considering these events. Each CAM instance comprises at least the event type, the identifier of the user who conducted the event, a timestamp, and the identifier of the involved object. The CAM instances used for the evaluation were collected in a period of three years from September 2009 to October 2012. Overall, we considered CAM events for 12,176 learning objects conducted by 620 users in 4,291 user sessions. A user session comprises on average 6.28 distinct learning objects and each learning object is used in 2.18 user sessions on average.

MACE offers users and domain experts the possibility of editing parts of the metadata, e.g. tags and classifications. We use the tags and classifications to create semantic similarities between learning objects as baselines to compare our results to. The tags are free text and can be assigned to learning objects by logged in users. The classifications are defined in a controlled vocabulary consisting of 2,884 terms and can only be set by domain experts. 78.69 percent of the used learning objects hold such additional semantic metadata, 70.8 percent hold tags, 14.83 percent hold classifications, and 8.82 percent hold both. Each tagged learning object holds on average 6.59 tags and each classified learning object holds on average 2.27 classifications. Additionally, logged in users are able to rate learning objects. We use the provided ratings to test and evaluate our recommender system. In total, 230 learning objects were rated by 73 users, each of these learning objects was rated at least once and at maximum four times (on average 1.2 times), and each of the 73 users rated 1-19 learning objects (on average 3.79). This results in a user-item-rating matrix with a sparsity of 98.35 percent.

4.2 Travel Well

The Travel well data set[6] [37] was collected on the learning resource exchange (LRE) portal that makes open educational resources available from more than 20 content providers in Europe and elsewhere. The data set contains information about the rating and tagging behaviour of 98 registered users over a period of six months (August 2008-February 2009). For each user activity, the date, user identifier, object identifier and the tag, respectively the rating is stored. As there is no timestamp but only the date, a user session comprises all activities conducted by a user in one day.

Overall, 14,248 events took place in 255 user sessions in which each user session comprises 55 distinct learning objects on average. Additionally, 79 users tagged 1,838 unique objects with 12,041 tags in total; consequently each object was assigned with 6.5 tags on average. Similarly to MACE, the learning objects can hold classification keywords from a controlled vocabulary additionally to the free text tags. 97.97 percent of the learning objects hold tags or classification, 95.53 percent hold tags, 69.04 percent hold classifications and 66.6 percent hold both. Additionally, 75 users rated 1,838 learning objects, each learning object was rated 1-10 times (on average 1.34 times) and each user rated 1-108 learning objects (on average 29.29) which results in a user-item-rating matrix with a sparsity of 98.17 percent.

5 Experiment 1: Usage Context and Content Similarity

5.1 Methodology

5.1.1 *Creating Usage Context-Based Similarities*

We calculate the usage context-based similarity for all learning object pairs in the MACE and in the Travel well data set, respectively, as described in Section 3. We start with calculating the co-occurrences and their significance values for each object using the association measures mutual information, log-likelihood, and the corrected χ^2-test (cor-χ^2). We use two ways to select the most significant co-occurrences for each object. First, we vary the co-occurrence vector sizes from 1-150 for Travel well and from 1-1,000 for MACE. The vector sizes for MACE get bigger as for Travel well since the MACE data set comprises more learning objects. Second, for each object, we calculate an object-specific threshold by averaging the significance values of all its co-occurrences. We calculate one threshold for each object and not one threshold for all objects, because the significance values vary depending on the times an object was used.

5. The MACE usage data set can be obtained from the authors.

6. A description on how to obtain the Travel well data set can be found at http://www.teleurope.eu/pg/pages/view/50630

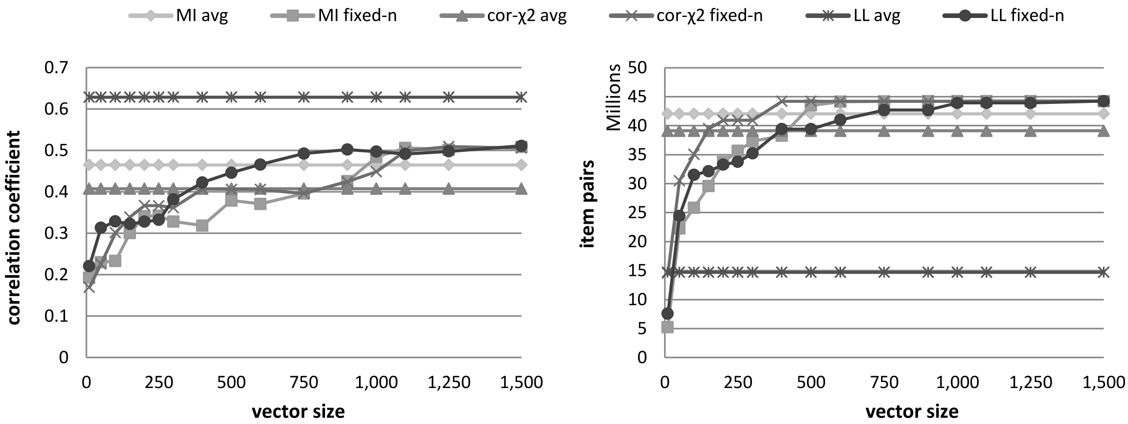

Fig. 2. MACE: a) Pearson correlation coefficient and b) number of considered object pairs.

Thereafter, we calculate the usage context-based similarity of each object pair by comparing their co-occurrence vectors using the cosine similarity.

5.1.2 Creating Semantic Metadata-Based Similarities

We calculate the semantic metadata-based similarity of all object pairs in MACE and Travel well to get a reference value for evaluating the usage context-based similarities. We do so by taking the tags and classifications into account. Since an object cannot be tagged more than once with the same keyword, we create a binary vector for each object and use the Jaccard similarity [38] for calculating the pair-wise semantic metadata-based similarity.

5.1.3 Calculating the Correlation of Both Similarity Measures

We use the semantic metadata-based similarity as tentative *gold standard*. Even though the semantic metadata-based similarity cannot be a perfect representation of the *real* similarity, e.g. because some learning objects only hold one or two tags, we assume it to be a good approximation. In order to prove our hypothesis that usage context similarity implies content similarity, we calculate the Pearson correlation coefficient [39] between the semantic metadata-based and the usage context-based similarity distribution, see equation 5 with X being the set containing all n usage context-based similarities with $\bar{x}$ as mean value and Y being the set containing all metadata-based similarities in the same order with $\bar{y}$ as mean value

$$r_{XY} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}. \quad (5)$$

We only calculate the Pearson correlation coefficient for learning objects that hold at least one semantic metadata-based and one usage context-based similarity to another learning object (not necessarily the same learning object) that is greater than zero. Thus, we exclude learning objects that do not hold a) sufficient amount of semantic metadata and/or b) a sufficient number of significant co-occurrences to establish semantic metadata- and usage context-based relations, respectively, to other learning objects. Even if a learning object holds semantic metadata, it might not be possible to create a semantic metadata-based similarity to another object greater than zero since 72.96 percent of the tags assigned to MACE objects and 73.87 percent of the tags assigned to Travel well objects are unique and cannot be used to compare objects. The same holds true for the usage context-based similarity, the less objects a co-occurrence vector holds, the less is the chance to find similarities to other objects. However, when at least one similarity was found for a learning object, the object is used for the calculation of the Pearson correlation coefficient with all possible object pair combinations.

5.2 Results for MACE

Fig. 2 shows a) the Pearson correlation coefficients for the semantic metadata-based and the usage context-based similarities that are calculated with different association measures (MI, LL, and cor-χ^2) and varying vector sizes (*fixed-n* means that a fixed number of co-occurrences was chosen, *avg* means that an object specific threshold, i.e., its average significance value, was used) as well as b) the number of object pairs that can be considered for each combination of association measure and vector size for the MACE data set.

As could be assumed, the more co-occurrences are considered as significant and, thus, are used to describe an object, the more usage context-based similarities can be detected between object pairs. Additionally, the correlation with the semantic metadata-based similarity increases with the co-occurrence vector size, whereas from a certain vector size on (here: 1,000), the number of similar object pairs found and the correlation gets stable. This is due to the fact, that most objects hold less than 1,000 co-occurrences and are already described to their full extend. With vector size 1,000 (which means at maximum 1,000) the real average vector size is 213 (LL), 255 (MI), and 261 (cor-χ^2). Interestingly, at vector size 1,000, LL performs best in terms of correlation (0.4973), followed by MI (0.4844) and cor-χ^2 (0.4487), which shows that *the more, the better* does not hold true, but the insignificant co-occurrences which can be considered as noise must be filtered to reach sufficient results.

The LL measure with the average threshold performs best in terms of correlation with a value of 0.629. However, it can only calculate similarity values for a subset of the

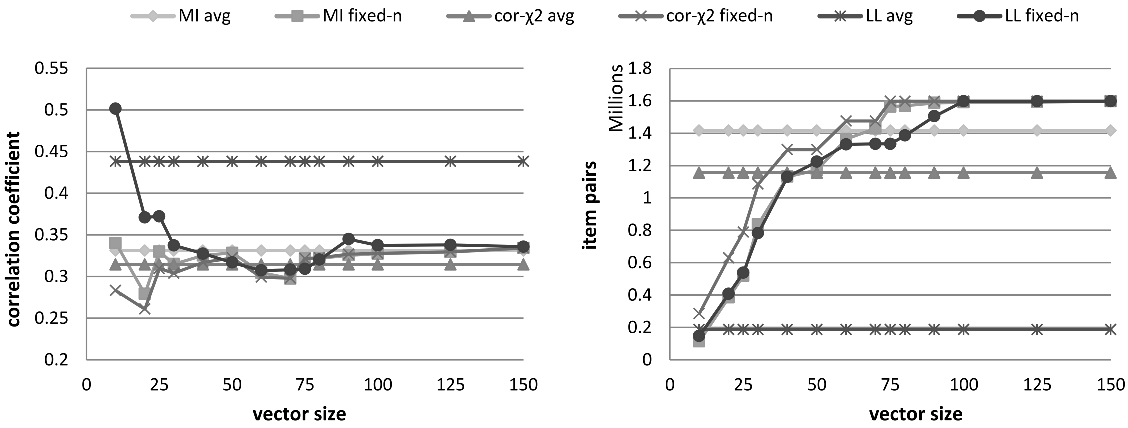

Fig. 3. Travel well: a) Pearson correlation coefficient and b) number of considered object pairs.

available objects (14,707,176 object pairs). When comparing the association measures MI and cor-χ^2 with the average threshold, MI performs significantly better with a correlation value of 0.4653 (compared to 0.4079) and 42,067,378 object pairs (compared to 39,121,435).

5.3 Results for Travel Well

Fig. 3 shows a) the Pearson correlation coefficients for the different association measures and varying vector sizes as well as b) the number of object pairs that can be considered for the calculation of the Pearson correlation coefficient for the Travel well data set. Similarly to MACE, the number of objects for which usage context-based similarities can be established with other learning objects increases with the vector size, but the results stabilise between vector size 100 and 150. This is due to the fact that the Travel well data set only holds 1,925 learning objects whereas the MACE data set holds 12,176 objects.

Apart from the outliers when using very small vector sizes (10-25), the correlation coefficient increases with the vector size. For vector size 150, the association measure LL performs best with a correlation coefficient of 0.3432, followed by MI (0.3324) and cor-χ^2 (0.3235). When using the average threshold, LL performs significantly better than the other measures in terms of correlation (0.4506) but only for a subset of objects (180,300 object pairs). MI outperforms cor-χ^2 with a correlation coefficient of 0.3317 (compared to 0.3227) and 1,326,006 object pairs (compared to 1,140,805).

5.4 Result Interpretation

The resulting correlation coefficients can be described as medium [40]. Since the similarity values do not follow a bivariate normal distribution, no statement can be made concerning the significance of the correlation coefficients. However, because of the large sample of object pairs that is considered, the correlation coefficients can be regarded as a good indicator for an existing relationship between the usage context- and the semantic metadata-based similarities.

A further important point is the fact that the considered metadata can only be interpreted as a shallow content representation. Thus, it is possible that the correlation between semantic metadata- and usage-based similarity represents a lower bound for the real correlation between content and usage context.

For the MACE data set, the reached correlation coefficients are significantly higher than for the Travel well data set; we assume this is due to the fact that the MACE data set holds more detailed usage data, e.g. each learning object access is tracked and not only the metadata provision activities rating and tagging as in Travel well. Additionally, the MACE usage data offer precise timestamps for the collected events, whereas Travel well only provides the date; thus, the user sessions are more accurate in the MACE data set.

The three association measures MI, LL, and cor-χ^2 behave quite similarly compared among each other for the MACE and the Travel well data set. For both data sets, the best performing set-up in terms of correlation coefficient and number of object pairs is LL in combination with a *large* vector size, in which the meaning of *large* must be defined depending on the data set. For MACE and Travel well, a vector size of about 7 percent of the number of distinct objects in the data set is recommended. This set-up is followed by the association measure MI in combination with the average threshold which has the advantage that no parameter for the vector size must be defined, thus, we use this set-up in the following experiment.

5.5 Manual Analysis for MACE

The considered metadata can only be interpreted as a shallow content representation that suffers from the sparsity problem and the resulting similarity values only serve as a tentative *gold standard* for evaluating the usage-based approach. Thus, it is possible that the correlation between the semantic metadata-based and the usage context-based similarity only represents a lower bound for the real correlation between content and usage context. Therefore, it seems reasonable to manually compare a chosen subset of learning objects. If the semantic metadata-based similarity does not sufficiently represent the content, it can be assumed to find a higher congruence between the content of the learning objects and usage-based similarity. In order to do so, the 100 learning object pairs with the highest usage context-based similarities are chosen. Since this manual analysis does not produce explicit similarity values that are comparable, the focus is on finding a content overlap of two learning objects.

For example, two learning objects are assumed to be semantically related if they show different buildings that were designed by the same architect or if they discuss different concepts for fire safety.

Overall, 92 percent of the considered learning object pairs show similarities, 4 percent are not accessible due to permission rights, and only 4 percent show no similarity at all. Many of the checked object pairs show content similarities that are not entailed in the metadata. For text documents, these similarities are in most cases related to the topics such as risk factor analysis or low energy construction. Several object pairs are found that show pictures, e.g. photos, sketches, or models of the same building or construction activity like panel cladding. Often, learning object pairs describe or depict similar buildings in which the similarity is given by different attributes like similar architectural style, building type (e.g. commercial buildings like banks), or the construction system containing the building material. Even though the MACE application profile offers the possibility to store such information, they are most often not contained in the semantic metadata. Furthermore, object pairs are identified that hold a similarity of the displayed content in terms of location and construction date, e.g. pairs which represent web sites containing pictures or articles of different historical buildings in the same town. Additionally, some pairs complement each other, e.g. a lecture and an exercise to the same topic. In one case a pair is found with both objects referring to web sites about graphical algorithms. One of these web sites provides the opportunity to browse through descriptions of existing shape generating algorithms (including pictures), while the other one provides the possibility to create and test such kinds of algorithms. Thus, using an object's usage contexts, content similarities to other objects can be found that are not entailed in the objects' semantic metadata. This also shows that the objects' descriptions are not the (only) reason that certain object pairs exhibit similar usage context profiles. Instead, the users' knowledge about the objects' contents is incorporated in the data analysing process without requiring the users to explicitly share it [41]. It was not possible to conduct a similar analysis for the Travel well data set as most resources were not accessible.

6 EXPERIMENT 2: USAGE CONTEXT-BOOSTED RECOMMENDATIONS

6.1 Methodology

6.1.1 Usage Context-Based Filtering

The usage context-based recommender system computes the expected rating $p(u,i)$ on an object i for a user u by averaging the ratings $r(u,i)$ given by the user to the other objects in her profile $P(u)$ while each rating is weighted by the corresponding similarity $sim(i,j)$, see equation 6

$$p(u,i) = \frac{\sum_{j\in P(u), i\neq j}(sim(i,j) * r(u,j))}{\sum_{j\in P(u), i\neq j}|sim(i,j)|}. \tag{6}$$

This formula is commonly used in item-based collaborative filtering to predict missing ratings. The difference, though, lies in how the similarity of two items is calculated.

6.1.2 Usage Context-Boosting of CF Approaches

Melville et al. [18] introduced content-boosted collaborative filtering, i.e., a feature-augmented hybrid recommender system (see Section 2.4) that uses the given rating history of users and content information of objects to predict the missing ratings in a user-item-rating matrix using a content-based recommender. This enhanced matrix is then used as input for traditional collaborative filtering.

In the previous section we show that the usage context-based similarity gives an indication for the semantic similarity of learning objects pairs. Here, we use the usage context-based recommender to predict the missing ratings in the user-item-rating matrix. This matrix is then used as input for several recommendation approaches which are described in the next section. Thus, the usage context-based boosting is similar to the content-based boosting but does not require any content information which are often not given in a sufficient number for learning objects.

6.1.3 Baseline Recommendation Approaches

In order to create a baseline to evaluate our approach against, we use the neighbourhood-based collaborative filtering methods IBCF (with adjusted cosine similarity) and UBCF (with Pearson correlation based similarity). Additionally, we tested the matrix factorisation methods (MF) offered by the PREA toolkit [42] (i.e., single value decomposition (SVD), Non-negative MF, Probabilistic MF, and Bayesian Probabilistic MF) as well as the MF methods offered by the Java port of the MyMediaLite Recommender System Library [43] (i.e., a standard MF as well as a Biased and a Factorised MF). Based on the performances of the different methods on our test sets and to not overload the diagrams, we choose to present the SVD [44] method from the PREA toolkit and the biased matrix factorisation (BMF) [45] from the MyMediaLite toolkit.

6.1.4 Experimental Set-Up

We perform a five-fold cross evaluation to compare the recommendation approaches using prediction and classification accuracy metrics. Prediction accuracy metrics measure the deviation between a predicted rating $p(u,i)$ and the user's u true rating $r(u,i)$ of the item i for all users U and their ratings $P(u)$ in the test set. The most commonly used prediction accuracy metrics are the *mean absolute error* (MAE) and the *root mean squared error* (RMSE). The MAE measures the average absolute deviation of the expected ratings and the users' true ratings, whereas RMSE squares the deviations before they are averaged. Thus, the RMSE gives a higher weight to large errors, see equation 7. In the following evaluations, the RMSE is used and additionally, because of the data sets' sparsity, the coverage, i.e., the percentage of user-item pairs in the test set for which a rating can be predicted, is calculated

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{u\in U}\sum_{i\in P(u)}(p(u,i) - r(u,i))^2}{\sum_{u\in U}|P(u)|}}. \tag{7}$$

Though, in many applications, it is not important whether a recommender system is able to predict if the user

TABLE 2
MACE: Collaborative Filtering

	RMSE	Cov.	Prec.	Rec.	F1
IBCF	1.0814	20.57%	0.7267	0.2082	0.3237
UBCF	0.5770	24.15%	0.9664	0.2981	0.4556
SVD	0.8303	30.00%	0.9511	0.3302	0.4902
BMF	1.1947	100%	0.8912	0.3394	0.4916

will rate an item with one or two stars on a scale from 1 to 5, but that the system is able to predict that the user will not like the item and vice versa. The ability of a recommender system to distinguish between items the user will most probably like (i.e., relevant items) and items the user will most probably not like (i.e., irrelevant items) can be evaluated by measuring the precision and the recall [46]. The precision is calculated as the fraction of the number of items that are correctly identified as relevant and the number of all recommended items, see equation 8. The recall is calculated as the fraction of the number of items that are correctly identified as relevant and all relevant items, see equation 9. Thus, the precision reveals how pure the recommendations are while the recall shows how many of the relevant items are found. These measures often hold an inverse relationship, i.e., it is possible to increase the precision at the cost of decreasing the recall and vice versa. The f_β score combines precision and recall in which the value of β allows to weight the score of one of the measures more than the other, see equation 10. When the harmonic mean of both is considered, β has a value of 1 [47]. All three evaluation metrics can reach any value between 0 and 1 with a result of 1 being the best.

$$\text{precision} = \frac{|relevant\ items \cap recommended\ items|}{|recommended\ items|} \tag{8}$$

$$\text{recall} = \frac{|relevant\ items \cap recommended\ items|}{|relevant\ items|} \tag{9}$$

$$\text{f}_\beta \text{ score} = (1+\beta^2) * \frac{precision * recall}{(\beta^2 * precision) + recall}. \tag{10}$$

6.2 Results for MACE

This section first discusses the results for the baselines, i.e., the CF approaches IBCF, UBCF, SVD, and BMF, as well as their content-boosted versions and the pure content-based approach. Thereafter, the results for the usage context-based and -boosted approaches are presented and compared to the baselines.

6.2.1 Baselines

Table 2 shows the RMSE, the coverage (Cov.), the precision (Prec.), the recall (Rec.), and the f_1 score (F1) that are achieved using the CF approaches IBCF, UBCF, SVD, and BMF on the MACE data set. The neighbourhood-based approaches perform similarly according to the number of predicted ratings, i.e., 20.57 percent (IBCF) and 24.15 percent (UBCF). Though, the IBCF clearly performs worse in terms of RMSE with a value of 1.0814 (IBCF) compared to 0.5770 (UBCF). This is due to the fact that each user rated on average 3.79 learning object while each object is only rated by on average 1.11 users. Thus, the similarities between users can be calculated with more confidence than the similarities between objects. In fact, the UBCF achieves the lowest RMSE of all CF approaches which is also reflected by the high precision it reaches (i.e., 0.9664). The SVD predicts 30 percent of the ratings which is slightly more than the NH-based approaches and its RMSE lies in the middle with a value of 0.8303. However, its precision (i.e., 0.9511) is almost as high as the one achieved by the UBCF and because of its higher recall (i.e., 0.3302) it also reaches a higher f_1 score (i.e., 0.4902). The BMF always predicts a rating, even if the user or the object of a user-object pair is not part of the associated training set which is also reflected by the comparatively high recall of the BMF (i.e., 0.3394). Thus, the BMF reaches the highest f_1 score (i.e., 0.4916) even though it performs worst in terms of RMSE and worse than UBCF and SVD in terms of precision.

TABLE 3
MACE: Content-Based Filtering

	RMSE	Cov.	Prec.	Rec.	F1
CBF	0.8877	56.60%	0.7895	0.4383	0.5637

Table 3 shows the results for the content-based filtering that uses the available semantic metadata to establish relations between the objects. The CBF reaches a medium RMSE (i.e., 0.8877) while it predicts ratings for 56.60 percent of the user-object pairs in the test sets. The only CF approach that predicts more ratings is the BMF which in return has a much higher RMSE. Furthermore, even though the CBF predicts less ratings, its recall is higher. Consequently, the CBF achieves a higher f_1 score than the BMF.

Table 4 shows the results for the content-boosted versions of the CF approaches. It can be seen that with a user-item-matrix that is filled in using content similarities, i.e., with a less sparse matrix, the results of the different CF approaches are converging. For example, the RMSE increases for the UBCF and the SVD, whereas it decreases for the IBCF and the BMF. However, all approaches benefit from the content-boosting which clearly can be seen by means of the f_1 scores that increase by at least 35.84 percent (for the UBCF) and up to 76.64 percent (for the IBCF). This increase can mainly be ascribed to the increased recall which in turn is forced by the higher coverage for the IBCF, the UBCF, and the SVD. For the BMF, that predicts a rating for each user-object pair, the increased recall can be explained by the smaller RMSE (i.e., 1.0169 instead of 1.1947), which causes more objects to be recognised as relevant. Overall, the BMF still produces the worst RMSE and

TABLE 4
MACE: Content-Boosted Collaborative Filtering

	RMSE	Cov.	Prec.	Rec.	F1
IBCF	0.9456	68.49%	0.7769	0.4524	0.5718
UBCF	0.9156	69.25%	0.8060	0.5023	0.6189
SVD	0.9260	70.94%	0.8276	0.5614	0.6690
BMF	1.0169	100%	0.7641	0.6471	0.7007

TABLE 5
MACE: Usage Context-Based Filtering

	RMSE	Cov.	Prec.	Rec.	F1
MI	0.9818	57.17%	0.7637	0.3486	0.4787
χ^2	1.0312	50.00%	0.6945	0.2855	0.4047
LL	1.0627	41.51%	0.7278	0.3359	0.4597

TABLE 6
MACE: Usage Context-Boosted Collaborative Filtering

	RMSE	Cov.	Prec.	Rec.	F1
IBCF	0.9201	69.62%	0.7858	0.4892	0.6030
UBCF	0.8948	70.00%	0.7922	0.5699	0.6629
SVD	0.8831	70.75%	0.8390	0.5652	0.6754
BMF	1.0009	100%	0.7013	0.7476	0.7237

also the lowest precision (i.e., 0.7641) of the four content-boosted CF approaches but also the highest recall (i.e., 0.6471) which results in the highest f_1 score (i.e., 0.7007).

This section shows that the CBF performs better regarding the recall and the f_1 score than the un-boosted CF approaches. On the contrary, the CF approaches tend to achieve higher values for the precision. Thus, a combination seems reasonable and in fact, all approaches benefit in terms of quantity and recall from the feature augmentation that helps to compensate the sparsity of the user-item-rating matrix. The precision decreases in most cases since the filled-in values are not completely accurate but this is balanced by the increasing recall as shown by the f_1 scores that increase for all CF approaches using the content-boosting. Overall, the UBCF performs best in terms of RMSE on both the original and the filled-in matrix while the SVD always reaches the highest precision. However, when considering the quantity, the recall, and the f_1 score, the BMF is the best performing approach on both matrices (i.e., the original and the content-boosted one).

6.2.2 Usage-Based Filtering

Table 5 shows the results for the usage context-based filtering approaches that utilise different association measures and the average threshold. The approaches differ not as much from one another in their performance as for example the CF approaches. However, the MI-based one is the best approach according to all evaluation metrics. This is consistent with the findings of the first experiment, i.e., the object similarities calculated using the MI-based approach show the highest correlation to the content-based similarities and additionally, it finds relations for more objects than the other approaches. Interestingly, the LL-based approach almost predicts as much ratings as the MI-based one and more than the approach using cor-χ^2, even though the LL-based approach creates relations for less objects. However, those objects that hold relations to other objects according to the LL-based approach also hold more relations. For example, the MI-based approach creates relations for 9,173 objects on which each object holds on average 944 relations to other objects whereas the LL-based one only creates relations for 5,423 objects, however, each object holds an average 1,966 relations.

Summing up, the MI-based approach is the best performing UC-based filtering. It achieves a value of 0.9818 for the RMSE and, thus, it also performs better than the IBCF and the BMF. Furthermore, this approach returns predicted ratings for 57.17 percent of the user-object pairs in the test sets which is slightly more than the CBF produces and approximately twice the number of ratings that can be predicted by the IBCF, the UBCF, and the SVD. Finally, it achieves a slightly higher recall than the BMF. Thus, in combination with the precision of 0.7637, it achieves an f_1 score of 0.4787 which approximates to the one of the un-boosted BMF (i.e., 0.4916).

6.2.3 Usage-Boosted Filtering

This section describes the evaluation of the usage-boosted filtering approaches. Therefore, the UC-based filtering that utilises MI is used to fill in the user-item-matrix which then serves as input for the different CF approaches. This combination is chosen because it performs bests when used in standalone recommender systems and they also receive the best appraisals in experiment 1. The associated results for the UC-boosted CF approaches are given in Table 6. Similarly to the content-boosted CF approaches, the RMSE increases for the UBCF and the SVD whereas it decreases for the IBCF and the BMF. However, all RMSE values are smaller than for the content-boosted versions. Furthermore, the number of predicted ratings approximately triples compared to the un-boosted IBCF, UBCF, and SVD to an even slightly higher value than for their content-boosted versions. Furthermore, while the precision decreases on average by 10.57 percent, the recall increases on average by 104.40 percent compared to the un-boosted versions. Thus, all approaches with IBCF as exception, achieve a higher f_1 score than the CF approaches including their content-boosted versions and the CBF. Again, the BMF reaches the worst RMSE (i.e., 1.0009) and precision (i.e., 0.7013) but the highest recall (i.e., 0.7476) and f_1 score (i.e., 0.7237).

6.3 Results for Travel Well

6.3.1 Baselines

Table 7 shows the results of the baseline approaches for the Travel well data set. The NH-based approaches and the SVD perform similarly in respect to all evaluation metrics. They are able to predict ratings for about a third of the user-item pairs in the test sets with 0.8946 as their median RMSE. Thus, their precision is rather high with 0.8844 as their median value and their recall is rather low with 0.3116 as their median value. The BMF reaches better values regarding all evaluation metrics except for the precision (i.e.,

TABLE 7
Travel Well: Collaborative Filtering Approaches

	RMSE	Cov.	Prec.	Rec.	F1
IBCF	0.8909	31.48%	0.8485	0.3116	0.4558
UBCF	0.8946	31.48%	0.8844	0.2920	0.4390
SVD	0.9063	35.83%	0.8957	0.3169	0.4681
BMF	0.8651	100%	0.7922	0.9617	0.8688

TABLE 8
Travel Well: Content-Based Filtering

	RMSE	Cov.	Prec.	Rec.	F1
CBF	0.8094	86.90%	0.8237	0.8140	0.8188

0.7922) where the other approaches reach values that are higher by 7.11 percent (IBCF) and up to 13.06 percent (SVD). However, due to its high recall (i.e., 0.9617) the BMF reaches the by far best f_1 score (i.e., 0.8688). The next best approach is the SVD with an f_1 score of 0.4681, followed by the IBCF (with 0.4558) and the UBCF (with 0.4390). The ordering of the approaches according to their f_1 score is similar to the one for the MACE data set, however, the performance of the BMF differs strikingly from the other approaches on the Travel well data set. The explanation for this lies most probably in the rating value distribution. In the Travel well data set, 77.96 percent of the rated objects hold a value of at least four out of five stars and are thus regarded as relevant for the associated users. This is to say, if a recommender system assumes all objects in the Travel well test sets to be relevant for the associated users, it would achieve a precision of 0.7796, a recall of 1, and an f_1 score of 0.8756. Thus, in the following, the evaluation focusses more on the precision than on the recall.

Table 8 shows the results for the content-based filtering on the Travel well data set. This approach predicts ratings for 86.90 percent of the user-item pairs and reaches a RMSE of 0.8094, i.e., its ratings are strikingly more accurate than those of the CF approaches. Overall, it reaches a recall of 0.8140 which is 15.36 percent lower than the one of the BMF, though, it reaches a 3.98 percent higher precision (i.e., 0.8237) which is in the focus here.

Similarly to the MACE data set, the utilisation of the content-boosting leads to a convergence of the CF approaches' results as can be seen in Table 9. The RMSE slightly decreases for all approaches (by 2.17-4.66 percent) while they predict ratings for over 93 percent of the user-object pairs in the test sets which forces a strong increase of the recall for all approaches except for the BMF (which already predicted all ratings). Thus, the f_1 score rises only by 0.07 percent for the BMF, but it rises by 78.08 percent (SVD) up to 92.73 percent (UBCF) for the other approaches. Overall, the BMF is the best performing approach on the content-boosted user-item-rating matrix in terms of RMSE, quantity, recall, and f_1 score. Additionally, it performs comparably to the other content-boosted CF approaches regarding the precision. Finally, its f_1 score is slightly higher than the one of the original BMF while it also reaches a lower RMSE and a higher precision.

TABLE 9
Travel Well: Content-Boosted Collaborative Filtering

	RMSE	Cov.	Prec.	Rec.	F1
IBCF	0.8494	93.26%	0.8095	0.8964	0.8507
UBCF	0.8537	93.26%	0.8151	0.8795	0.8461
SVD	0.8714	93.41%	0.8238	0.8436	0.8336
BMF	0.8463	100%	0.8181	0.9275	0.8694

TABLE 10
Travel Well: Usage Context-Based Recommendation

	RMSE	Cov.	Prec.	Rec.	F1
MI	0.8145	94.20%	0.8266	0.8680	0.8468
χ^2	0.8166	88.63%	0.8329	0.7901	0.8109
LL	0.8606	53.60%	0.8223	0.5253	0.6411

6.3.2 Usage-Based Filtering

Table 10 shows the results for the usage context-based filtering approaches using the different association measures on the Travel well data set. Apart from LL, the measures perform relatively similar with MI being the best in terms of RMSE, quantity, recall, and f_1 score. This is similar to the MACE data set and also consistent with the findings of the first experiment. In contrast to the MACE data set, LL does not perform well and receives the worst values in all categories which is due to the fact that it finds fewer object relations than the other approaches. All UC-based filtering approaches perform clearly better than the content- and un-boosted CF approaches regarding the RMSE and outperform the un-boosted IBCF, UBCF, and SVD in terms of the f_1 score. The UC-based filtering using MI achieves a value of 0.8468 for the f_1 score which approximates to the f_1 score of the BMF (i.e., 0.8688). Additionally, its RMSE is smaller (i.e., 0.8145 compared to 0.8651) and the precision is higher (0.8266 compared to 0.7922). Finally, it performs better than the content-based approach in terms of precision, recall, and f_1 score.

6.3.3 Usage Context-Boosted Filtering

This section holds the results for the usage context-boosted approache. Similarly to the MACE data set, the association measure MI is used in combination with the UC-boosted approaches because this combination reaches the best results when used as a standalone recommender system. Table 11 shows the results for the usage context-boosted CF approaches. The precision of the IBCF, the UBCF, and the SVD decreases by 6.67 percent (IBCF) up to 8.51 percent (SVD) compared to their un-boosted versions. In return, the recall of these approaches increases by 171.22 percent (SVD) up to 190.44 percent (UBCF). Thus, they achieve f_1 scores between 0.8336 (UBCF) and 0.8443 (SVD). For the BMF, the precision increases by 4.03 percent whereas the recall decreases by 4.13 percent. Overall, the UC-boosted BMF achieves a slightly higher f_1 score than the un-boosted BMF (i.e., 0.8703 compared to 0.8688) and holds a higher precision (i.e., 0.8241 compared to 0.7922) and a lower RMSE (i.e., 0.8196 compared to 0.8651). Concluding, the UC-boosted BMF even reaches

TABLE 11
Travel Well: Usage Context-Boosted Collaborative Filtering

	RMSE	Cov.	Prec.	Rec.	F1
IBCF	0.8769	94.67%	0.7919	0.8963	0.8409
UBCF	0.9255	94.67%	0.8196	0.8481	0.8336
SVD	0.8849	94.81%	0.8296	0.8595	0.8443
BMF	0.8196	100%	0.8241	0.9220	0.8703

a slightly better precision than the content-boosted BMF which results in a slightly better f_1 score.

6.4 Interpretation

The data set collected in the MACE portal offers more usage data (i.e., more tracked event types) than the data set collected in the Travel well portal. Furthermore, the events in the MACE data set hold timestamps and, thus, a more specific user session definition can be applied when analysing the data. Therefore, as shown in the first experiment, the MACE usage data is better suited to create usage context-based similarities that imply semantic relations. In contrast, the objects in the Travel well data set hold more content information than the objects in the MACE data set (i.e., about 97 percent of the learning objects in the Travel well data set hold tags or classifications which only holds true for about 79 percent of the MACE objects, see Section 4). As a result, all usage context-boosted approaches perform better than the content-boosted approaches for the MACE data set, whereas for the Travel well data set, the content-boosted approaches perform better.

For both data sets, the MF methods SVD and BMF profit from the usage context-based and the content-based boosting approach. The standard collaborative filtering approaches IBCF and UBCF are more sensible concerning expected ratings in the user-item-rating matrix that differ from the true user ratings.

To conclude, boosting can be recommended for the use in learning portals, especially in combination with the biased matrix factorisation approach. If the collected usage data is fine-grained, i.e., all events concerning a learning object are stored and the timestamp is given, the use of usage context-based similarity can even outperform the use of semantic metadata-based similarity.

7 Conclusion

In this paper, we introduce a new way to calculate similarities between learning objects by considering their usage contexts. The usage context of a learning object is defined by the objects it significantly often co-occurs with in user sessions. This way, learning objects are similar if they co-occur with the same learning objects. This also means, two learning objects can be similar even if they were never used together in the same user session. Our hypothesis that usage context-based similarity is an indication for content similarity is supported by the experimental evaluation using the data sets MACE and Travel well. This means that the presented approach is able to utilise the users' knowledge and context, which is inherent in their activities, in order to reveal item relations without forcing the users to explicitly share their knowledge.

The usage context-based similarity can be used in several ways. Here, we create a usage context-based recommender and use it to boost existing recommendation approaches by filling up the underlying user-item-rating matrix with expected ratings that are calculated using the usage context-based similarity. We evaluate this approach against the original recommendation approaches and their content-boosted equivalents. The evaluation shows that the usage context-based similarity is able to even outperform the content-based similarity. This is due to the fact that by analysing an object's usage contexts, a system can be enabled to profit from the users' knowledge without forcing them to explicitly share it. Thus, simple log files of systems providing learning objects are sufficient to calculate similarities between the objects that can be incrementally updated.

The results motivate us to further develop this approach. First, we will try to find a suitable way to combine usage context and content-based similarities to not waste any available information. Additionally, we plan to evaluate our approach in a running system to gather more insights about the usefulness of the created recommendations.

References

[1] N. Manouselis, H. Drachsler, R. Vuorikari, H. Hummel, and R. Koper, "Recommender systems in technology enhanced learning," in *Recommender Systems Handbook*, L. Rokach, B. Shapira, and P. Kantor, Eds. New York, NY, USA: Springer, 2010, pp. 387–415.

[2] K. Verbert, H. Drachsler, and N. Manouselis, "Dataset-driven research for improving tel recommender systems," in *Proc. 1st ACM Int. Conf. Learn. Anal. Knowl.*, 2011, pp. 44–53.

[3] C. Desrosiers and G. Karypis, "A comprehensive survey of neighborhood-based recommendation methods," in *Recommender System Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. New York, NY, USA: Springer, 2011, pp. 107–144.

[4] A. Ceglar and J. F. Roddick, "Association mining," ACM Comput. Surv., vol. 38, no. 2, Jul. 2006. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1132956.1132958

[5] R. V. Meteren and M. V. Someren, "Using content-based filtering for recommendation," in *Proc. MLnet/ECML2000 Workshop*, 2000, pp. 47–56.

[6] M. Degemmis, P. Lops, and G. Semeraro, "A content-collaborative recommender that exploits WordNet-based user profiles for neighborhood formation," *User Model. User-Adapted Interact.*, vol. 17, no. 3, pp. 217–255, 2007.

[7] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web*, 2001, pp. 285–295.

[8] G. Linden, B. Smith, and J. York, "Amazon.com recommendations-item-to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, Jan. 2003.

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

[10] X. Luo, Y. Xia, and Q. Zhu, "Applying the learning rate adaptation to the matrix factorization based collaborative filtering," *Knowl.-Based Syst.*, vol. 37, pp. 154–164, Jan. 2013.

[11] M. Zhang, J. Tang, X. Zhang, and X. Xue, "Addressing cold start in recommender systems: A semi-supervised co-training algorithm," in *Proc. 37th Int. ACM SIGIR Conf. Res. Develop. Inform. Retrieval*, 2014, pp. 73–82.

[12] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Model. User-Adapted Int.*, vol. 12, no. 4, pp. 331–370, Nov. 2002.

[13] B. Mobasher, X. Jin, and Y. Zhou, "Semantically enhanced collaborative filtering on the web," in *Proc. 1st Eur. Web Min. Forum*, Sep. 2003, pp. 57–76.

[14] H. Khrouf and R. Troncy, "Hybrid event recommendation using linked data and user diversity," in *Proc. 7th ACM Conf. Recommender Syst.*, pp. 185–192.

[15] T. Hornung, C.-N. Ziegler, S. Franz, M. Przyjaciel-Zablocki, A. Schatzle, and G. Lausen, "Evaluating hybrid music recommender systems," in *Proc. IEEE/WIC/ACM Int. Joint Conf. Web Intell. Intell. Agent Technol.*, Nov. 2013, pp. 57–64.

[16] T. Tran and R. Cohen, "Hybrid recommender systems for electronic commerce," in *Proc. Knowl.-Based Electron. Markets, AAAI Workshop*, 2000, pp. 78–83.

[17] M. Van Setten, "Supporting people in finding information," Ph.D. dissertation, Telematica Instituut, Enschede, Univ. Enschede, the Netherlands, 2005.

[18] P. Melville, R. J. Mooney, and R. Nagarajan, "Content-boosted collaborative filtering for improved recommendations," in *Proc. 18th Nat. Conf. Artif. Intell.*, 2002, pp. 187–192.
[19] P. Forbes and M. Zhu, "Content-boosted matrix factorization for recommender systems," in *Proc. 5th ACM Conf. Recommender Syst.*, 2011, p. 261.
[20] N. Manouselis, H. Drachsler, R. Vuorikari, H. Hummel, and R. Koper, "Recommender systems in technology enhanced learning," in *Recommender System Handbook*. New York, NY, USA: Springer, 2011, pp. 387–415.
[21] P. Brusilovsky, L. N. Cassel, L. M. Delcambre, E. A. Fox, R. Furuta, D. D. Garcia, F. M. Shipman, and M. Yudelson. (2010, Jan). Social navigation for educational digital libraries. *Procedia Comput. Sci.* [Online]. *1(2)*, pp. 2889–2897. Available: http://linkinghub.elsevier.com/retrieve/pii/S1877050910003303
[22] H. Drachsler, D. Pecceu, T. Arts, E. Hutten, L. Rutledge, P. van Rosmalen, H. Hummel, and R. Koper, "ReMashed-Recommendations for mash-up personal learning environments," in *Proc. 4th Eur. Conf. Technol. Enhanced Learn.*, 2009, pp. 788–793.
[23] N. Manouselis, R. Vuorikari, and F. Van Assche. (2010, Jul). Collaborative recommendation of e-learning resources: An experimental investigation. *J. Comput. Assist. Learn.*. [Online]. *26(4)*, pp. 227–242. Available: http://doi.wiley.com/10.1111/j.1365-2729.2010.00362.x
[24] O. C. Santos, "A recommender system to provide adaptive and inclusive standard-based support along the elearning life cycle," in *Proc.ACM Conf. Recommender Syst.*, 2008, p. 319
[25] R. Vuorikari and X. Ochoa, "Exploratory analysis of the main characteristics of tags and tagging of educational resources in a multi-lingual context," *J. Digit. Inform.2*, vol. 10, no. 2, 2009.
[26] G. Adomavicius, "Maximizing aggregate eecommendation diversity: A graph-theoretic approach," in *Proc. Workshop Novelty Divers. Recommender Syst., Held Conjunction ACM RecSys*, 2011, pp. 3–10.
[27] F. de Saussure, *Course in General Linguistics*. Chicago, IL, USA: Open Court Publishing, re-print 1986.
[28] G. Heyer, U. Quasthoff, and T. Wittig, *Text Mining: Wissensrohstoff Text: Konzepte, Algorithmen, Ergebnisse*. Herdecke, Germany: W3L-Verlag, 2006.
[29] R. Cooley, B. Mobasher, and J. Srivastava, "Web Mining: Information and pattern discovery on the world wide web introduction," in *Proc. 9th IEEE Int. Conf. Tools Artif. Intell.*, 1997, pp. 558–567.
[30] C. D. Manning, P. Raghavan, and H. Schütze, *An Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2008.
[31] S. Evert, "The Statistics of Word Cooccurrences: Word Pairs and Collocations" Ph.d. dissertation, Institut für maschinelle Sprachverarbeitung, University Stuttgart, Germany, 2004.
[32] H. DeGroot M. and M. J. Schervish, *Probability and Statistics*, 3rd ed. Boston, MA, USA: Addison-Wesley, 2002.
[33] T. E. Dunning, "Accurate methods for the statistics of surprise and coincidence," *Comput. Linguistics*, vol. 19, no. 1, pp. 61–74, 1993.
[34] S. Evert, "Corpora and collocations," in *Corpus Linguistics. An International Handbook*. Berlin, Germany: De Gruyter, 2009, vol. 2, pp. 223–233.
[35] M. Stefaner, E. D. Vecchia, M. Condotta, M. Wolpers, M. Specht, S. Apelt, and E. Duval, "Mace-enriching architectural learning objects for experience multiplication," in *Proc. 2nd Eur. Conf. Technol. Enhanced Learn.*, 2007, pp. 322–336.
[36] H.-C. Schmitz, M. Wolpers, U. Kirschenmann, and K. Niemann, "Contextualized attention metadata," in *Human Attention in Digital Environments*, C. Roda, Ed. Cambridge, U.K.: Cambridge Univ. Press, 2011.
[37] R. Vuorikari and D. Massart, "datatel challenge: European schoolnet's travel well data set," in *Proc. 1st Workshop Recommender Syst. Technol. Enhanced Learn.*, 2010.
[38] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, 1983.
[39] K. Pearson, *On Further Methods of Determining Correlation*. Cambridge, U.K.: Cambridge Univ. Press, 1907.
[40] J. Bortz, *Statistics*. Heidelberg, Germany: Springer, 1993.
[41] K. Niemann, M. Scheffel, M. Friedrich, U. Kirschenmann, H. C. Schmitz, and M. Wolpers, "Usage-based object similarity," *J. Universal Comput. Sci.*, vol. 16, no. 16, pp. 2272–2290, 2010.
[42] J. Lee, M. Sun, and G. Lebanon, "A comparative study of collaborative filtering algorithms," *ArXiv e-prints*, May 2012.
[43] Z. Gantner, S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "MyMediaLite: A free recommender system library," in *Proc. 5th ACM Conf. Recommender Syst.*, 2011, pp. 305–308.
[44] A. Paterek, "Improving regularized singular value decomposition for collaborative filtering," in *Proc. 13th ACM Int. Conf. Knowl. Discovery Data Mining Workshop*, 2007, pp. 39–42.
[45] S. Rendle and L. Schmidt-Thieme, "Online-updating regularized kernel matrix factorization models for large-scale recommender systems," in *Proc.ACM Conf. Recommender Syst.*, 2008, pp. 251–258.
[46] X. Su and T. M. Khoshgoftaar. (2009). A survey of collaborative filtering techniques. *Adv. Artif. Intell.*. [Online]*2009(Section 3)*, pp. 1–19. Available: http://www.hindawi.com/journals/aai/2009/421425/
[47] C. van Rijsbergen, *Information Retrieval*, 2nd ed. London, U.K.: Butterworth, 1979.

Katja Niemann studied computational linguistics and computer science at the University of Heidelberg where she received her master's degree in 2007. She has been working as a researcher at Fraunhofer FIT since 2008 and is working towards the PhD degree at the RWTH Aachen University. At FIT, she is mainly involved in e-learning projects supported by the EC like MACE, OpenScout, and OpenDiscoverSpace. In her PhD thesis, she analysed the usage of data objects in online portals to infer semantic similarities between them and, thus, to enhance recommender systems. Her research interests comprise recommender systems, data mining, text mining, information retrieval, and learning analytics amongst others.

Martin Wolpers received the PhD degree in electrical engineering and information technology from the Leibnitz University Hannover. He is the head of the research department "Context and Attention in Personalised Learning Environments" at Fraunhofer Institute for Applied Information Technology. He also holds the title "visiting professor" at the Katholieke Universiteit Leuven, Belgium. He has been strongly involved in a number of highly successful European Projects, e.g., coordinator of the ROLE project as well as partner in OpenDiscoverySpace, OpenScout, and NaturalEurope projects. He is the vice-president of the European Association of Technology Enhanced Learning (EATEL) and the president of the MACE Association, result of the successful coordination of the eContent+ project MACE. His research interests include the improvement of technology enhanced learning by relying on new and emerging technologies. This includes work on infrastructures for information processing, knowledge management and provision as well as contextualised attention metadata and information elicitation (among others).