Journal of Universal Computer Science, vol. 16, no. 16 (2010), 2272-2290
submitted: 31/3/10, accepted: 31/7/10, appeared: 28/8/10 © J.UCS

Usage-based Object Similarity

Katja Niemann, Maren Scheffel
Martin Friedrich, Uwe Kirschenmann
Hans-Christian Schmitz, Martin Wolpers
(Fraunhofer Institute for Applied Information Technology FIT
Schloss Birlinghoven, Sankt Augustin, Germany
{katja.niemann, maren.scheffel, martin.friedrich, uwe.kirschenmann
hans-christian.schmitz, martin.wolpers}@fit.fraunhofer.de)

Abstract: Recommender systems are widely used online to support users in finding relevant information. They can be based on different techniques such as content-based and collaborative filtering. In this paper, we introduce a new way of similarity calculation for item-based collaborative filtering. Thereby we focus on the usage of an object and not on the object's users as we claim the hypothesis that similarity of usage indicates content similarity. To prove this hypothesis we use learning objects accessible through the MACE portal where students can query several architectural repositories. For these objects, we generate object profiles based on their usage monitored within MACE. We further propose several recommendation techniques to apply this usage-based similarity calculation in real systems.

Key Words: attention metadata, recommender systems, item-based collaborative filtering
Category: H.3.3, H.4.0, L.3.2

1 Introduction

With more and more information available online it can be difficult to find what one is really looking for, what is interesting in the current circumstances or what is best to be looked at next. Instead of aimlessly browsing through large amounts of data by themselves, people often use recommender systems [Adomavicius and Tuzhilin 2005] to suggest to them what to read, do, buy, watch or listen to next.

We pick up the approach described in [Friedrich et al. 2009] where we claimed the hypothesis that usage similarity gives rise to content similarity and can thus be used for recommendations. Based on this assumption it is possible to recommend an object to a user because its usage context is similar to the usage context of the object she is currently looking at although the two objects do not have to have been used together before. In this paper, we describe the approach in more detail and enhance it by integrating and evaluating further usage-based – or more precisely usage context-based – object similarity metrics. Furthermore, we outline several methods of how to use these metrics for recommendation. We refer to the MACE project[1] as a test bed. MACE (*Metadata for Architectural*

[1] http://portal.mace-project.eu

Contents in Europe [Stefaner et al. 2007]) provides graphical metadata-based access to learning resources in architecture, which are connected across repository boundaries to enable students to find relevant information more efficiently.

The rest of this paper is structured as follows: First we will explain basic principles of our approach and a definition of context in section 2, followed by a presentation of different types of recommender systems and an explanation of where our approach fits in in section 3. In section 4 we will explain the different similarity measures we use and how they are calculated. After describing our test bed, section 5 deals with the analysis of the collected data and presents our results. In section 6 we introduce three different ways of using our approach for recommendation. In the conclusion we then close with possible further work.

2 Basic notions: context and paradigmatic relatedness

The understanding of context is fundamental to correctly interpret user interactions and to design suitable system reactions [Vuorikari and Berendt 2009]. However, there is no uniform definition of the term "context" in the literature as it depends on the domain and the purpose of the contextualisation what needs to be considered as context. [Dey 2001] describes the context as any information that can be used to characterise the situation of entities. With such a broad notion it becomes clear that a concrete context cannot be exhaustively specified without focussing on some selected parameters [Zimmermann 2007].

The notion of context used here is inspired by a notion successfully used in linguistics, namely by the notion of a word context. In language, words stand in linear orders. The context of a word W is defined by the words that occur before and after W. The same word can stand in different contexts, and different words can stand in very similar contexts. If two words have very similar contexts, then they are said to be paradigmatically related [Saussure 1986]. A simple example: the word "car" can appear in different sentences and thus in different contexts. In very many cases it can be replaced by "vehicle". Therefore, the two words share a great number of contexts; they are paradigmatically related. It has been shown that paradigmatically related words are also semantically related. Thus, paradigmatic relations lead us to semantic relations or, in other words, context similarity correlates with content relatedness [Heyer et al. 2006].

We take up this insight and form our working hypothesis that it holds true not only for words but also for arbitrary data objects that usage context similarity correlates with content relatedness. Before we can test this hypothesis however, we have to define contexts of data objects in analogy to word contexts. We do so with an intermediate step, namely by defining contexts of user actions first and then reducing these to object contexts. To this end, we assume that users' activities can be described as linear sequences of atomic actions. In analogy to the context of a word W, the context of an atomic action A can then be defined by the actions that were performed before and after A. The actions carried out before form A's pre-context; the actions carried out after A form its post-context.

We record user actions (e.g. clicking a hyperlink, adding a tag, downloading a document) and store them as CAM instances (*Contextualized Attention Metadata*, [Wolpers et al. 2007]). A CAM instance comprises the user, the accessed object, the action performed on the object and additional parameters like the time and the session in which the action occurred – a session comprises all actions occurred during the login and logout of a user. CAM instances of a singular user have time stamps and can be put in a linear order. For every CAM instance representing a user action A, a context as defined above comprises the CAM instances recorded before and after A, but within the same session.

Every CAM instance representing a user action involves a user and a data object. Instead of taking complete instances into account we can focus exclusively on the involved data objects. That is, we can reduce actions to the involved data objects, action sequences to object sequences and action contexts to object contexts. By this last step we finally specify the context of an object O by the objects that have been used before and after O. Like a word W, an object O can occur in various contexts, and different objects can occur in similar contexts. If they occur in similar contexts, then they are paradigmatically related. According to our hypothesis, paradigmatically related objects are also semantically related.

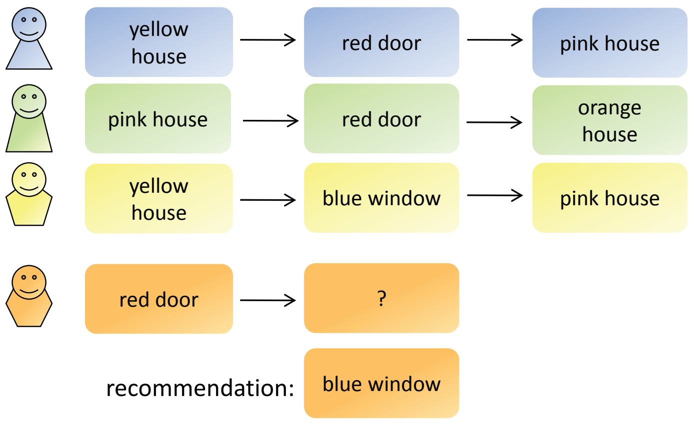

Figure 1: Recommendation of an object based on its usage context

Figure 1 illustrates our approach. As the objects *red door* and *blue window* appear in similar contexts, i.e. the objects accessed before respectively after *red door* were also accessed before respectively after *blue window*, namely *yellow* *house* and *pink house*, it is likely that they are also related by their content and are thus in the user's line of interest. After accessing *red door* the fourth user gets the recommendation to access *blue window* although *red door* and *blue window* have never been accessed within the same session. We will therefore argue that recommendations can be improved (i) by comparing the current usage history of a user with the usage history of data objects and (ii) by relating data objects according to their usage similarity.

3 Recommender systems

In this chapter we will first describe existing approaches of object similarity calculation used in recommender systems to then state where our approach fits in and in which way it differs from the existing approaches. Recommender systems deal with the delivery of items selected from a collection that the user is likely to find interesting or useful. Over the last decade a lot of research has been done on recommender systems and their improvement. The most common approaches of object similarity calculation used in recommender systems are collaborative filtering and content-based filtering as well as hybrid approaches which combine aspects of more than one filtering technique [Candillier et al. 2009].

Content-based systems use the item's attributes and the user's preferences for recommendation. Item profiles can be created automatically, e.g. through keyword extraction for text documents, or manually, e.g. for a database of cars holding attributes like brand and horsepower. User profiles can be built explicitly by asking the users about their interests or implicitly by a user's given ratings. For recommendation the user profiles are matched against the item profiles and the most suitable unknown items are recommended. Several systems were developed that use content-based filtering to help users find information. PRES (Personalized REcommender System) [Meteren and Someren 2000] creates dynamic hyperlinks for a web site that contains a collection of advises about do-it-yourself home improvement to enable the user to find relevant articles more easily. The items are represented through a set of automatically extracted terms. Based on the items a user finds interesting a user model is induced that enables the filtering system to classify unseen items into a positive or a negative class. Syskill & Webert [Pazzani et al. 1996] is a software agent that generates user profiles based on explicit feedback. The profile of the current user can then be used to suggest which links the user would be interested in exploring and also to construct queries to find pages on the World Wide Web that would interest her. However, recommender systems relying on content-based filtering suffer from problems like the new user problem, i.e. user profiles first have to evolve to suffice for recommendations, and overspecialisation, e.g. a user mainly listening to heavy metal music hardly gets recommendations for classical music, even if she listens to it once in a while. Additionally, it can be time-consuming and expensive to maintain the item profiles.

Systems based on collaborative filtering do not consider the item's attributes but make use of user ratings on items that can be explicit (e.g. rate a book with 3 stars) or implicit (e.g. visit a site, listen to a song). These ratings are the basis for user-based collaborative filtering techniques where an item is suggested to a user based on ratings of that item by users most similar to her. A prominent example for such a system is MovieLens[2], which is a movie recommendation website that works by matching together users with similar opinions about movies. Each member of the system has a neighbourhood of other like-minded users and the ratings from these neighbours are used to create personalised recommendations. Advantages of collaborative filtering are that no item profiles need to be created and cross-genre niches can be identified. However, the new user problem still exists and the new item problem, i.e. an item that is rated by a few users only will not be recommended, is introduced. Additionally, the user-based collaborative filtering approach makes it necessary to calculate similarities between users on the fly since changes of user profiles need to be considered immediately. This approach is thus computationally expensive, particularly when the data set or the number of users is growing.

Hybrid systems are implemented to exert the advantages from more than one technique while the drawbacks of single techniques can be compensated. Commonly, hybrid systems combine content-based and collaborative-based techniques, but knowledge-based or demographic-based techniques can for example be integrated as well [Burke 2007]. [Melville et al. 2002] introduced content-boosted collaborative filtering. For each user of a movie recommendation web site the system tries to predict a rating for every unrated film based on the rating data of similar users. If the confidence value for a prediction is under a pre-defined threshold, for example due to too little ratings for this movie, the rating is predicted based on the movie's textual description and the ratings the user assigned to movies with similar content. Other approaches to integrate content-based and collaborative filtering are for instance combining the ratings using a weighting scheme as proposed in [Mobasher et al. 2004] or showing the user the results from both techniques in a combined way as illustrated in [Smyth and Cotter 2000].

However, the scalability problem of user-based collaborative filtering cannot be compensated by combining it with other techniques. One approach to handle this problem is to reduce the data size. This can be done by comparing the user only to a small group of other users, by using a smaller range of items for the comparison of users or by ignoring very popular or very unpopular items. While these approaches might lead to better scaling, they usually worsen the

[2] http://movielens.umn.edu/

recommendation quality [Linden et al. 2003].

Item-based collaborative filtering approaches do not compare users but calculate the similarity of items by using the user's implicit and explicit ratings as a basis for recommendations. As the calculation of the item similarity table can be done offline, such algorithms are quicker and scale more easily [Sarwar et al. 2001]. A system relying on this approach is Amazon.com[3] where a product-to-product similarity matrix is used as basis for further calculations. The similarity of two products relies on the number of similar users the products share. That is to say, products often bought by the same users, either during one session or in general, get a higher similarity value than products that do not share so many users [Linden et al. 2003]. This implies that products that were never used or rated by the same users, do not get a similarity value.

We adapt the item-based collaborative filtering approach but use a different way of calculating item similarities. For two objects to be deemed similar, their context of usage needs to be similar. We can therefore recommend object O_2 to a user who previously used O_1 based on the fact that O_1 and O_2 have similar usage contexts and not on the fact that they were used by the same users.

4 Similarity calculation

In the following, we will describe the notion and generation of usage context profiles. Thereafter, we describe different ways to calculate the usage-based object similarity using the usage context profiles with an evaluation in the next chapter.

For every object used we create a usage context profile (UCP) which contains one usage context for every session in which the object was used. We define a usage context of a data object O as consisting of a pre-context and a post-context. The pre-context is the sequence of objects that were accessed before O, and the post-context is the sequence of objects that were accessed after O, within the same session. The sequences of objects are handled as sets or bags of objects in the following calculations. A session consists of all recorded actions between a user's log in and log out event. An object can be used several times and in different contexts. The UCP of an object is the set of its different usage contexts. Within MACE, UCPs can be derived from mere data transformation and integration of the CAM recordings.

More detailed: for every object, its UCP consists of one or several usage contexts (UC), that is pairs of pre- and post-contexts: $\{\langle(UC_1^{pre}), (UC_1^{post})\rangle, ..., \langle(UC_n^{pre}), (UC_n^{post})\rangle\}$. Figure 2 shows three sessions of different users. Based on these sessions, a UCP for the learning object *Materials.pdf* can be generated which is a set of two usage contexts consisting of the objects that were used before

[3] http://www.amazon.com/

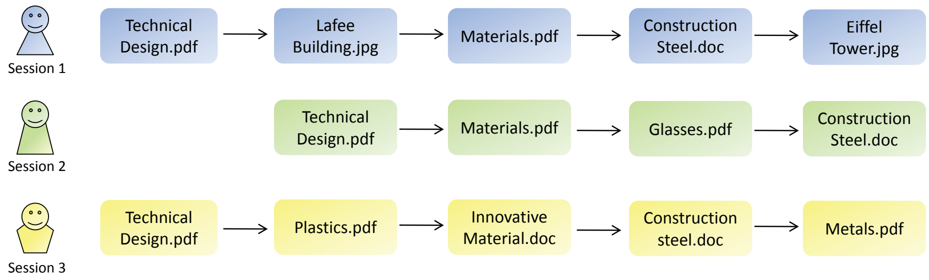

Figure 2: Learning paths of three different users

and after *Materials.pdf* in session 1 and session 2: $\{\langle($*TechnicalDesign.pdf*, *Lafee-Building.jpg*$)$, $($*ConstructionSteel.doc*, *EiffelTower .jpg*$)\rangle$, $\langle($*TechnicalDesign.pdf*$)$, $($*Glasses.pdf*, *ConstructionSteel.doc*$)\rangle\}$.

One way to calculate a similarity of two UCPs is to consider the pair-wise similarity of their respective usage contexts. The similarity of two usage contexts arises from similarities of the associated pre- and post-contexts. When pre- and post-contexts are handled as sets, the similarity between a pair of pre- or post-contexts is calculated using the Jaccard similarity measure [Matsumoto 2003], i.e. the ratio of the cardinality of their intersection and the cardinality of their union:

$$simPreUC_{set}\left(UC_1^{pre}, UC_2^{pre}\right) = \frac{\left|\cap\left(UC_1^{pre}, UC_2^{pre}\right)\right|}{\left|\cup\left(UC_1^{pre}, UC_2^{pre}\right)\right|} \tag{1}$$

$$simPostUC_{set}\left(UC_1^{post}, UC_2^{post}\right) = \frac{\left|\cap\left(UC_1^{post}, UC_2^{post}\right)\right|}{\left|\cup\left(UC_1^{post}, UC_2^{post}\right)\right|} \tag{2}$$

When pre- and post-contexts are handled as bags, i.e. they are represented as vectors, the similarity is calculated using the cosine similarity between the two vectors. This approach differs from the set-based approach in that extend that it is taken into account whether an object is accessed more than once in a pre- respectively post-context when calculating the context similarity:

$$simPreUC_{bag}\left(UC_1^{pre}, UC_2^{pre}\right) = \frac{UC_1^{pre} \cdot UC_2^{pre}}{\|UC_1^{pre}\| \, \|UC_2^{pre}\|} \tag{3}$$

$$simPostUC_{bag}\left(UC_1^{post}, UC_2^{post}\right) = \frac{UC_1^{post} \cdot UC_2^{post}}{\left\|UC_1^{post}\right\| \, \left\|UC_2^{post}\right\|} \tag{4}$$

The cosine similarity always takes a value between 0 and 1. The higher the value the higher is the similarity between the respective objects. The similarity of two usage contexts – pre-/post-context pairs – is defined as the arithmetic mean of the pre- and the post-context similarities:

$$simUC\left(UC_1, UC_2\right) = \frac{simPreUC\left(UC_1^{pre}, UC_2^{pre}\right) + simPostUC\left(UC_1^{post}, UC_2^{post}\right)}{2} \quad (5)$$

The similarity of two UCPs UCP_1 and UCP_2 can be defined as the arithmetic mean (cf. formula (6)) of the summarised pair-wise usage context similarities or as the median (cf. formula (7)) of these context similarities. The median is a measure that separates a frequency distribution into two equal halves. Moreover, it marks the value from which all other values of the distribution, when summarised, deviate in a minimum. The arithmetic mean and the median therefore have different characteristics. It is especially in asymmetric distributions that a median holds significance.

$$simUCP_{arithMean}\left(UCP_1, UCP_2\right) = \frac{\sum\limits_{\langle X,Y\rangle \in UCP_1 \times UCP_2} simUC\left(X, Y\right)}{\left|UCP_1 \times UCP_2\right|} \quad (6)$$

$$simUCP_{median}\left(UCP_1, UCP_2\right) = \begin{cases} simUC_{\frac{n+1}{2}}, & \text{n odd} \\ \frac{1}{2}\left(simUC_{\frac{n}{2}} + simUC_{\frac{n+1}{2}}\right), & \text{n even} \end{cases} \quad (7)$$

where $(simUC_1, ..., simUC_n)$ is the sorted set of pair-wise UC similarities of the usage contexts contained in UCP_1 and UCP_2.

The median is more stable towards outliers than the arithmetic mean. However, when the median is applied, the context similarity between two objects often becomes null even when there are UC similarities holding a higher similarity value. Chapter 5.1 discussed to what extend this is an advantege or disadvantege.

To demonstrate these definitions in more detail, the following formulas (i.e. (8), (9) and (10)) show the context similarity calculation of the learning objects *Materials.pdf* and *InnovativeMaterial.doc* based on the three example sessions of Figure 2. As there are no objects that were accessed more than once in a session, there is no difference between the set and the bag approach when calculating the similarities between the pre- and post-contexts respectively.

$$\begin{aligned} simPreUC\left(UC_1^{pre}, UC_3^{pre}\right) &= \frac{1}{3} \\ simPostUC\left(UC_1^{post}, UC_3^{post}\right) &= \frac{1}{3} \\ simUC\left(UC_1, UC_3\right) &= \frac{1}{3} \end{aligned} \quad (8)$$

$$
\begin{aligned}
simPreUC\left(UC_2^{pre}, UC_3^{pre}\right) &= \frac{1}{2} \\
simPostUC\left(UC_2^{post}, UC_3^{post}\right) &= \frac{1}{3} \\
simUC\left(UC_2, UC_3\right) &= \frac{5}{12}
\end{aligned}
\tag{9}
$$

$$
\begin{aligned}
simUCP_{arithMean}\left(UCP_{Materials.pdf}, UCP_{InnovativeMaterial.pdf}\right) &= \frac{9}{24} \\
&= 0,375
\end{aligned}
\tag{10}
$$

The UCP of *Materials.pdf* comprises the usage contexts UC_1 and UC_2, which are derived from session 1 and session 2, the UCP of *InnovativeMaterial.doc* holds only the usage context UC_3 which is derived from session 3. Therefore, the similarity calculation of *Materials.pdf* and *InnovativeMaterial.doc* using the arithmetic mean of the usage context similarities $simUC\left(UC_1, UC_3\right)$ and $simUC\left(UC_2, UC_3\right)$ results in a context similarity value of 0.375. The similarity calculation of *Materials.pdf* and *InnovativeMaterial.doc* using the median of the respective usage-based similarities also results in a similarity value of 0.375 since we only have two similarity values which are averaged according to formula (7).

5 Analysis

In the following we first introduce the test bed for our experiments, namely the MACE project, to then describe the calculation of the similarity values between learning objects derived from the MACE repository based on their semantic metadata which serve as tentative 'gold standard'. Thereafter, we illustrate the evaluation of our hypothesis that usage-based similarity gives rise to content similarity by calculating the correlation between semantic metadata-based and usage-based object similarity. Finally, we present the results of the manual comparison of content-based and usage-based similarity.

5.1 MACE as test bed

MACE (*Metadata for Architectural Contents in Europe*) is a European project where digital learning resources about architecture, stored in various repositories, are related with each other across repository boundaries to enable new and powerful ways of finding relevant information. Therefore, the metadata representations of the learning objects are stored on a central server. The representations base on the MACE application profile which in turn is based on the Learning

Object Metadata (LOM) standard [IEEE LOM Standard]. The MACE application profile comprises several categories that are used to specify learning objects in more detail, such as the general category where basic information about a learning object is stored and the annotation category where comments about a learning object's usefulness for education and the comments' origins can be stored.

While interacting with the MACE portal, users are monitored and their activities are recorded as CAM instances. Captured and for our calculation considered actions comprise downloading an object, viewing metadata of an object, and metadata provision activities like tagging. We use the CAM recordings to derive UCPs of respective MACE objects. In the long term, we aim to improve the system by recommending objects to the user for her current context based on her usage history and the UCPs of the MACE objects.

At the time of our data collection there were 1686 active users within the MACE test bed. 430 of these users had registered MACE accounts, the other 1256 users logged in as guests. We identified 4396 sessions – 3130 sessions of registered users and 1266 sessions of guests. In total, 13525 learning objects were accessed: 84.5% of them were accessed by registered users only, 5.6% by guests only and 9.9% by both groups. As the number of objects accessed by guests was very low and their average session length was only 2.34 accessed learning objects, we only analysed the sessions of registered users. These sessions had an average length of 13.67 accessed learning objects per session. All together we considered 12285 learning objects in the further calculations with each of these objects occurring on average in 3.35 sessions.

5.2 Semantic metadata-based object similarity

We use the LOM instances stored in the MACE repository to calculate the similarity between objects based on their semantic metadata. To calculate the similarity of two learning objects based on their semantic metadata, we consider the following assortment of available information: the English titles and descriptions, the repository the learning object is derived from, the learning resource types the learning object holds as well as the free text tags, classifications and competences the learning object is tagged with. 95% of the learning objects hold English titles and descriptions. By reason that every learning object is derived from a repository, each learning object holds a value for this attribute. 65% of the considered learning objects hold learning resource types like "narrative text" or "figure" with 1.25 learning resource types per learning object on average. 48% of the learning objects hold free text tags assigned by logged in users, where each tagged object holds on average about 8 tags. MACE also offers the possibility of editing advanced parts of the metadata to domain experts, namely classifications and competences, each of them being defined by a controlled vocabulary. The classification vocabulary is a taxonomy consisting of 2884 terms. The competence vocabulary contains 107 terms to describe the suitability of learning objects for the acquisition of special competences, e.g. 'Knowledge of internal environment control' and 'Understanding interaction between technical and environmental issues'. 39% of the considered learning objects already have classifications, with each object containing an average of 2.5 classification terms. Finally, 26% of the considered learning objects are already assigned with competences, where each of them contains an average of 6 competence terms.

Before calculating the semantic metadata similarity, the titles and descriptions are pre-processed: after removing stop words from the free text values (titles, descriptions, tags) the remaining words undergo a stemming using the Snowball Stemmer [Porter 1980]. To compare the learning objects document vectors describing the learning objects are generated and the similarity is calculated using the cosine similarity, i.e. measuring the similarity between two vectors by calculating the cosine of the angle between them:

$$cos\,(\theta) = \frac{X \cdot Y}{\|X\|\,\|Y\|} \tag{11}$$

where X is the vector of object O_1 and Y the vector of object O_2. The cosine similarity always takes a value between 0 and 1, with a higher value standing for a higher similarity.

5.3 Correlation of semantic metadata-based and usage-based object similarity

To prove our hypothesis that usage-based similarity indicates content similarity, we use the metadata of the learning objects as shallow content representations as described in chapter 5.2 and calculate the Pearson Correlation Coefficient [Pearson 1907] between the semantic metadata-based and the usage-based similarity distribution with the following formula:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \cdot \sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{12}$$

where X is the set containing all usage-based similarities with $\overline{x}$ as mean value and Y is the set containing all metadata similarities with $\overline{y}$ as mean value.

As described in chapter 4, there are different methods to calculate the usage-based similarity. Pre- and post-context similarities can be calculated by using the Jaccard Coefficient when the pre- and post-contexts are handled as sets (cf. formulas (1) and (2)), or by using the cosine similarity when the pre- and post-contexts are handled as bags (cf. formulas (3) and (4)). Furthermore, the similarity between UCPs can be calculated by using the arithmetic mean (cf.

formula (6)) of the pair-wise UC similarities or by using the median (cf. formula (7)) of these similarities.

As shown in Table 1, the usage of the median for similarity calculation lowers the correlation between usage-based and semantic metadata-based similarity. This is due to the fact that the median often becomes null even when some UCs of the compared UCPs hold a higher similarity value.

	set	**bag**
arithmetic mean	0.32	0.35
median	0.2	0.25

Table 1: Correlation values between semantic metadata-based and usage-based similarity using different methods to calculate the usage-based similarity

When comparing the usage of the Jaccard coefficient (i.e. set) with the cosine similarity (i.e. bag), the cosine similarity results in a slightly better result. This leads to the assumption that when students access an object more than once in one session, this is a characteristic of the context that needs to be considered for similarity calculation.

In general terms a correlation coefficient between 0.1 and 0.3 is described as low [Faller and Lang 2006]. As the median calculations do not seem to be a valid approach because of the tendency to bias the results, it is more promising to look at the mean values. The resulting coefficient can be described as medium and as we regard a large sample of objects, the coefficient can said to be representative although no separate tests for bivariate normality have been undertaken [Bortz 1993]. Moreover, because of the big sample size all correlations are significant on the 5%-level (p=.05).

A further important point is the fact that the considered metadata can only be interpreted as a shallow content representation. The resulting similarity values serve as a tentative 'gold standard' for evaluating the usage-based approach, even though we are aware of the fact that the metadata suffer from the sparsity problem. Because of this it is possible that the correlation between semantic metadata-based similarity and usage-based similarity represents a lower bound for the real correlation between content and usage context. Therefore, it then makes sense to manually compare a chosen subset of learning objects. If the semantic metadata-based similarity does not sufficiently represent the content, we assume to find a higher congruence between semantic metadata-based similarity and usage-based similarity. In the next section we follow this path by comparing the 100 learning object pairs with the highest usage-based similarity.

5.4 Manual comparison of content-based and usage-based object similarity

Since the manual proof for content similarity does not produce an explicit similarity value that can be compared to other content similarity values, we focused on finding the content overlap of two learning objects. Therefore, we accessed the content of the learning object pairs directly and compared them with each other. We found that 92% of the considered learning object pairs showed similarities, 4% were not accessible due to permission rights and only 4% showed no similarity at all.

Many of the checked learning object pairs showed content similarities that were not entailed in the metadata: for example, text documents which handle the same topic such as 'risk factor analysis', 'low energy construction', 'music and architecture' or 'fire safety' or learning object pairs where one of them was an exercise and the other one showed a text about the same topic. In one case we found a pair where both objects refer to a website about graphical algorithms. One of these websites provides the opportunity to browse and see examples of shape generating algorithms (including pictures), while the other one provides the possibility to create and test such kind of algorithms. Some of the learning object pairs refer to different web pages of the same domain. For instance, both objects refer to different entries of the ArchiplanetWiki[4] and show similar buildings. The similarity of these buildings is expressed by different attributes like similar construction date, 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 were not contained in the metadata. Another category of content similarity is about picture similarity. Thus we discovered documents showing similar pictures, e.g. photos, sketches or models of the same building or construction activity like panel cladding. Furthermore, we could also identify learning object pairs containing a geographical similarity of the displayed content, e.g. pairs which represent websites containing pictures or articles of different historical buildings in the same town.

In many cases these content similarities are not entailed in the learning object's metadata but can be detected by the user due to her prior knowledge (e.g. knowing that two buildings were designed by the same architect). As this shows that usage similarities can hint at content similarities that have not been considered so far and enable the system to use user knowledge to enhance the recommendation without forcing the users to explicitly share it, we see it as a motivation to continue on this track.

[4] http://www.archiplanet.org/wiki/Main_Page

6 Usage context profiles for recommendations

In the previous sections we first described how usage context profiles (UCPs) are generated and then evaluated in what way they can give rise to content similarity. In this chapter we will outline how the UCPs can be used to recommend objects, we will therefore describe the three methods that seem most promising to us.

6.1 Ranking objects according to their context similarity

The most obvious approach is to recommend learning objects that have similar UCPs to the UCPs of the objects used in the actual session. Therefore, first the similarities between the objects of the actual session and the not yet used objects are calculated. Then, for each not yet used object, the arithmetic mean of all similarity values this object holds is computed. The objects are then sorted by their similarity and the results with the highest similarity values are shown to the user.

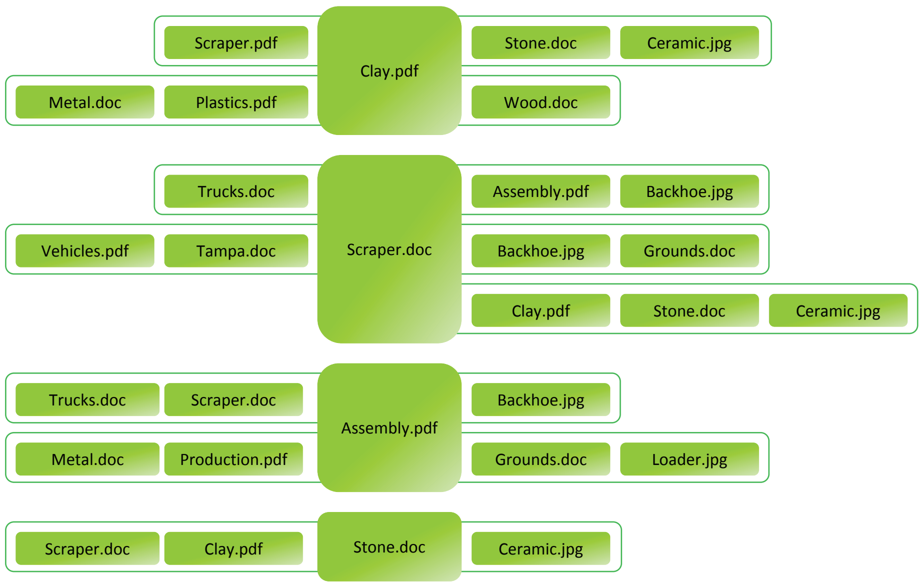

Figure 3: UCPs for the objects *Clay.pdf*, *Scraper.doc*, *Assembly.pdf* and *Stone.doc*

Exemplarily, the UCPs of the learning objects *Clay.pdf*, *Scraper.doc*, *Assembly.pdf* and *Stone.doc* are illustrated in Figure 3. Let's further assume a user looked at the learning objects *Clay.pdf* and *Scraper.doc* in the actual session, as illustrated in Figure 4.

Figure 4: Session of a user

To decide what learning object(s) to recommend, the similarities between the objects that can be recommended and the objects of the actual session must be calculated based on their UCPs.

$$\begin{aligned}
&SimUCP\,(UCP_{Clay.pdf}, UCP_{Assembly.pdf}) = 0,1042\\
&SimUCP\,(UCP_{Clay.pdf}, UCP_{Stone.doc}) = 0,25\\
&SimUCP\,(UCP_{Scraper.doc}, UCP_{Assembly.pdf}) = 0,1528\\
&SimUCP\,(UCP_{Scraper.doc}, UCP_{Stone.doc}) = 0,0556
\end{aligned} \tag{13}$$

These calculations result in the following similarity weights with AC representing the objects of the actual session:

$$\begin{aligned}
&SimUCP\,(AC, UCP_{Assembly.pdf}) = 0,1285\\
&SimUCP\,(AC, UCP_{Stone.doc}) = 0,1528
\end{aligned} \tag{14}$$

Since the similarity of *Stone.doc*'s UCP to AC is a little bit higher than the similarity of *Assembly.pdf*'s UCP to AC, *Stone.doc* gets recommended first. However, using this approach the similarities are quite near each other.

An extension of this approach is to rank the objects also according to the position of the objects they are similar with. That is to say, objects similar to the actual object of the session get ranked higher than, for example, objects similar to the first object of the session. To achieve this ranking, the similarity weights are multiplied with a factor that states the position of the object in the actual session they are similar with.

6.2 Actual session as pre-context of the object to be recommended

Another approach is to compare a user's current usage history with the pre-contexts of all other learning objects and recommend those objects that have highly similar pre-contexts. The basic idea of this approach is that all objects of the actual session together constitute one pre-context of the object that needs to be recommended.

In respect to the previous section, the actual session (AC) that comprises the learning objects *Clay.pdf* and *Scraper.doc* is compared to the pre-contexts of *Assembly.pdf* and *Stone.doc*.

$$\begin{aligned} &simPreUC\left(AC, UC^{pre}_{Assembly.pdf\_1}\right) = \frac{1}{3} \\ &simPreUC\left(AC, UC^{pre}_{Assembly.pdf\_2}\right) = 0 \\ &simPreUC\left(AC, UC^{pre}_{Stone.doc\_1}\right) = 1 \end{aligned} \tag{15}$$

The overall similarity of the learning objects and the actual session is calculated by using the arithmetic mean.

$$\begin{aligned} &simPreUC\left(AC, UC^{pre}_{Assembly.pdf}\right) = \frac{\frac{1}{3}+0}{2} = \frac{1}{6} \\ &simPreUC\left(AC, UC^{pre}_{Stone.doc}\right) = 1 \end{aligned} \tag{16}$$

Assembly.pdf comprises two usage contexts from which only one pre-context contains an object of the actual session, *Stone.doc* comprises one usage context whose pre-context contains all objects of the actual session. Therefore, *Stone.doc* is assumed to be more similar to the actual context and gets recommended first.

If more than one pre-context of a learning object holds objects of the actual session, the overall similarity of the learning object and the actual session can also be calculated using the median or by choosing the highest similarity weight.

6.3 Search queries holding UCPs

The last approach is to also generate UCPs for search query terms and use them for the ranking of search results. When a user enters a query term, the pre-contexts of that query term are compared to the user's current usage history. Objects from the query term's post-contexts whose corresponding pre-contexts are similar to the user's usage history are ranked higher in the list of search results.

Exemplarily, the UCP of the query *"Smith"* is illustrated in Figure 5. *"Smith"* is a highly common name. In the MACE system a lot of buildings and sculptures are created or built by persons with this surname but *"Smith"* can also be part of a name of a structure, e.g. in *"Smith Private Airport"* or in *"Smiths Grove Presbyterian Church"*. Here, we can distinguish between sessions in which a user was interested in the work of *"Tony Smith"*, an architect, sculptor and noted theorist on art, and sessions in which users looked at buildings from *"Adrian Smith"*, one of the most recognised architects in the world who designed notable super-tall skyscrapers.

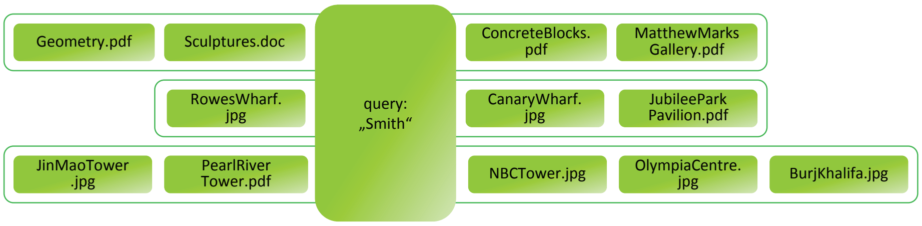

Figure 5: UCP for search query "Smith"

As example, if a user looks at the learning object *JinMaoTower.jpg* and then searches for *"Smith"* as depicted in figure 6, the system assumes that the user is interested in the same *"Smith"* that is meant in the UCP given in figure 5 and is therefore interested in the same learning objects as the user of the last shown usage context of figure 5. Therefore, the learning objects *NBCTower.jpg*, *OlympiaCentre.jpg* and *BurjKhalifa.jpg* will be ranked higher when the recommendations are computed for the user of figure 6's query.

Figure 6: Session of a user containing a query

This approach can also deal with different competences of students which are assumed to be implicitly given by the objects a user looked at in the actual session. An advanced student searching for "construction sky scraper" might be looking for other learning material than a beginner. However, since the pre-contexts of their sessions most probably hold different learning objects, they are not assumed to be similar even when searching for the same query terms and the system will recommend different learning objects.

7 Conclusion and Future Work

In this paper we introduced a new idea for recommending learning objects based on their usage contexts. The hypothesis behind our approach that usage context-based similarity is an indication of content similarity was supported by our results. This motivates us to further develop this approach.

First, we aim at improving the quality of the usage-based similarity calculation. Currently, we do not consider the length of usage contexts which most probably will influence the usage-based similarity. A pair sharing 3 of 4 occurring objects could be deemed less similar than a pair sharing 6 out of 8 objects although they both have a similarity of 0.75. Additionally, short usage contexts could miss the relevant objects and therefore display a tendency towards low similarities. Therefore, longer usage contexts could be weighed higher. Another approach would be to only consider a set number of objects before and after the object currently in use, since objects that are more distant from the actual object might be less significant for its usage context description. Thereby, the optimal count of objects for pre- and post-context needs to be figured out in experiments and might differ depending on the domain. Additionally, the order of the learning objects could be taken into account when calculating the usage-based similarities. This might improve the similarity calculations but requires a huge dataset containing user sessions. In contrast to this approach, it is also possible to give up the distinction between pre- and post-context and just consider the 'whole context'. This might improve recommendations in sessions where for example a student views learning objects about buildings built from a special architect or located in the same area, but it might worsen the recommendations when learning objects build upon each other. Furthermore, we could distinguish between different actions carried out on the objects. Currently we handle all actions just as 'accesss' and do not differentiate between, for example, just viewing a document or changing its metadata. To enable a better comparison of content and metadata similarity, we will in the next steps base the content similarity not only on the objects' metadata but also use the textual content of the objects. Thereafter, we aim at including our findings into the MACE project to thus enable a large scale evaluation with real users.

References

[Adomavicius and Tuzhilin 2005] Adomavicius, G., Tuzhilin, A.: "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), 2005, 734-749.

[Bortz 1993] Bortz, J.: Statistik [Statistics]. Springer, Heidelberg, 1993: Springer Verlag.

[Burke 2007] Burke, R.: "Hybrid recommender systems: Survey and experiments". The Adaptive Web, 2007.

[Candillier et al. 2009] Candillier, L., Jack, K., Fessant, F., Meyer, F.: "State-of-the-Art Recommender Systems". In: Chevalier, M., Julien, C., Soule-Dupuy, C. (eds.): Collaborative and Social Information Retrieval and Access - Techniques for Improved User Modeling. Idea Group Publishing, 2009, 1-22.

[Dey 2001] Dey, A.K.: "Understanding and using context". Personal and Ubiquitous Computing, 5(1), 2001, 4-7.

[Faller and Lang 2006] Faller, H. and Lang, H.: Medizinische Psychologie und Soziologie. Springer, Heidelberg. 2nd fully reworked edition, 2006.

[Friedrich et al. 2009] Friedrich, M., Niemann, K., Scheffel, M., Schmitz, H.-C., Wolpers, M.: "Object Recommendation based on Usage Context". Workshop:

Context-aware Recommendation for Learning at the STELLAR Alpine Rendez-Vous, Garmisch-Partenkirchen, Germany, November 30 - December 2, 2009. available at http://www.stellarnet.eu/index.php/download_file/-/view/572

[Heyer et al. 2006] Heyer, G., Quasthof, U., Wittig, T.: Text Mining: Wissensrohstoff Text. Konzepte, Algorithmen, Ergebnisse. W3L-Verlag, Herdecke, 2006.

[IEEE LOM Standard] LOM IEEE Standard for Learning Object Metadata, IEEE Std 1484.12.1, 2002.

[Linden et al. 2003] Linden, G., Smith, B., York, J.: "Amazon.com Recommendations - Item-to-Item Collaborative Filtering". IEEE Internet Computing 7 (1), 2003, 76-80.

[Matsumoto 2003] Matsumoto, Y.: "Lexical Knowledge Acquisition". In: Mitkov, R.: The Oxford Handbook of Computational Linguistics. Oxford University Press, Oxford, 2003, 395-413.

[Melville et al. 2002] Melville, P., Mooney, R.J., and Nagarajan, R.: "Content-Boosted Collaborative Filtering for Improved Recommendations". Proceedings of the 18th National Conference for Artificial Intelligence, 2002.

[Meteren and Someren 2000] Meteren, R. van, Someren, M. van: "Using content-based filtering for recommendation". Proceedings of MLnet/ECML2000 Workshop, Barcelona, Spain, 2000.

[Mobasher et al. 2004] Mobasher, B., Jin, X., and Zhou, Y.: "Semantically Enhanced Collaborative Filtering on the Web". Berendt, B. et al. (eds.): Web Mining: From Web to Semantic Web. LNAI Volume 3209, Springer, 2004.

[Pazzani et al. 1996] Pazzani, M., Muramatsu, J. and Billsus, D.: "Syskill and Webert: Identifying interesting Web sites". Proceedings of Thirteenth National Conference on Artificial Intelligence, AAAI'96, Portland, OR, 1996, 54-61.

[Pearson 1907] Pearson, K.: On further methods of determining correlation. Cambridge University Press, London, 1907.

[Porter 1980] Porter, M.-F.: "An algorithm for suffix stripping". Program 14: 1980, 130-137.

[Sarwar et al. 2001] Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: "Item-Based Collaborative Filtering Recommendation Algorithms". Proceedings of the 10th International World Wide Web Conference, ACM Press, 2001, 285-295.

[Saussure 1986] Saussure, F. de: Course in General Linguistics. Open Court Publishing, Chicago (re-print 1986).

[Smyth and Cotter 2000] Smyth, B. and Cotter, P.: "A Personalized TV Listings Service for the Digital TV Age". Knowledge-Based Systems 13: 53-59.

[Stefaner et al. 2007] Stefaner, M., Dalla Vecchia, E., Condotta, M., Wolpers, M., Specht, M., Apelt, S., Duval, E.: "MACE - enriching architectural learning objects for experience multiplication". In: Duval, E., Klamma, R., Wolpers, M. (eds.): Creating New Learning Experiences on a Global Scale, LNCS, vol. 4753, Springer, Heidelberg, 2007, 322-336.

[Vuorikari and Berendt 2009] Vuorikari, R. and Berendt, B: "Study on contexts in tracking usage and attention metadata in the field of multilingual Technology Enhanced Learning". In: Fischer, S., Maehle, E., and Reischuk, R. (Eds.): Informatik 2009. Im Focus das Leben: Beitrge der 39. Jahrestagung der Gesellschaft fr Informatik e.V. (GI), 28.9. - 2.10.2009 in Lbeck. GI, Bonn, 2009, 1654-1663.

[Wolpers et al. 2007] Wolpers, M., Najjar, J., Verbert, K., Duval, E.: "Tracking Actual Usage: the Attention Metadata Approach". Educational Technology & Society 10 (3), 2007, 106-121.

[Zimmermann 2007] Zimmermann, A.: "Context Management and Personalisation: A Tool Suite for Context- and User-Aware Computing". Dissertation thesis, RWTH Aachen, 2007.