KNOWLEDGECHECKR: Intelligent Techniques for Counteracting Forgetting

Martin Stettinger[1] and **Trang Tran**[2] and **Ingo Pribik**[3] and **Gerhard Leitner**[4]
and **Alexander Felfernig**[2] and **Ralph Samer**[2] and **Müslüm Atas**[2] and **Manfred Wundara**[5]

Abstract. Existing e-learning environments primarily focus on the aspect of providing intuitive learning contents and to recommend learning units in a personalized fashion. The major focus of the KNOWLEDGECHECKR environment is to take into account forgetting processes which immediately start after a learning unit has been completed. In this context, techniques are needed that are able to predict which learning units are the most relevant ones to be repeated in future learning sessions. In this paper, we provide an overview of the recommendation approaches integrated in KNOWLEDGECHECKR. Examples thereof are *utility-based recommendation* that helps to identify learning contents to be repeated in the future, *collaborative filtering* approaches that help to implement session-based recommendation, and *content-based recommendation* that supports intelligent question answering. In order to show the applicability of the presented techniques, we provide an overview of the results of empirical studies that have been conducted in real-world scenarios.

1 Introduction

The concept of *inverted learning* is gaining momentum in different types of educational settings [13]. The one-way delivery of information is replaced by face-to-face interaction with work in small groups and a clarification focus. The major focus of the KNOWLEDGECHECKR environment[6] is to provide intelligent techniques that support inverted learning scenarios. On the basis of recommendation functionalities, the system is able to propose learning content and questions that enable students (learners) to better focus on topics where they need to catch up. Also, teachers have better insights into the performance of students and thus can immediately adapt their focus in onsite teaching sessions. KNOWLEDGECHECKR is based on recommendation approaches that support learning-related tasks such as *scheduling repetition cycles*, *recommending questions and knowledge units*, and *supporting Q&A scenarios*. In this paper, we provide an overview of KNOWLEDGECHECKR recommendation approaches and report the results of empirical studies that show in which way recommenders can improve the quality of learning.

A major focus of KNOWLEDGECHECKR is the provision of techniques that help counteracting forgetting [10]. Our motivation to develop such techniques is based on an empirical study conducted with *N=70 companies* in Austria from diverse domains such as financial services, software development, production, transport, telecommunications, and higher education. Participants of the study ranged from lower, medium, up to higher management. On an average, the study participants reported to spend around *5 hours per week* to answer questions of colleagues although these colleagues should already be able the answer the questions, since they visited topic-related educational programs. The major related knowledge categories are summarized in Table 1.

We experienced similar results in university contexts where, for example, PhD project relevant knowledge has to be "manually" transferred a couple of times to assure that the knowledge is available when needed. Examples thereof are issues such as how to write papers, what are the correct formulations to explain an example, and how to perform a logical proof. In our study, 91.43% of the participants agreed that mechanisms that help counteracting the forgetting of company-relevant knowledge are extremely important and 97.24% mention that a personalized knowledge transfer for counteracting forgetting is important for the company. These were major motivations that lead to the development of KNOWLEDGECHECKR.

category	example	support
products	what are the new product features?	43.48%
norms	how to communicate with colleagues?	33.33%
laws	general data protection regulation	36.23%
internal processes	how to behave during an evacuation?	34.78%
business processes	production, customer complaints	65.22%

Table 1. Major knowledge categories which could profit from techniques for counteracting forgetting.

The major contributions of this paper are the following: we show how recommendation technologies can be applied to (1) counteract forgetting processes, (2) recommend relevant learning contents, and (3) support Q&A scenarios in an intelligent fashion. Furthermore, we report initial results of empirical studies that show the applicability and business relevance of our approach. An overview of existing industrial and university-level deployments of KNOWLEDGECHECKR is provided in Table 2.

domain	description	#users
public administration	3 municipalities in Austria	1.000
higher education	5 universities in Austria	100
hardware production	Flex Austria	100
medicine	1 hospital in Austria	200

Table 2. Existing real-world deployments of KNOWLEDGECHECKR.

[1] SelectionArts, email: m.stettinger@selectionarts.com
[2] Graz University of Technology, email: {ttran, afelfernig, rsamer, muatas}@ist.tugraz.at
[3] Flex Austria, email: ingo.pribik@flex.com
[4] University of Klagenfurt, email: Gerhard.Leitner@aau.at
[5] Information Technology Department, Villach, Austria, email: manfred.wundara@villach.at
[6] A responsive HTML-5 application: www.knowledgecheckr.com

The remainder of this paper is organized as follows. In Section 2, we provide an overview of KNOWLEDGECHECKR recommendation technologies. Thereafter, in Section 3 we provide examples of the system user interface and discuss the provided functionalities. In Section 4, we summarize the results of user studies that show the applicability and business relevance of the KNOWLEDGECHECKR environment. In Section 5, we provide an overview of related work. We conclude the paper with a discussion of future work in Section 6.

2 Recommendation Technologies

KNOWLEDGECHECKR recommendation approaches support different goals which can be summarized as (1) recommending *question sequences* [17] (following the paradigm of test-enhanced learning [18]), (2) recommending *questions for repetition purposes* in order to be able to counteract forgetting [10], and (3) recommending further *learning units* that might be of interest for the user [16]. In KNOWLEDGECHECKR, such scenarios are supported by recommendation techniques. Questions are recommended (1) for *learning purposes* when users start to engage in a learning process and (2) for *repetition purposes* after initial learning has been completed. The former is supported by *session-based recommendation* that guides a user through a learning process with increasing question complexity, the latter by a *utility-based recommendation approach* that identifies questions with a higher probability of already being forgotten.

Session-based Recommendation. In many application scenarios, users of KNOWLEDGECHECKR prefer to stay anonymous especially when using the system the first time. In such scenarios, not much information regarding the knowledge level and domain-specific experiences of a user is available. In KNOWLEDGECHECKR, a session-based recommendation approach [20] is provided where already completed similar (nearest neighbor) sessions (sessions that show a similar user interaction behavior) are applied to recommend the next questions to the current (anonymous) user. In KNOWLEDGECHECKR, the session-based approach is based on collaborative filtering [4]. A simplified example of the approach is depicted in Table 3, where sequences of questions answered by other users are stored in a log. The underlying idea is to find sequences (rankings of questions) which are easy to complete where a question ranking is determined on the basis of the question selection behavior of one or a set of nearest neighbors. This goal can be achieved with collaborative filtering, since (implicit) dependencies between questions (e.g., question x is a precondition of y) can be taken into account by learning from similar sessions.

In the example depicted in Table 3, the user in the current session s_c has already successfully answered the questions $\{q_1, q_2, q_3\}$ in the order $[q_3, q_2, q_1]$ but did not answer the questions $\{q_4, q_5\}$. In this context, KNOWLEDGECHECKR tries to figure out the ordering (ranking) in which the unanswered questions should be presented to the user. Following a session-based recommendation approach, the system identifies the *n-nearest neighbors* and derives a question sequence that might be of relevance for the user.

In KNOWLEDGECHECKR, the most similar sessions are used to predict questions of potential relevance to the current (anonymous) user. First, the similarity between the session s_c (the current session) and a session s_a can be determined on the basis of Formula 1 where $correct(q_i, s_x)$ indicates whether question q_i has been correctly answered in session s_x and Q denotes the complete set of questions in a specific KNOWLEDGECHECKR application. In our example, $sim(s_c, s_1) = 1.0$ since there is a complete overlap in the terms of correct questions already answered in s_c.

$$sim(s_a, s_c) = \frac{|q_i \in Q : correct(q_i, s_a) \wedge correct(q_i, s_c)|}{|q_i \in Q|} \quad (1)$$

Formula 2 helps to determine an overall evaluation of question q in the context of the current session s_c with regard to its relevance for the user. In this context, $r(q, s_i)$ denotes the ranking of question q in the session s_i where SNN denotes the n-nearest neighbor sessions of session s_c (the current session). Assuming $SNN = \{s_1\}$ (for the purpose of our example, we follow a 1-nearest neighbor approach), the overall evaluations of the two up-to-now unanswered questions would be $eval(q_4, s_c) = 5$ and $eval(q_5, s_c) = 4$ where $r(q_4, s_1) = 5$ and $r(q_5, s_1) = 4$.

$$eval(q, s_c) = \frac{\Sigma_{s_i \in SNN(s_i \neq s_c)} r(q, s_i) \times sim(s_i, s_c)}{|SNN|} \quad (2)$$

Finally, Formula 3 helps to determine a prediction for the ranking of question q in the context of session s_c. The rank of the last question answered within the scope of session s_c is 3, i.e., $currentqrank(s_c) = 3$. Furthermore, $rank(eval(q_5, s_c)) = 1$ and $rank(eval(q_4, s_c)) = 2$. As a consequence, $pred(q_4, s_c) = 3 + 2$ and $pred(q_5, s_c) = 3 + 1$ which results in the recommendation of q_5 as the next question to be posed to the user in s_c. If a user provides a wrong answer within the scope of a learning session, the corresponding question is dropped from the list of recommended questions for a specific time period (the default setting is 20 minutes).

$$pred(q, s_c) = currentqrank(s_c) + rank(eval(q, s_c)) \quad (3)$$

session	q_1	q_2	q_3	q_4	q_5
s_1	3	2	1	5	4
s_2	2	1	3	?	?
s_3	1	2	4	5	3
s_4	3	2	4	1	5
s_c	3	2	1	?	?

Table 3. A simple interaction log in session-based recommendation. The table entries represent session-specific orderings of posed questions q_i, question marks represent still unknown orderings.

In our simplified example, the nearest neighbor of session s_c is session s_1, since the order in which questions have been answered by the users are the same. The ordering for the next questions that will be recommended by the system is $[q_5, q_4]$. This session-based collaborative approach is used to support the ramp-up in situations where users interact with the learning application maybe the first time and prefer to apply KNOWLEDGECHECKR in an anonymized fashion. After the knowledge level of individual users becomes more transparent, KNOWLEDGECHECKR is able to switch to a *utility-based recommendation approach* where the utility of individual questions is estimated depending on time intervals since questions have been answered correctly the last time.

Utility-based Recommendation. In KNOWLEDGECHECKR, utility-based recommendation [8] is applied to implement functions that help *counteracting forgetting*. The underlying idea is that answering the same question repeatedly within specific time intervals helps to consolidate the learning material [10]. Furthermore, utility-based recommendation determines a ranking where the most

relevant repetitions are presented first. Users under time pressure can thus focus on the most relevant topics.

The functions used to determine questions of relevance are the following. The *relevance (rel)* of a question q for a user u is defined as the complement of the share of correct answers compared to the number of answers to question q. This factor is weighted by a time aspect, i.e., the more days already passed since the last time the question q has been answered by user u ($\#dayssince(q,u)$), the higher the corresponding relevance since the probability becomes higher that a user is not able to answer the question correctly. In this context, $\#daystoforget(q)$ represents the assumption that after x days the correct answer will have been forgotten. This value can be pre-specified when defining a question or approximated based on historical data. Equation 4 represents a basic way of ranking questions.

$$rel(q,u) = (1 - \frac{\#correctans(q,u)}{\#totalans(q,u)}) \times \frac{\#dayssince(q,u)}{\#daystoforget(q)} \quad (4)$$

Equation 5 is an extension of Equation 4 which additionally takes into account the aspects of *importance* and *complexity* of a question (see the Equations 6 and 7). This means that the higher the *importance* of a question and the lower the *complexity* of a question, the higher the probability that this question will be recommended to the current user. In this context, *complexity* and *importance* can be estimated on the basis of *data related to the global share of correct answers to a specific question q* and the *average feedback of users regarding the importance level of question q*. The higher the importance and the lower the complexity, the higher the relevance of the question for a specific user. The underlying idea is that knowledge about simple contents/questions is the precondition for answering more complex ones.

$$rel'(q,u) = rel(q,u) \times \frac{importance(q)}{complexity(q)} \quad (5)$$

$$complexity(q) = 1 - (\frac{\#correctans(q)+1}{\#totalans(q)+1}) \quad (6)$$

$$importance(q) = \frac{\Sigma_{i=1}^{\#feedbacks(q)} feedbackval(i)}{\#feedbacks(q)+1} \quad (7)$$

Content-based Recommendation. In learning apps with a large amount of questions, content-based recommendation [16] is used to support intelligent Question & Answering (Q&A) which is an orthogonal way to exploit questions and answers stored in KNOWLEDGECHECKR. The underlying scenario is, for example, the following: a user of a KNOWLEDGECHECKR learning application on *model-based diagnosis* is currently preparing for the exam related to the course. Just two hours before the exam, the question comes to his/her mind, *which diagnosis approach can guarantee the retrieval of minimal cardinality diagnoses*. Such queries can be entered to the search interface of the system. On the basis of the query string (e.g., *which diagnosis approach does support minimal cardinality?*), a content-based recommender determines the similarity between question features and corresponding query features. The answers to questions most similar to the query are then shown to the current user. In order to determine the similarity between the query q_c posed by the current user and a question $q_i \in Q$, the following basic similarity metric is applied (see Formula 8) [5, 16].

$$sim(q_i,q_c) = \frac{2 \times |features(q_i) \cap features(q_c)|}{|features(q_i)| + |features(q_c)|} \quad (8)$$

A simplified example of the content-based recommendation approach in KNOWLEDGECHECKR is given in Table 4. Note that such recommendation services are provided in individual learning applications but as well on the global level to support situations where users are not completely sure which of the available learning applications could answer their questions. In the following example (Table 4), we assume that the user poses the query in the context of a specific learning application (*model-based diagnosis*). In our example, the question with the highest similarity to the query q_c is q_6 ($sim(q_6,q_c) = \frac{2 \times 3}{9} = 0.67$). Consequently, the answer specified for q_1 would be shown as answer for q_c. For further details on content-based recommendation approaches we refer to [16].

question	question features (and question features)
q_1	conflict, algorithm, QuickXPlain
q_2	FastDiag, algorithm, time, complexity
q_3	FastDiag, algorithm, space, complexity
q_4	hitting, set, search, tree, breadth, first, minimal, cardinality
q_5	hitting, set, conflict, symmetry
q_6	minimal, cardinality, diagnosis, search
$query(q_c)$	diagnosis, approach, support, minimal, cardinality

Table 4. KNOWLEDGECHECKR: a simple example of a content-based recommendation setting.

3 KNOWLEDGECHECKR User Interfaces

KNOWLEDGECHECKR provides the possibility to create individual learning apps (see Figure 1) that consist of contents such as movies and slides and a corresponding set of questions that can be used for personal self tests (knowledge checks), learning sessions, exercises, exams, and competitions. The supported question types are *multiple choice*, *sequencing tasks*, *text completion tasks*, and *image analysis tasks* (see also Figure 2). For each question type, a corresponding explanation can be defined that is shown (if activated) if a user is not able to provide a correct answer to a question.

The discussed recommendation approaches (see Section 2) are useful especially in the context of learning sessions where users try to answer questions in order to improve their knowledge level in specific categories. Since questions have different knowledge levels, the recommender system helps a user to focus on questions with a high probability of being answered before being forwarded to more complex ones. Figure 1 provides an overview of a KNOWLEDGECHECKR list of learning applications – one example of such an application is the *Anatomy Guide* app which is used in medical domains. Registered users dispose of the additional service of *repetitive recommendations* where questions already posed in the past are posed again in order to achieve the goal of counteracting the *forgetting curve* [15]. Each learning app can provide learning content in a personalized fashion and also proposes questions a user should try to answer in the next learning iteration [18].

As already mentioned, KNOWLEDGECHECKR provides different ways of asking questions. Figure 2 provides an example of an image recognition task where the task of a user is to answer a medical question by selecting the corresponding image areas which represent the answer. If a question could not be answered correctly, a corresponding explanation can be shown – in the case of images, an explanation is a visualization of the correct answer areas in the image (including a textual explanation as to why the shown area is the correct one).

Finally, KNOWLEDGECHECKR includes a user interface where the expertise of the community and also of individual users can be
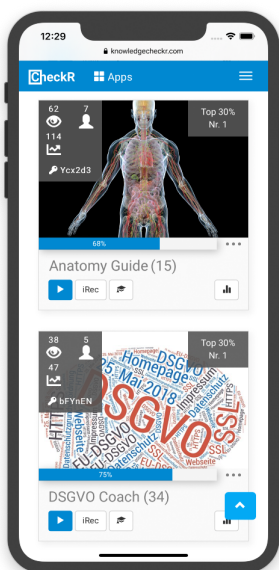
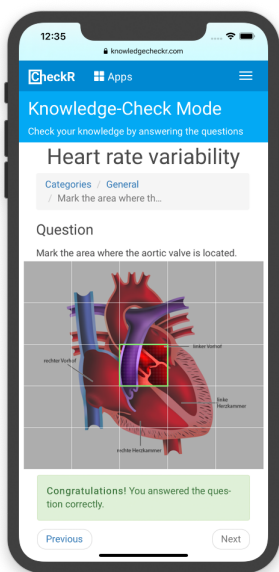**Figure 1.** KNOWLEDGECHECKR view on learning apps.

Figure 2. KNOWLEDGECHECKR graphical interaction mode (find the areas in question): an example from the medicine domain where heart parts in question have to be identified.

analyzed. The interface supports a fine-grained analysis of critical knowledge areas where there is a need to improve the community knowledge or the knowledge of individual users (if the parametrization of system allows this). The analysis section entails a motivational aspect since, for example, the personal comparison with the whole community immediately leads to more system interaction with the goal to be at least as good or even better than the average performance of the community. KNOWLEDGECHECKR also provides

mechanisms to directly update participants of a learning application with regard to new contents and questions. This update channel can be configured in terms of the way notifications are explained (see Section 4) and the frequency of knowledge updates.
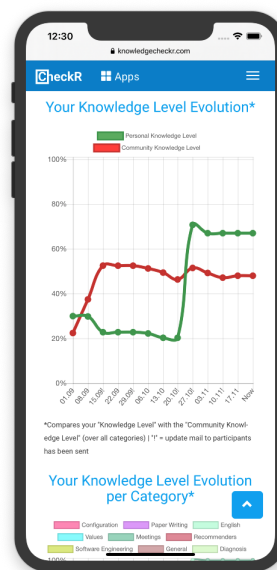

Figure 3. KNOWLEDGECHECKR view on the development of the personal and community knowledge level.

4 User Studies and Benefits

As already mentioned, KNOWLEDGECHECKR has already been deployed and is applied in a couple of application scenarios (see Table 2). The system is applied for various purposes out of which we will discuss a couple of aspects in the following.

Public Administration. In public administration, the system is applied in e-learning scenarios related to topics such as *safety and security*, *dealing with computers*, *programming best practices*, *requirements engineering best practices*, and *sensitization of citizens*. Whereas the former scenarios focus on knowledge transfer directly to employees of the public administration, the latter one focuses on knowledge transfer between public administration and citizens. Examples thereof are topics such as *healthy eating behavior*, *environmental protection*, and *first aid*. Especially KNOWLEDGECHECKR *competitions* can be regarded as a question-driven learning channel [18] where the questions (and corresponding answers) are a major mean to increase the sensitiveness of citizens with regard to the mentioned topics. Motivation in this context is not self-intrinsic and must be stimulated on the basis of remuneration mechanisms such as prizes provided by companies.

We have conducted a *usability study* that focused on public administration end-users of the KNOWLEDGECHECKR environment. The questionnaire was based on the *System Usability Scale* (SUS) [1] with N=20 participants providing feedback on the usability of the system in the context of the mentioned administration-internal applications. Overall, the participants of the study provided positive feedback regarding general usability aspects summarized via SUS questionnaire (see Figure 4). Benefits from the application of KNOWLEDGECHECKR in the public administration are (1) reduced efforts
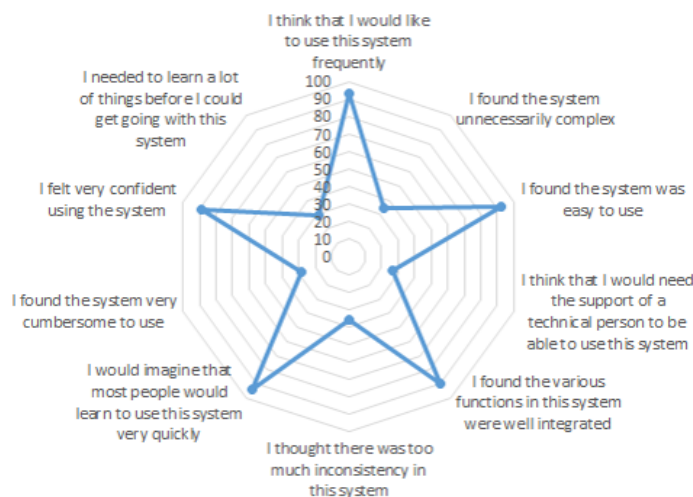
Figure 4. System Usability Scale (SUS) [1] evaluation in public administration (evaluation scale 0: *strongly disagree* .. 100: *strongly agree*).

for managers to keep their team up-to-date and (2) the avoidance of cost-intensive mistakes. On an average, time savings due to reduced "update efforts" were reported to be around *2-3 weeks per year*.

University Courses. Experiences from the application of KNOWL-EDGCHECKR in the university context provide similar results. Leaders of research teams experience similar effort reductions (*2-3 weeks per year*) related to the update of their students. An example thereof are systematic updates regarding basics of their research topics, practices when writing papers, and criteria and strategies for successfully completing their PhD studies. In the context of university courses, KNOWLEDGECHECKR has been applied in computer science teaching. The related learning applications provide an additional means for students to prepare for courses and check their knowledge level in different categories. This is an extremely important feature for students since the system enables them to focus their learning effort on relevant topics in which they have to catch up.

When used as an additional means to understand course topics and to prepare for an exam, around 10% of the students use KNOWLEDGECHECKR from the very beginning throughout the whole course. Furthermore, around 80% of the students primarily use the system directly ahead of an exam with goal to check their knowledge level and to be optimally prepared. Finally, on an average 20% of the students do not use KNOWLEDGECHECKR at all. Consequently, the usage in scenarios where users are not forced to use the system (contrary to industrial contexts), follows an *80-20 rule*. Similar percentages have been observed in four different courses.

Finally, we measured the prediction quality of the utility-based approach discussed in Section 2, since in most of the university courses students are used to sign-in (non-anonymous mode) to interact with the system. On an average, the prediction quality of the utility-based recommendation approach in terms of precision [7] is around 0.9 overall all learning apps, i.e., in nearly 90% of the cases, the system manages to predict the item that will also be chosen by the user.

In a course with N=350 Computer Science students, we measured potential increases in student output quality that can be explained by the application of KNOWLEDGECHECKR. This evaluation was performed in the context of the course *Object-oriented Analysis and Design* where topics and support team did not change over three

years and a significant improvement could be observed in terms of student grades. Compared to previous years, there was a significant reduction of negative grades (evaluation "insufficient") (6.42%) and a significant increase of excellent grades (evaluation "very good") after KNOWLEDGECHECKR has been provided (5.5%). The previously discussed *80-20 rule* could be confirmed. In this scenario, 350 users answered around 50.000 questions which means around 143 answered questions per user.

User Motivation and Explanations. User motivation is an important aspect since it is the precondition for a wide-spread application of the system. First, KNOWLEDGECHCKR provides functionalities that help to inform participants of learning applications in cases where new contents have been entered into the system or existing ones have been updated. It is important to know that depending on the learning domain, the formulation of related persuasive explanations (arguments) should differ. In *high-involvement learning domains* (users have a high interest in understanding the learning content) such as university courses, persuasive explanations should follow the line of *socialness* (in the line of the persuasion dimensions proposed in Cialdini [3]). An example thereof is the explanation *other users who answered the following questions correctly, managed to pass the exam in 95% of the cases*. Vice-versa, in *low-involvement learning domains* (users have a low or nearly no interest in understanding the learning content), such as *learning fire protection rules*, instead of socialness, *time-related arguments* seem to be more important. An example of such an argument is the following: *the answering of the following six questions takes only three minutes.*

5 Related Work

Recommender Systems. Recommender systems are used to retrieve items of relevance for the user from a large and potentially complex item assortment [9]. *Collaborative filtering* [4] exploits the preferences of so-called nearest neighbors, i.e., users with preferences similar to the current user, and recommends items that have already been consumed (and rated positively) by the nearest neighbors but not by the current user. *Utility-based recommendation* is based on a utility analysis of different items using a utility function [8]. *Content-*

based recommender systems [16] focus on evaluating the similarity between a new item a user did not notice up to now and the user profile derived from previous item consumptions. Finally, *knowledge-based recommender systems* [2, 6] focus on the recommendation of items characterized by attributes where the recommendation knowledge is often described either in terms of constraints or in terms of similarity metrics.

Recommender Systems in e-Learning. Overall, the application of recommender systems in e-learning scenarios primarily focuses on the personalized provision of learning content – for an overview of recommender systems in e-learning we refer to [11, 12]. In KNOWLEDGECHECKR, collaborative filtering is applied in the context of providing (session-based) recommendation services to anonymous users, utility-based recommendation is used to implement functionalities to counteract forgetting, and content-based recommendation is included to provide basic Q&A services. Thus, the application of recommender systems is extended by specifically taking into account requirements of inverted learning processes [13].

Counteracting Forgetting. Research focused on the analysis of forgetting processes can primarily be found in the psychological literature [10, 15, 18]. For example, Pashler et al. [15] analyze possibilities to enhance learning processes and retarding forgetting and point out clear improvements that can be achieved when providing related technologies. In the context of recommender systems, forgetting processes are primarily taken into account in models that represent (long-term) preference shifts [14]. In this context, knowledge about forgetting processes is exploited to infer preference shifts whereas knowledge about forgetting processes in KNOWLEDGECHECKR is used to develop strategies that help to counteract forgetting.

Explanations and Recommender Systems. The selection of explanation types implemented in a recommender system strongly depends on the overall goal of the explanation [19]. Examples of such goals are *increasing the purchase probability of specific items*, *increasing a user's item domain knowledge*, *persuading a user to take specific actions*, and *increasing the trust level of a user*. A major focus of explanation approaches in the KNOWLEDGECHECKR environment is (1) to persuade for system usage and (2) to provide explanations in situations where a user is not able to answer a question correctly. In both cases, the overall impact of these explanations is that users increase their domain-specific knowledge.

6 Conclusions & Future Work

KNOWLEDGECHECKR is a learning environment that supports question-enhanced learning processes that enable counteracting forgetting on the basis of recommendation technologies. In this paper, we provided an overview of the algorithmic approaches integrated in KNOWLEDGECHECKR and also discussed example aspects of the user interface provided by the system. In this context, we also reported results from empirical studies conducted on the basis of real-world deployments of KNOWLEDGECHECKR.

Our plans for future work include further extensions of KNOWLEDGECHECKR. First, we plan to integrate automated video segmentation functionalities that help to cut sequences from videos that best help to explain content/question-specific aspects. Second, we will further improve the predictive quality of question recommendations. Third, sentiment learning from chats will be used to estimate more precisely different dimensions such as quality and complexity of a question. Furthermore, we plan to include services such as the optimization of the working load of a user to achieve specific goals. For example, we will provide mechanisms that recommend learning items that have to be "consumed" to achieve specific learning goals such as *successfully passing an exam with minimum effort*. Finally, we plan to analyze in more detail the impact of repetitions on the personal knowledge level evolution.

ACKNOWLEDGEMENTS

REFERENCES

[1] A. Bangor, P. Kortum, and J. Miller, 'An Empirical Evaluation of the System Usability Scale', *International Journal of Human–Computer Interaction*, **24**(6), 574–594, (2008).

[2] R. Burke, 'Knowledge-based Recommender Systems', *Encyclopedia of Library & Information Systems*, **69**(32), 180–200, (2000).

[3] R. Cialdini, *Influence: The Psychology of Persuasion*, Pearson, 2014.

[4] M. Ekstrand, J. Riedl, and J. Konstan, *Collaborative Filtering Recommender Systems, Foundations and Trends in Human-Computer Interaction*, now Publishers, 2010.

[5] A. Felfernig, L. Boratto, M. Stettinger, and M. Tkalcic, *Group Recommender Systems – An Introduction*, Springer, 2018.

[6] A. Felfernig and R. Burke, 'Constraint-based Recommender Systems: Technologies and Research Issues', in *ACM International Conference on Electronic Commerce*, pp. 17–26, Innsbruck, Austria, (2008).

[7] J. Herlocker, J. Konstan, L. Terveen, and J. Riedl, 'Evaluating Collaborative Filtering Recommender Systems', *ACM Transactions on Information Systems*, **22**(1), 5–53, (2004).

[8] S. Huang, 'Designing utility-based recommender systems for e-commerce: Evaluation of preference-elicitation methods', *Electronic Commerce Research and Applications*, **11**(4), 398–407, (2011).

[9] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender Systems – An Introduction*, Cambridge University Press, 2010.

[10] S. Kang, 'Spaced repetition promotes efficient and effective learning: Policy implications for instruction', *Policy Insights from the Behavioral and Brain Sciences*, **3**(1), 12–19, (2016).

[11] A. Klasnja-Milicevic, M. Ivanovic, and A. Nanopoulos, 'Recommender systems in e-learning environments: a survey of the state-of-the-art and possible extensions', *AI Review*, **44**(4), 571–604, (2015).

[12] C. Krauss, R. Chandru, A. Merceron, and T. An, 'You Might Have Forgotten This Learning Content - How the Smart Learning Recommender Predicts Appropriate Learning Objects', *International Journal on Advances in Intelligent Systems*, **9**(3–4), 472–484, (2016).

[13] A. Lakmal and P. Dawson, 'Motivation and cognitive load in the flipped classroom: definition, rationale and a call for research', *Higher Education Research & Development*, **34**(1), 1–14, (2015).

[14] P. Matuszyk, J. Vinagre, M. Spiliopoulou, A. Jorge, and J. Gama, 'Forgetting techniques for stream-based matrix factorization in recommender systems', *Knowledge and Inf.Sys.*, **55**(2), 275–304, (2018).

[15] H. Pashler, D. Rohrer, N. Cepeda, and S. Carpenter, 'Enhancing learning and retarding forgetting: choices and consequences', *Psychonomic Bulletin & Review*, **14**(2), 187–193, (2007).

[16] M. Pazzani and D. Billsus, 'Content-based recommendation systems', in *The Adaptive Web*, eds., P. Brusilovsky, A. Kobsa, and W. Nejdl, volume 4321 of *LNCS*, 325–341, Springer, Berlin, Heidelberg, (2007).

[17] M. Quadrana, P. Cremonesi, and D. Jannach, 'Sequence-Aware Recommender Systems', *ACM Computing Surveys*, (66), (2018).

[18] H. Roediger and J. Karpicke, 'Test-enhanced learning: Taking memory tests improves long-term retention', *Psychological Science*, **17**.

[19] N. Tintarev and J. Masthoff, 'Designing and evaluating explanations for recommender systems', in *Recommender Systems Handbook*, eds., F. Ricci, L. Rokach, B. Shapira, and P. Kantor, 479–510, Springer, Boston, MA, USA, (2011).

[20] M. Wang, P. Ren, L. Mei, Z. Chen, J. Ma, and M. deRijke, 'A Collaborative Session-based Recommendation Approach with Parallel Memory Modules ', in *42nd ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 345–354, Paris, France, (2019).