iTalk2Learn
2013-10-31

Deliverable 2.1

Report on state-of-the-art of and requirements for machine learning methods for intelligent tutoring systems

31 October 2013

Project acronym: iTalk2Learn

Project full title: Talk, Tutor, Explore, Learn: Intelligent Tutoring and Exploration for Robust Learning

Work Package: 2

Document title: D2.1-report_on_state-of-the-art_and_requirements

Version: **1.0**

Official delivery date: 31.10.2013

Actual publication date:

Type of document: Report

Nature: Restricted to a group specified by the consortium

Main Authors: Beate Grawemeyer (BBK) and Carlotta Schatten (UHi) with contributions from all partners

Version	Date	Sections Affected
0.1	04/09/2013	First draft
		Approaches for tutoring support in Intelligent Tutoring Systems (UHi)
		Approaches in exploratory learning (BBK)
		Requirements on target tasks to support (BBK + UHi)
		Requirements on transaction for data collection (BBK)
		Requirements on background data collection (BBK + UHi)
0.2	27/09/2013	Input received from General Meeting
0.3	10/10/2013	Whizz Internal Review
0.4	20/10/2013	RUB Internal Review
0.5	30/10/2013	Submitted Version

Executive Summary

This deliverable reports on the state-of-the-art in machine learning methods for intelligent tutoring systems, and discusses requirements for these systems. The state-of-the-art section is divided into two parts. The first part describes approaches in machine learning for tutoring support in Intelligent Tutoring Systems. The second part describes approaches to intelligent support in exploratory learning environments. Following this, the requirements for adaptive intelligence in robust learning support are outlined. This last section covers the requirements relating to (1) target tasks to support; (2) transaction on data collection; and (3) background data collection.

Table of Contents

List of Figures

List of Abbreviations

MF	Matrix Factorisation
RL	Reinforcement Learning
ITS	Intelligent Tutoring System
ELE	Exploratory Learning Environment
BKT	Bayesian Knowledge Tracing
BN	Bayesian Network
LFM	Learning Factors Analysis
PFM	Performance Factors Analysis
IFM	Instructional Factors Analysis
MDP	Markov Decision Process

1 Introduction

The iTalk2Learn project is dedicated to developing adaptive intelligence for robust learning. This deliverable is part of work package 2. The aim of this work package is to provide data and background knowledge driven, efficient intelligence for different aspects of robust learning systems. In particular, improved predictive models and more efficient inference algorithms that more appropriately select:

- problems, (T2.2)
- interventions, (T2.2)
- support in exploratory learning, (T2.3)
- between the trade-off of exploratory learning vs. structured learning (T2.4)
- by incorporating task, learner and context characteristics, e.g., relevant input skills or mood of the learner. (All tasks)

Finally, the last objective is to

- orchestrate the different intelligence components for adequate use of triggers such as praise and constructive comments. (All tasks)

Structured as well as exploratory / conceptually-oriented learning is supported in the iTalk2Learn platform through the integration of intelligent tutoring systems, Fractions Tutor and Whizz, as well as an exploratory learning environment (ELE). In order to facilitate adaptive intelligence, a recommender system will be developed, capable of sequencing structured tasks and switching between structured and exploratory / conceptually-oriented learning tasks according to individual needs. Additionally, task-dependent support for the ELE will be developed. Also, task-independent support will be provided. This task-independent support acts across the different learning tasks. Figure 1 shows the main components of iTalk2Learn. It highlights the components that this deliverable focuses on: the recommender; the task-independent support; and task-dependent support for the exploratory learning environment.

The recommender communicates with the iTalk2Learn platform and receives data from the different learning systems (ELE, Whizz, and Fractions Tutor) as well as the speech recognition software. The recommender system will use this data to decide which type of exercise (structured or exploratory / conceptually-oriented) the student will be confronted with next.

The task-independent support also communicates with the iTalk2Learn platform and receives data from the speech recognition software. Support will be provided based on the student's speech. This component might also send data back to the different learning systems (ELE, Whizz, Fractions Tutor) in accordance with the support provided.

The task-dependent support communicates directly with the ELE as well as the speech recognition software. The data received will be used to provide adaptive support to the student while interacting with the ELE.

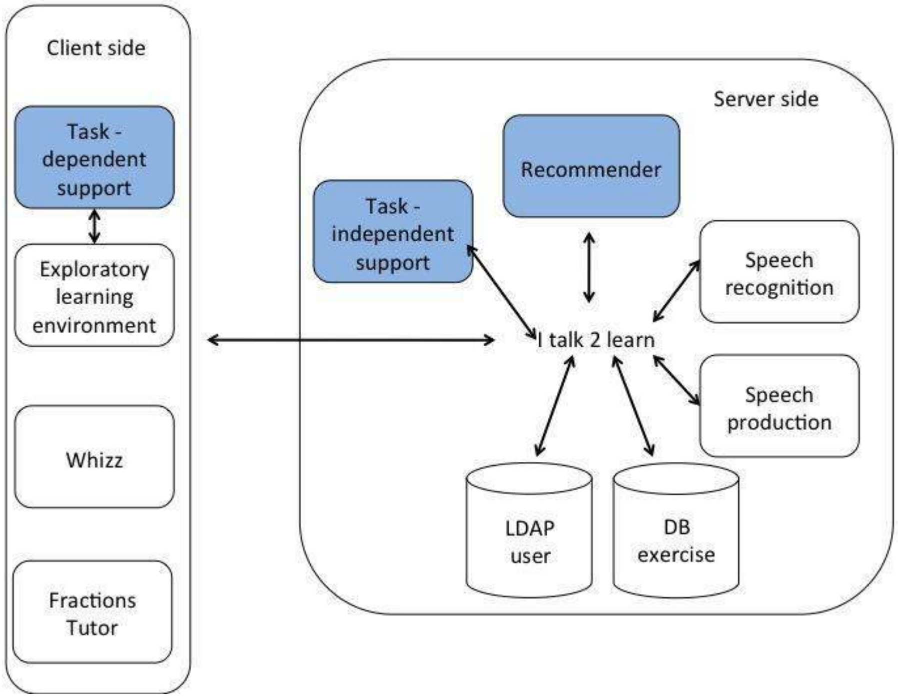

Fig. 1: Main platform and components of iTalk2Learn

As this is the first deliverable of WP2, we describe the state-of-the-art of intelligent components in ITS, as well as intelligent support for exploratory learning environments, which are based on machine learning methods. We also report on the requirements for the development of the three components including the data available for modelling.

2 State of the art of intelligent support

In this section we collect machine learning based state-of-the-art methods for structured and exploratory activities. Given the tasks of the project, we will focus on the methods that work at student level and are related to the task of the project.

'Machine learning algorithms' refer to algorithms that can learn from the available data. They are able to distinguish between different situations, e.g. distinguish between picture of different objects, and predict an outcome, e.g. price prediction, score prediction fuel consumption prediction. In order to do so, a data set is needed that is big enough for the algorithms to be able to abstract from the single examples. This approach is common to all machine learning experiments independently from the specific algorithm utilized and from the task.

In the specific case of the project, we discuss student performance prediction (T2.2), intervention sequencing (T2.3) and planned learning (T2.4). These are specific machine learning applications, although the underlying mathematical formulation allows the same algorithm to be used for several completely different tasks.

Different experimental settings are used according to the mentioned tasks, but some aspects are equal. Given a data set, composed by a combination of input characteristics (features) and connected outcomes, a part of it is used to build (train) the model. On hand of this model we are able to predict the outcome for unforeseen input characteristics. How accurate is the prediction is computed by means of the second part of the data set. The procedure consists in simulating a real word usage, where the input features are given and the outcome is unknown. The model predicts the outcome for the given features and its results is compared with the real outcome recorded in the data set. The smaller is the difference between predicted and recorded outcome, the more it is assumed that the system will perform correctly in a real scenario.

A slightly different approach is adopted for the sequencing tasks. The algorithm needs to retrieve a policy in light of the possible interactions with the system and the subsequent state of the user. This kind of problems is more difficult than the previous ones, since it cannot be evaluated in a laboratory with the simulated real world application of the model. The reason is that we are creating a model able to suggest a sequence and given the past outcomes. Consequently, we are evaluating an adaptive and highly individualized set of sequenced actions, whose effect is distributed over time. Given our specific task of sequencing structured and exploratory tasks (T2.2 and T2.4), the optimality of a sequence can be measured only after a student has interacted with the proposed tasks. The amount of learning can be estimated in comparison with the knowledge acquired with another reference sequencer, generally random or with other state of the art ones. Anyway, the test of the sequencer needs to be done on the field, whereas the model can be trained by evaluating a collected data set. As a matter of fact, machine learning methods targeting at retrieving the correct sequence of actions, were developed for robots or artificial agents. Therefore, issues regarding experimentation on human test subjects were not considered. For instance, with robots there is the possibility of recording different sequences of actions, without considering the fatigue of the subject under test. This did not put a constraint on number of data required for creating a model. Measuring the success of the sequence means measure the knowledge of the student before and after the test. Given the small amount of time for testing, it is difficult to measure quantitatively the difference in amount of learning, where the only difference is the sequence presented.
Also the data set has to possess particular characteristics. In a prediction task it was not important if the data collected was sequenced with a specific policy. Here, instead, it is needed to evaluate a good percentage of the possible combination of actions. Otherwise the algorithm has not enough information to retrieve an optimal policy. In order to collect this special kind of data sets, called exploratory corpus, the tasks are sequenced randomly. This makes an ethical question arise about the rights we have in suggesting difficult contents to novices or easy contents to experts.

2.1 Approaches for tutoring support in Intelligent Tutoring Systems

Machine learning methods that try to adapt the tutoring system by learning from past behaviour will be investigated, along with how they extend to given background knowledge. We will start with performance prediction algorithms, which are used not only to predict the score of a student, but also to build an internal representation of his or her state. These tasks are connected to the next section where we describe how state-of-the-art methods sequence tasks in order to maximise students' learning.

Finally, we will conclude with adaptive hint management and an analysis on speech indicators, as an extension of the previously mentioned tasks.

2.1.1 Performance prediction algorithms

Different algorithms have been applied to model the knowledge acquisition process with the objective of performance prediction. These algorithms are of primary importance for T2.2 where recommender system models for problem and intervention selection are under study. Recommender systems are performance prediction methods, whose prediction is used to recommend and then accordingly sequence the tasks.

In order to do so a data set containing the performances over time of the students in the different tasks is required. Public ones are available like ASSISTments (Feng et al. 2009) and Bridge and Algebra (http://pslcdatashop.web.cmu.edu/KDDCup/rules_data_format.jsp). They are composed by log files of ITS, i.e. recordings of the score of a student in the tasks he or she accomplished, number of hints and domain information. Example of domain information are the domain of appartenance (fractions, additions, etc.), number of skills required to solve the exercise and other information necessary in order to individuate an univoque step, if we are talking of a multiple step tasks.

The most widely used algorithm for performance prediction is **Bayesian Knowledge Tracing** (BKT), which was introduced by Corbett and Anderson in 1995 and extended and refined in subsequent years. The first implementation consisted of a simple Hidden Markov Model where the performance prediction of all students was modelled by four variables, two representing the performance (probability of learning and probability of forgetting) and two representing the knowledge on a single skill of a student population (probability of guessing and of slipping). In this particular model the knowledge variables considered are called latent features because they are never observed directly. Moreover, knowledge and performance are represented as binary features (i.e., it is assumed that just two states are possible): for a skill learned/not learned and for an answer correct/wrong. Another important variable is the prior probability representing the prior knowledge of the student at the moment he or she starts to use the system.

BKT algorithm can be resumed in two steps, which we previously called ‘training’ and ‘testing’. In the training phase, the four previously mentioned variables are estimated using a data set. Then the model evolves during the testing phase by increasing or decreasing the probability of learning of a skill according to the student’s answers. This is done until the skill can be considered as learned. A skill is considered as learned if the probability of a correct answer is greater than 0.95.

Since this representation is quite limiting, the model was extended. In the various extensions proposed the researchers have focused on different aspects, such as multiple-skill modelling, personalisation, time, and partial credit.

Multiple skill modelling has been one of the most important targets of BKT researchers. Creating a classic BKT model for each skill cannot infer properly on the score of multiple step exercises and consequently limit the use of BKT to simple structured exercises[1]. Xu and Mostow (2012) make a comparison between different approaches based upon both joint and not joint computation of the multiple skills. Also, the work of Cen et al. (2006) and Gong et al. (2010) is based on a single skill which is entirely responsible for the outcome of the considered step. Koedinger et al. (2011), instead, suggest the first combination of skill knowledge to predict student performances. This method is called Conjunctive Knowledge Tracing (CKT), which is used as comparison for the methods that learn how the possible known skills combine themselves in a given step. Nevertheless, the student's knowledge and performance is still modelled as a binary variable. Another more recent approach is proposed by Xu and Mostow (2013), where Item Response Theory is applied (instead of Logistic Regression as done in the previous approaches) to refine knowledge tracing. The work done on Item Response Theory is in its early stages. In the paper proposed there are not clearly outperforming results, consequently, we consider the aforementioned state-of-the-art methods as equally relevant.

Regarding personalisation, there are two papers that are of particular interest for iTalk2Learn. The first, by Pardos and Heffernan (2010), proposes a multiple prior knowledge parameter. The algorithm will decide to which level a student appertains. As a consequence, different students' levels are defined. The second, Lee and Brunskill (2012), points out the necessity of modelling all the variables differently for the students because the probability of knowledge is not equal for each student at a specific time step. The authors suggest creating a model for each student. This information is then used to compute the necessary time each student requires to practice (i.e. the number of exercises to be solved).

Time is a more recent introduction of BKT and shows that training a model with data that is too old to train has a negative influence on the model accuracy. This happens because, from one learning session to the others, the student's behaviour and knowledge can change (Nooraei et al., 2012).

Partial Credit was also introduced by Wang and Heffernan (2011). A simple equation is developed to create from a binary performance (the score can be either 0 or 1 in public data sets) a continuous one (a score defined between 0 and 1). This strategy proved to be effective for ameliorating the accuracy of BKT.

In the following Figure we report the formulas of two Bayesian implementation, the so called standard BKT (Corbett et al., 1994) and CKT (Koedinger et al., 2011). These formulas show how the probability of knowledge, guessing, and slipping are used to predict the probability of the student giving the correct score. Moreover, they also show how the model changes over time, i.e. how the probability of knowledge is updated. In particular, Fig. 3 reports that, as aforementioned, CKT consider the probabilities of all involved skills for computing the performance and knowledge probability.

[1] The algorithms distinguish between single step and multiple step tasks. Single step exercises like 1+2=?, requires just the knowledge of a skill. Multiple steps one, instead, like find the value of x for 6x+3=5, requires more than a skill. The probability of solving a task is dependent of each skill involve consequently the probability of knowledge of each skill needs to be modelled jointly.

Standard BKT Prediction

$$P\left(y^t=1\right)=P\left(k_j^t\right)\left(1-P\left(S\right)_j\right)+\left(1-P\left(K\right)_j^t P\left(G_j\right)\right) \tag{1}$$

Standard BKT skill update for successful step

$$P_{\text{posterior}}\left(k_j^t\right)=P\left(k_j^t \mid y^t=1\right)=\frac{P\left(k_j^t\right)\left(1-P\left(S\right)_j\right)}{P\left(k_j^t\right)\left(1-P\left(S\right)_j\right)+\left(1-P\left(K\right)_j^t P\left(G_j\right)\right)} \tag{2}$$

Standard BKT skill update for failed step

$$P_{\text{posterior}}\left(k_j^t\right)=P\left(k_j^t \mid y^t=0\right)=\frac{P\left(k_j^t\right)P\left(S\right)_j}{P\left(k_j^t\right)P\left(S\right)_j+\left(1-P\left(K\right)_j^t\left(1-P\left(G_j\right)\right)\right)} \tag{3}$$

Standard KT next–step update

$$\begin{aligned}P_{\text{posterior}}\left(k_j^{t+1}\right)=P_{\text{posterior}}\left(k_j^t\right)+ \quad (4)\\ +P_{\text{posterior}}\left(k_j^t\right)\left(1-P(F)\right)+\left(1-P_{\text{posterior}}\left(k_j^t\right)\right)\left(1-P\left(G_j\right)\right)\end{aligned}$$

Fig. 2: Standard Bayesian Knowledge Tracing Formulas for Prediction, Skill Update and Next Step Update

CKT Performance Prediction

$$P\left(y^t=1\right)=\prod_j P\left(k_j^t\right)\left(1-P\left(S\right)_j\right)+\left(1-P\left(K\right)_j^t P\left(G_j\right)\right) \tag{5}$$

CKT Skill Update for failed step

$$\begin{aligned}P_{\text{posterior}}\left(k_j^t \mid y^t=0\right)= \quad (6)\\ =\frac{\left(P(S)_j\left(1-P\left(S\right)_j\right)\prod_{i\neq j}\left(P\left(k_i^t\right)\left(1-P\left(S\right)_i\right)+\left(1-P\left(k_i^t\right)\right)P\left(G\right)_i\right)P\left(k_j^t\right)}{1-\prod_j P\left(k_j^t\right)\left(1-P\left(S\right)_j\right)+\left(1-P\left(K\right)_j^t P\left(G_j\right)\right)}\end{aligned}$$

t	time step
j	skill j
y^t	Performance at time step t
k^t	knowledge at time t, hidden state
$P(K)$	learning probability
$P(G)$	guessing probability
$P(S)$	slipping probability probability

Fig. 3: Modified formulas for CKT.

Recommender systems for ITSs are generally used for user performance prediction. Several methods coming from item recommendation tasks were adapted for performance prediction. Subsequently, we mention the most important ones.

Matrix Factorisation (MF) was developed at first as item recommender system technique and became recently the most popular state of the art algorithm. Given a data set of user rates on items, one wants to know which item will be positively or negatively rated in the future. The final goal is to inform potential clients of items they might be interested in purchasing. In Thai-Nghe et al. (2010), the parallelism between the relation student-item-rate and student-exercise-score becomes clear. Thai-Nghe et al. (2010) show how it is possible to apply the same sort of algorithms to recommendation and performance prediction. MF consists in the approximation of an incomplete matrix (i.e. the table student-task-score Fig. 4), by decomposing it in two different ones: the elements of the two matrices are called latent features and have no physical meaning. Using the available cells (i.e. the scores that were recorded from previously performed tasks), we can compute the missing ones by means of very fast optimisation algorithms. Consequently, MF can predict the score of a specific student on a specific task. The developer does not need to insert information on the specific domain because the single tasks are considered separately and the features are modelling the various domain information (Thai-Nghe et al., 2010).

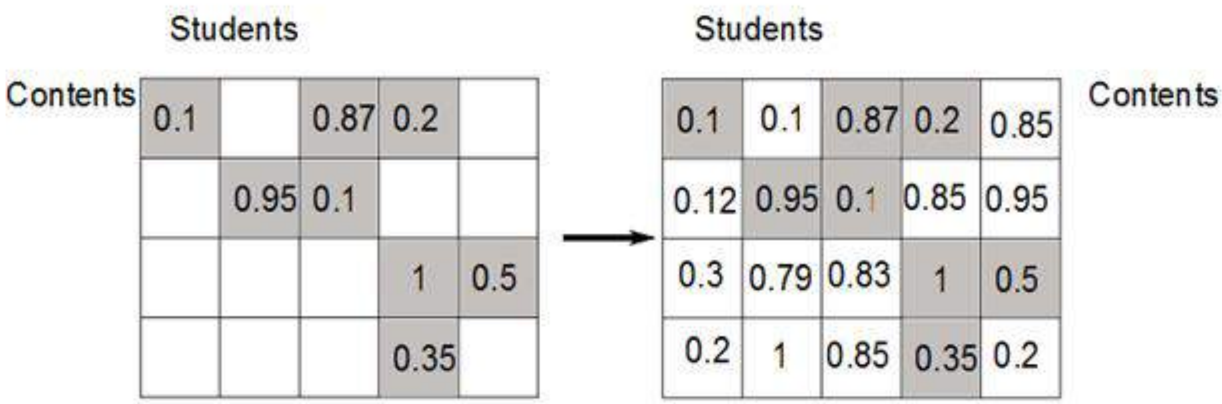

Fig.4 : Computing of the missing values by means of MF. Students received a score on specific tasks, represented by the gray squares, and MF predicts the missing scores they could receive in unforeseen ones. The task, from a machine learning perspective, is similar to the recommendation problem. A user rates specific items and this information is used to predict the ratings on other items.

MF is criticized for not considering the impact of time in its updates (i.e., records of the student in the past have the same importance as scores obtained recently). Moreover, it is impossible to predict the score of a student that was never recorded or an exercise that was never solved. Solutions to this problem have been proposed. For example, in Gantner et al. (2010), the temporal context was explicitly modelled. Krohn-Grimberghe et al. (2011) proposed a method to ameliorate the performance prediction by selecting only a subset of the new available data to build the model. Thai-Nghe et al. (2011, 2012), instead, proposed the Tensor Factorisation Forecaster (TFF). The matrix, considered in Fig. 3, becomes a tensor because the time dimension is added. The function used to compute the next score of a student within content – the prediction function – can be seen in Fig. 4 Eq. 1. Only the most recent data recordings of the students are considered and the single elements of the learned tensor are weighted in different ways. The following figure reports the most successful method.

$$p_{i,u,T^*} = \mu + b_u + b_i + \sum_{k=1}^{K} w_{uk} h_{ik} \Phi_{T^*k} \qquad (7)$$

Where:

i	represents an item or task
u	represents an user or student
k	represent the latent feature
K	Total number of latent features
μ	$= \frac{\sum_{p \in D^{train}} p}{\lvert D^{train} \rvert}$, with D^{train} as sequence of performances in the train sample. It is the average performance
b_u	$\frac{\sum_{p^u \in D^{train}} p^u - \mu}{\lvert p^u \in D^{train} \rvert}$ user bias
b_i	$\frac{\sum_{p \in D^{train}} p^i - \mu}{\lvert p^i \in D^{train} \rvert}$ item bias
Φ_{T^*k}	$\frac{\sum_{t=T^*-L}^{T^*-1} h'_{tk} q_{tk} p_t}{L}$

with:

T^*	Time when the step is performed
h'_{tk}	Latent factor of the previous solved task
q_{tk}	Time latent factor
p_t	Previous performance of the student
L	Number of time instant considered

Fig. 5: Equations computing the predicted score for the Tensor Factorization Forecaster

Other recommendation related algorithms include Multi-Relational Matrix Factorisation, where more relations are taken into consideration. Instead of considering a single incomplete table (students-contents), other tables are added such as tasks and skills required or students and skill known (Thai-Nghe et al., 2011). This is done in order to include more information in the model and ameliorate the prediction accuracy. Another approach is the Factorisation Machine (Rendle et al. 2010), which is used to predict the student performance (Thai-Nghe et al. 2012). This algorithm is a combination of Matrix Factorisation and Support Vector Machines, i.e. a machine learning method which finds the separating hyperplane, which maximises the margin or the distance between two data points of different classes (Vapnik 1998).

Another alternative to the aforementioned algorithms for performance prediction is **Performance Factors Analysis** (PFM) based algorithms. As pointed out by Gong et al. (2010), Performance Factors Analysis is an alternative to BKT for predicting student performance. The first related method is Learning Factors Analysis (LFM) (Cen et al. 2006) where subject ability on a skill, easiness of a skill and the learning rate for each skill are modelled.

$$\textbf{AFM:}\quad \ln \frac{p_{ij}}{1-p_{ij}} = \theta_i + \sum_k \beta_k Q_{kj} + \sum_k Q_{kj}\left(\gamma_k N_{ik}\right) \quad (8)$$

$$\textbf{PFM:}\quad \ln \frac{p_{ij}}{1-p_{ij}} = \theta_i + \sum_k \beta_k Q_{kj} + \sum_k Q_{kj}\left(\mu_k S_{ik} + \rho_k F_{ik}\right) \quad (9)$$

$$\textbf{IFM:}\quad \ln \frac{p_{ij}}{1-p_{ij}} = \theta_i + \sum_k \beta_k Q_{kj} + \sum_k Q_{kj}\left(\mu_k S_{ik} + \rho_k F_{ik} + \nu_k T_{ik}\right) \quad (10)$$

Where:

i	Represents a student
j	Represents a step
k	Represents a skill or a KC
p_{ij}	is the probability that student i would be correct on step j
θ_i	is the coefficient representing the proficiency of student i
β_k	is the coefficient representing the difficulty of the skill or KC k
Q_{kj}	is the matrix connecting steps and skills
γ_k	is the learning rate for KC or skill k
N_{ik}	is the number of times a student has practiced on a skill
μ_k	Coefficient representing the benefit of privious successes
S_{ik}	Number of previous successes of student i on skill k
ρ_k	Coefficient representing the benefit of previous failure on skill k
F_{ik}	Number of previous failures of student i on skill k
ν_k	Coefficient representing the benefit of previous tells on skill k
T_{ik}	Number of previous tells of student i on skill k

Fig 6: Probability for correct response in AFM, PFM and IDM.

LFM has been earlier applied to multiple-skills (Lszczenski et al. 2007, Cen et al. 2008). Although the model considers frequency, the outcome of the exercises performed is not considered. As a consequence, PFM is suggested (Pavlik et al. 2009). Nevertheless the two algorithms are still considered equivalent since, comparing different error and accuracy measures, one does not clearly outperform the other (Chi et al. 2011). Chi et al. (2011) proposes a new Factors Analysis based algorithm in order to consider instructional interventions during the exercises. The algorithm used is called Instructional Factor

Analysis (IFM). In Fig. 6 one can find how the probability of a correct response is computed by the different algorithms.

2.1.2 Adaptive Sequencing of contents: Reinforcement Learning

As shown in the previous section, student modelling has been used to predict student performance, but few comments were formulated on the task of sequencing. If we consider the use cases of the project, the Whizz scheduler possess adaptivity. It intervenes in the sequence, changing the difficulty of the tasks, if the student is not able to solve the previous ones. The Fraction Tutor, instead, has a fixed sequence of tasks. For these reasons, for planned learning in task 2.4, we suggested Reinforcement Learning based algorithms and other Operational Research connected methods, which are state-of-the-art method for finding the best sequence, or optimal policy. In particular switching between structured and exploratory activities can be seen as a special case of sequencing.

Reinforcement Learning (RL) is generally used to find the best sequence of actions in a stochastic environment[2]. It consists of three elements: a state space (i.e. states in which a student can be), an action set (i.e. number of actions that can be performed), and a reward function (i.e. a function that evaluates how good a transition from one state to the other is). As already said goal is to find the best policy to sequence the given actions. This is done by the algorithm trying to maximize the reward that it receives each time an action is performed. Possible states and actions, as well as the reward for an action at a specific states, needs to be defined and change according to the own understanding of the problem. Examples how to do this can be found in the state of the art. For instance, given the task of sequencing structured activities, the state of the student could be represented by his or her knowledge of the skills involved. The possible actions are the tasks that can be proposed to the student next and the reward is a measure of how much the actions were successful. Generally the reward is computed by considering the time needed to solve the task by the student, the score obtained and the difficulty.

After having defined a state space, an action set and a reward function, the policy can be retrieved with different strategies. RL can be subdivided into two main approaches. The first one is the so-called **model-based** approach and can be seen as a Markov Decision Process (MDP). As said in the introduction, it is required to evaluate each probability of transition from one state to the other and by testing each possible action. Data set collected utilizing a heuristic, like showing the exercises in difficulty order, are not suitable since they exclude the possibility, for instance, to show more difficult tasks to novices. Consequently, the system cannot learn each transition probability from one state to the other. If we consider the previous suggested example, the system is not able to compute the probability of success in presenting a difficult task to a novice and an own strategy cannot be retrieved. The **model-free** approach, instead, takes decision while student is playing with it. These RL algorithms do not create a model from a given data set, but evaluate the situation each time they need to take a decision. They modify their behaviour based on the past and present outcomes. Without any initial knowledge of the

[2] With stochastic environment is meat a system that is not deterministic, i.e. the transition to one state to the other is governed by a probability. For instance, answering correctly to a task is not deterministic because there is a certain probability that the student has not understood it or slipped it.

system, at the beginning the policy could be random and damage the machine which is steering or, in our case, confuse the interacting person. Consequently, student simulators are needed to retrieve an initial approximated policy before experimenting on humans (Sutton & Barto, 1998).

In the following Figure, we show how the "quality" Q of the selected action is measured in model-based RL. In order to do so, two mathematical weights, learning rate and a discount factor, need to be selected experimentally. The learning rate defines how the new information is important in respect with the past experience. The discount factor, instead, gives more important to the present reward than the future one, i.e. it is more important to reach a middle quality state now than the optimal state in the future. The optimality of a state is defined with a reward by the developer. The more the state goes near the optimal state the more and the higher will be the reward.

$$Q : S \times A \to \mathbb{R} \tag{11}$$

$$Q_t\left(s_{t+1}, a_{t+1}\right) = \left(1 - \alpha_t\left(s_t, a_t\right)\right) Q_t\left(s_t, a_t\right) + \alpha_t\left(s_t, a_t\right)\left(R_{t+1} + \gamma \max_a Q_t\left(s_{t+1}, a_t\right)\right) \tag{12}$$

where

a	Action
s	State
A	Action space
S	State space
$Q_t\left(s_{t+1}, a_{t+1}\right)$	Quality of the state-action combination
$\left(1 - \alpha_t\left(s_t, a_t\right)\right)$	Inverse learning rate
$\alpha_t\left(s_t, a_t\right)$	Learning rate
γ	Discount factor
R_{t+1}	Reward
$\max_a Q_t\left(s_{t+1}, a_t\right)$	Estimate of optimal future quality value

Fig 7: Equations of Q-learning, a model-based RL algorithm.

RL is utilized as an adaptive/intelligent sequencing method in ITS. Beck et al. (2001), Sarma et al. (2007) and Malpani et al. (2011) sequence exercises[C1] , whereas Beck et al. (2001) only do so specifically for autistic students[C2] . Martin et al. (2004) use RL for hint management. Chi et al. (2010), instead, propose RL as decision method for eliciting or telling the solution. Iglesias et al. (2003) sequence instruction session and exercises sessions, deciding also in which format they will be displayed. Later in

2009, the same group of researchers extended the work in testing different RL algorithms. We used this state of the art to define the requirements of Section 3 for sequencing tasks.

The tasks accomplished there are various and located at different intervention levels. Nevertheless, the work on this topic is limited because of the intrinsic difficulties of the problem. One of these difficulties has already been mentioned – i.e. the difficulty to find an appropriate dataset in order to construct the optimal policy, or a coherent simulator for modelling students' behaviour. As explained in Chi et al. (2011), this task is far from trivial. Moreover, another less evident difficulty is the demonstration of the effectiveness of a system designed with adaptive sequencing: real students are necessary in a statistical significant amount in order to evaluate the difference in learning and the creation of the models. The papers mentioned demonstrates a partially successful outcome, generally focusing on the time needed to learn the different content more than on the total amount of learning.

Some attempt of sequencing is also discussed in BKT; generally in ITS (exploiting this technique for user modelling), the next exercise is selected to try to maximise the most unlearned skill. Koedinger et al. (2011) explain how this method is not suitable for exercises that involve more than one skill. The old learning strategy would select the exercises with more not mastered skills and consequently display a preference for the harder ones. Intuitively, this would result in a steep learning slope. Instead, Koedinger and colleagues proposed a solution which displays a preference for problems where a few not mastered skills are present. As already pointed out by Koedinger and colleagues, we would like to stress the problem of modelling tasks with multiple skills. Until now, scaffolding has been used to cope with the single skill modelling limitation of BKT. As a result, multiple skills exercises are represented as single ones reducing the selection modelling ability of the system.

2.1.3 Speech in ITS

Speech has been used for different purposes in ITS and a detailed description was already given in D3.1 and D3.2. In this brief section, we would like to focus on the state-of-the-art speech based methods utilised to ameliorate ITS. Although the task is related to WP 3 T3.4, it is strongly connected both to T2.2 and T2.3 as their possible extension. Consequently, we will report in Section 3.1 its requirements and present here the (little) work on this topic.

In the following we resume the consideration done in D3.1 and D3.2 about the speech roles in ITS. Joshi and Kaur (2013) report on different algorithms that were used for the task of speech recognition. Generally, the algorithms applied for solving this task are the ones used also for phoneme recognition, e.g. Hidden Markov Models, Gaussian Mixture Models, Artificial Neural Networks, k-Nearest Neighbors, Maximum Likelihood Bayesian Classifiers and Support Vector Machine (Mao et al. 2009, Chavan et al. 2010). Another important role played by speech in ITS is related to emotion recognition, mostly with facial expressions and voice modulation from a synthetic teacher (Graesser 2005)[3]. Facial expressions and gestures from the interacting students have also been analysed (Whitehill et al. 2008). The purpose

[3] See D3.1 and D3.2 for more details.

of this is mainly to create self-confidence and to motivate the students with encouraging utterances by the teacher and correct interventions. Emotions are also recognised in the voice, but, to the best of our knowledge, they are not used for performance prediction, as it is planned for T2.3 and T3.4. In the appendix, we report a list of previous work on emotion recognition that we could analyse to extend the algorithms.

The work by Worsley and Blikstein (2011) is particularly relevant for the project, since it extracts and analyses indicators from speech. The final goal of the researchers was to check whether, from the extracted features, it is possible to understand if a student has a novice, intermediate or expert knowledge in a topic. No machine learning algorithms have been applied, but the statistical results let us assume that it is possible to retrieve proficiency from speech.

In the project we will combine the mentioned work with the Automatic Speech Recognition System of Sail. At the same time, we will have a speech recognition system and features indicating the performance of the students. The scope of the project is not to recognise the emotion in a direct way, but to exploit the information to ameliorate the already implemented intelligent modules

2.1.4 Hints management in ITS

Another important task in ITS is the automatic managements of hints, i.e. the design of intelligent systems is able to give the adequate and individualized help to student at an optimal time. This topic is relevant for T2.2 where intervention strategies were proposed to be implemented with adaptive methods.

Mavrikis (2010) utilises Bayesian Networks (BN) to model if the student is learning from the ITS and if the student can answer a question without help request. BN are acyclic graphical models used to represent probabilistic dependencies among random variables. These are represented as nodes, whereas the conditional dependencies are modelled as directed arcs. A node without arcs pointing at it is called a parent node and the others, called conditional nodes, are connected with a conditional probability table (CPT). The latter quantifies the effect that the parent nodes have on their children nodes. The graphical model allows to derive the probability of each random variable involved and also to define the joint probability density function. Nodes not connected to each other are independent from each other and consequently the joint probability function can be simplified. In conclusion, BN can be mainly used for representing or retrieving the structure of a process and determine the probability of the variables involved. If the variables are also time-dependent, the BN is called Dynamic Bayesian Networks and the probabilities change over time. BN can simply be used to model probabilistic environments, in which dependencies and prior probabilities are already known. The approach is different if the only given information is a set of data. The structure of the BN needs then to be learned with an optimisation algorithm that performs a feature selection in parallel – i.e. it tells which features are more important than others to determine whether or not a student requires help. This work is reported in Mavrikis (2010) and Lallé et al. (2013) and will be utilised to decide new features for iTalk2Learn data collection.

Other machine learning approaches have been investigated for the task of hint management or sequencing. Reinforcement Learning algorithms have been proposed to sequence hints and other types of action for students in difficulty (Chi et al. 2011, Martin et al. 2004). In Chi (2010) useful features were also analysed.

Hint management is particularly effective against the students that try to 'game' the system because it avoids that the student is able to type senseless answers or ask for help in order to receive the bottom-out hint[4]. Diziol et al. (2009) and Baker, Walonosy et al. (2008) report how students game the system: they try to access the bottom-out hint, before even trying to think of the solution. In the same article it becomes also clear, that in multiple step exercises hints can induce a correct problem-solving attitude.
In this section we discussed the use of adaptive hints in structured activities. The same problem can be transferred from structured activities to exploratory ones, as shown in Section 2.2.3.

2.2 Approaches in exploratory learning

In this section we report on the state of the art required for T2.3, intelligent support for Exploratory Learning Environments (ELEs). ELEs are able to support students to discover and understand underlying domain concepts, rather than supporting drill and practice activities to reinforce procedures as typically applied in ITS (see D1.1 and D3.2 for more details).

As described in Trudel and Payne (1995) and Mavrikis et al. (in press), the learning performance in an ELE depends on the learner's ability to formulate goals as well as their ability to reflect on the effectiveness of the means of achieving these goals, including planning and carrying out tasks. Additionally, the affective state of the learner is an important factor for learning performance (Porayska-Pomsta et al., 2013).
As described in D1.1, conceptual knowledge can be developed through students engaging with exploratory tasks related to domain-specific content. This exploration entails self-regulated processes. Self-regulated learning can be described as an active process of controlling and evaluating one's own learning. It includes, for example, the setting of goals as well as the monitoring and control of the learning behaviour directed and constrained by those goals (Pintrich, 2000). Different forms of support have been devised to assist the learner at different reasoning stages during exploration in a learning platform. Devolder et al. (2013) provide an overview of supporting mechanisms for learning phases in self-regulated learning. The support provided for self-regulated learning partly overlaps with support developed in ELEs. While support in self-regulated learning focusses mainly on goal setting and reflection, in ELEs support is additionally provided in the exploration phase during task execution.

In order to develop students' conceptual knowledge, in relation to ELEs in particular we are considering the different support mechanisms when students are undertaking exploratory tasks against Pólya's framework for solving mathematical problems (Pólya, 1945). We have chosen Pólya's reasoning stages as a way to structure the discussion below because they resemble the recursive and iterative problem solving processes involved in exploratory tasks and reflect the strategies required for developing

[4] See explanation in D1.1. Bottom-out hints are those hints telling, or almost telling, the exact solution to the student.

conceptual knowledge for robust mathematical knowledge (Schoenfeld, 1992). This allows us to organise a discussion of the state of the art in terms of stages, as follows:

1. **Understanding the problem and formulation of goals**
   Based on the understanding of the problem, goals have to be formulated in order to manage the exploration of the ELE. The formulation of goals determines the focus of the exploration.
2. **Devising a plan that includes certain tasks in order to achieve goal(s)**
   Once goals have been formulated, strategies for achieving them need to be devised, such as how to explore the ELE including projected sequences of actions.
3. **Carrying out the plan/ tasks**
   This stage involves the execution of the plan or strategy to achieve the goals. It refers to a goal-driven exploration of the ELE.
4. **Reflecting on the plan and outcome**
   This stage involves reflecting on the effectiveness of the exploration, including whether the plan of action to achieve the goal worked well, or a new plan is needed. It also includes reflection on new knowledge that has been learned through the exploration. This might lead to new goal formulation due to the additional knowledge gathered concerning the problem domain.

The following sections describe the state-of-the-art in intelligent support in respect of the different reasoning stages described above.

2.2.1 Formulating of goals

As described above, the formulation of goals determines the focus of the exploration within an ELE. This section describes methods through which this stage can be supported.

Sabourin et al. (2013) compare machine learning techniques to classify students into self-regulated learning categories in an educational game. The classification was based on their goal setting, as well as monitoring of their behaviour via text-based responses to update their status. While a naïve Bayesian model was the best predictor at an initial state for interacting with the environment, after the model was trained with the data, Decision Trees showed best performance for classifying students into the self-regulated learning categories.

Mavrikis et al. (in press) describe how goal setting is supported in their exploratory learning environment where the learner is presented with a set of goals. The system provides feedback on completion of a goal or suggestions of next steps towards the goal. The adaptive support is provided through declarative rules.

Non-adaptive support is provided by Moos and Azevedo (2008), who pre-formulate textual guiding questions at the beginning of a learning activity within a hypermedia learning environment to teach the circulatory system to increase learning performance. Similarly, Manlove et al. (2009) describe a system for enhancing physics skills that presents a goal hierarchy and goal description to enhance learning outcomes. Another example is the system described by Colancies & Nussbaum (2008) where WebCT is used for online discussions to promote reasoning skills. The system includes a goal instruction mechanism, where statements are given at the end of a discussion, which indicate what students should achieve.

The mechanisms discussed above are able to assist the learner in their goal formulation and guide the learner towards formulating a plan to achieve those goals/tasks.

2.2.2 Formulating a plan

This section looks at supporting the learner in planning or defining a strategy for exploring the ELE, including projected sequences of actions.

During this phase, only non-adaptive support has been provided. Davis & Linn (2000) describe a system that provides pre-formulated non-domain specific prompts for planning what tasks need to be performed to achieve a goal. Another example is Simons & Klein (2007), who present guided pre-formulated questions to formulate a plan of action towards a goal. Additionally, Gurlitt & Renkl (2008) presents learners at the beginning of the exploration with a concept-mapping task, which could be seen as a prior knowledge activation task that encourages planning of action.

2.2.3 Carrying out plan/task

This stage involves provision of supporting mechanisms during exploration, when a plan or strategy for achieving the goals is executed.

Different machine learning techniques were applied in order to model the learners' exploration behaviours. Similarly to ITSs, some form of probabilistic reasoning was applied. For example, Bunt & Conati (2003) describe an adaptive coach that provides hints and warnings to support feedback on learner's exploratory process in carrying out tasks. It uses a Bayesian Network to model students' interaction with the system.

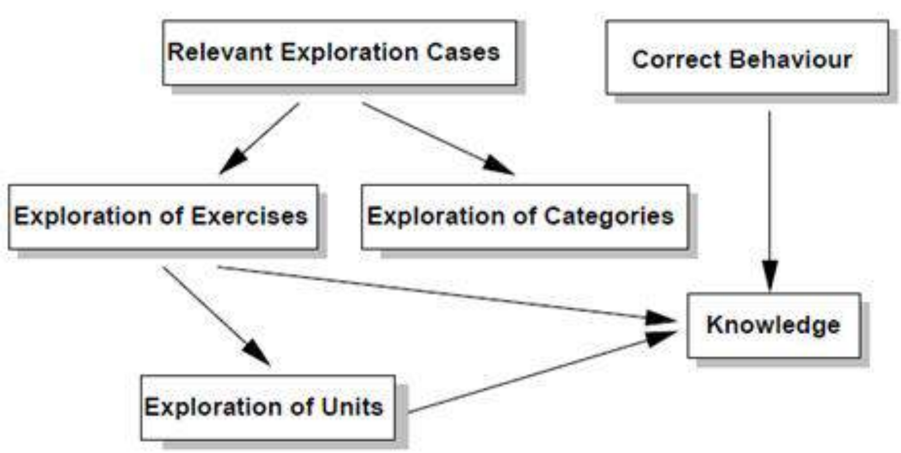

Fig. 8: An example of Bayesian Network used in exploratory learning (Bunt and Conati 2003).

Figure 8 shows an example of a Bayesian Network (Bunt & Conati, 2003). It includes two types of nodes: exploration nodes that represent the effectiveness of a learner's exploration behaviour; and knowledge nodes that represent the learner's understanding of the domain concept. This Bayesian Network is able to predict a learner's knowledge of the concept based on their exploration behaviour. Through the network, the system is able to provide hints, warnings and suggestions.

Similarly, Conati et al. (2013) describe an educational game that provides adaptive textual hints based on a probabilistic student model. The model is based on students' game actions. Hints are provided at incremental levels of details, according to the extent to which the model predicts that the student does not have certain types of knowledge. Additionally, Conati and Zhou (2002) describe a system that models the learner's emotional state with a Dynamic Decision Network in order to decide when and how to provide help during the exploration phase. Kickmeier-Rust and Albert (2010) explain how the learner can be supported in their competencies within an educational game environment. They use probabilistic reasoning and Knowledge Space Theory for intelligent support of the learner, based on their interaction with the environment. Lintean et al. (2011) look at how best to provide adaptive textual dialogue during exploration. Different machine learning techniques were compared. The word weighting method combined with Bayesian Nets provided best accuracy.

Other techniques include rule-based systems. Kim et al. (2009) describe an educational game that supports negotiation skills while performing different tasks within the game. A rule-base system is used to provide responses to learners' actions. Gutierrez-Santos et al. (2012) describe a microworld eXpresser that is able to provide different feedback strategies, based on the student's interaction with the learning environment. The system includes a reasoning layer that includes a knowledge base of rules. Based on the student's interaction with the system, a rule is selected and the associated feedback is given.

Kardan & Conati (2013) outline a system that is able to support the learner while exploring the learning environment. An unsupervised clustering algorithm is used to provide textual hints and interface changes. Those adaptive interventions are based on class association rules, based on discovered patterns.

A different strategy has been applied by Cocea et al. (2010), who apply case-based reasoning to model learners' actions in an exploratory learning environment. The system transforms the learners' actions into a sequence of simple cases (strategies) and compares it to all the strategies in the case base for a particular task. The most similar case is then retrieved and the solution is used to provide scaffolding for the learner in performing the task.

Webber et al. (2002) applied a different technique. Here a divide-and-conquer approach has been used to model the learners' exploratory behaviours. A group of small specialised agents are used to examine the ability of the learner to perform a particular task. The agents use a voting mechanism to aggregate different views. The behaviour observed at micro-level can then be interpreted at macro-level.

2.2.4 Reflecting on performance

As described above, this stage involves reflection on the effectiveness of the exploration. Different approaches have been developed to support reflection on planning or learning outcomes.

Non-adaptive self-reflection questions and prompts are commonly used in exploratory learning environments (e.g. Ergazaki et al., 2007; Crippen & Earl, 2007; Chang, 2007; Fund, 2007; Furberg, 2009; Lindstoem et al., 2011; Nash et al., 2011; van der Meij & de Jong, 2011; Jones et al., 2013; Thillmann et al., 2009). They are commonly presented either during or at the end of a learning session, as self-explanation or looking-back prompts. Similarly, self-reflective questions are provided if help is requested during exploration, as described in Mavrikis (in press). Manlove et al. (2007) provides cues to remind the learner to take notes to self-reflect while monitoring whether the student has taken any notes in the last 10 minutes or has switched to a different activity.

Adaptive approaches to provide self-reflection include work by Liao et al. (2011), who provide feedback via changing a character's appearance according to the learner's performance. This is similar to Jones et al. (2013), who includes an open learner model in their exploratory learning environment, which reflects the learner's performance and interaction. Another example is Joolingen (2012), who describes a system that provides reflection of knowledge through emerging learning objects.

It can be seen that, for different reasoning stages, different approaches have been taken to support the learner within an exploratory learning environment. Machine learning has been mainly applied while the plan and task performance is carried out, while formulating goals and planning mainly involved pre-formulated prompts. Additionally, most of the self-reflection phase involved prompts or questions. However, some approaches also involved a graphical (non-textual) component that enabled metacognitive, self-reflective processes, such as the inclusion of an open learner model.

3 Requirements for adaptive intelligence for robust learning support

This section describes the requirements for the aforesaid tasks of WP2. We focus on the types of data required and modalities of testing.

3.1 Requirements on target tasks to support

This section covers the target tasks to support in respect of the recommender system (T2.2), the sequencer for planned learning (T2.4), the task-independent as well as the task-dependent support (T2.2, T2.3).

Three requirements are common to all the tasks listed below. Firstly, enough data needs to be available so that the model can abstract enough from the single examples. Secondly, the time required to apply the model needs to be evaluated in order to provide real time interaction with the system. Finally, experiments need to be planned to test the models created.

Other requirements are related to the algorithms that one chooses. Depending on the methods selected, the domain information required is different – i.e. number of possible skills and skills involved in a domain, common misconceptions, difficulty of a domain etc. This analysis needs to be done by an expert for the iTalk2Learn domain (i.e. fractions) and will be described in D1.2 and D1.3. In the following sections we will focus on the data that is collected automatically while the students interact with the system. We will also stress the technical aspects in comparison with the state-of-the-art approaches of adaptive intelligence.

We start in describing the different target tasks to support without speech indicators, followed by a description that includes speech indicators for the recommender system (problem and intervention selection), the task-dependent and the task-independent support.

Problem and intervention selection in machine learning

The requirements for this task depend upon the abstraction level it will act. Given iTalk2Learn contents, we will start sequencing the different activity in a particular domain. In order to do so, information after each exercise will be required. The data available for structured and exploratory activities is listed in the Appendix and is coherent with state-of-the-art approaches. Less information is given by the literature about how to alternate the two activities. The list of collected data was carefully selected in order to grant the feasibility of this task.

Tests for sequencing can only be done by experimenting with real students or simulating their interaction with the system. It is impossible to predict which sequence (incl. switching) will be selected because it is decided at each time step by the algorithm according to the status of the user. From the machine learning perspective, this is an added difficulty, since generally algorithms can be evaluated in a laboratory with a test set of data.

Task-dependent support for exploratory learning environment

Within the exploratory learning environment, the target tasks to be supported include goal management, planning, task performance and reflection on performance. In order to provide support for the different reasoning stages, different types of knowledge need to be modelled. This includes knowledge about the learner's goals and tasks, as well as their exploratory behaviour (including what type of representation is used in order to perform the task). Additional knowledge such as common misconceptions might also be included within the model.

Problem and intervention selection with machine learning with speech indicators

A list of possible data is given in the Appendix. For this section, we will exploit the pre-processed data available through Sail's ASR. Experimenting with state-of-the-art method and iTalk2Learn platform, we will evaluate which data can be utilised for the task (given the bottleneck created by the data transfer through the internet). The bandwidth required by the iTalk2Learn platform will have to grant usability both in schools and private houses. Given the design of Sail technology, we will focus on word usage and speech fillers ('uhm', 'eeh', 'hä' etc.). The data will be used to provide adaptive interventions. Moreover, since the task is new and no public datasets are available, we will plan to collect the dataset for this task within the second year of the project.

Task-dependent support for exploratory learning environment with speech indicators

As in the previous task, data from the speech recognition software is used to detect keywords and speech fillers. With this information, the learning environment will be able to respond to certain keywords, such as 'help' or 'I do not know' (n-grams); and might even help in detecting students' affective states (e.g., frustration), which can be used to adapt the support provided.

Task-independent support with speech indicators

The task-independent support will act across the different learning systems (Fractions Tutor, Whizz, and ELE), additionally exploiting the data from the ASR. The support provided here will be independent of the current learning system. It is being designed in the context of WP1.

3.2 Requirements on transaction for data collection

The purpose of this section is to cover the dataflow between the different components needed for modelling. In order to perform the tasks mentioned above, the recommender system needs to communicate with the different learning systems (Fractions Tutor, Whizz, ELE) to provide appropriate sequencing and switching of the learning exercises. Additionally, the task-dependent support for the exploratory learning environment needs to access information directly from the exploratory environment as well as from the speech recognition software. The task-independent support needs to communicate with the speech recognition software (and possibly also with the different learning systems).

Figure 9 shows the data flow between the different components that is relevant for modelling.

The iTalk2Learn platform is at the center of communication between the different components. Every component is in communication with it. The first prototype of the iTalk2Learn platform includes the communication between Fractions Tutor, Whizz, the exploratory learning environment, and the speech recognition software. D4.1 gives a detailed description of how those components interact with the platform.

For the communication with the Fractions Tutor, a log file is sent to the platform (during the learning session), which includes different types of information such as Attempt, Result, Hint request, and Hint message.

Whizz is able to return a set of data at the end of each learning session which includes information about the learner's performance.

The exploratory learning environment is still in development. Here it will be necessary to provide data concerning the student's behaviour during each learning session.

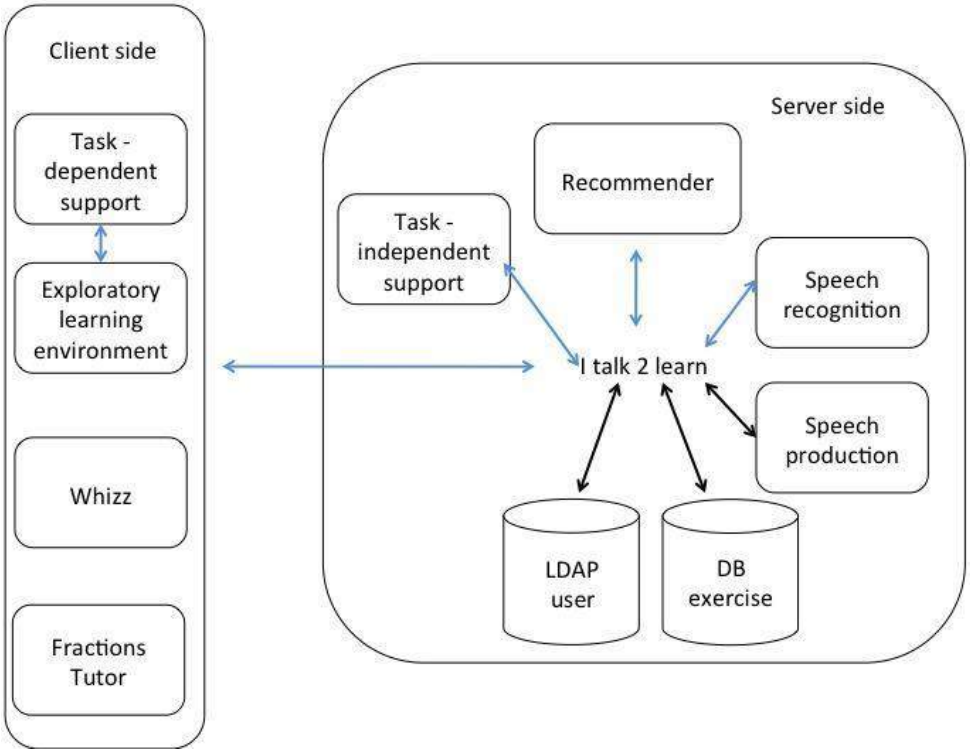

Fig. 9 Main platform and components of iTalk2Learn with highlighted data flow relevant for modelling

The speech recognition data provides a transcript with the student's words in a textual format.

The recommender system is able to access all the necessary information from the iTalk2Learn platform in order to provide the sequencing and switching of the relevant exercises, such as student performance with a structured learning activity etc. The different learning systems (Fractions Tutor, Whizz, ELE) will provide the relevant information about the current student.

The task-dependent support communicates with the exploratory learning environment. Data about the learner's behaviour will be used to provide the intelligent support. Additionally, the iTalk2Learn platform needs to provide the relevant information from the speech recognition software. This can then be used to provide support based on the student's speech.

The task-independent support component communicates with the iTalk2Learn platform and receives data from the speech recognition software. Support will be provided based on the student's speech. This component might also send information to the different learning systems (Fractions Tutor, Whizz, ELE) in accordance with the support provided.

3.3 Requirements on background data collection

At the moment we are able to detect the following information from Fractions Tutor, Whizz and the speech recognition software through the iTalk2Learn platform. However, there is additional data that could be extracted, which is listed in the Appendix. We focus on the data that is already given by the different components and will evaluate whether to extend the module interfaces later.

Fractions Tutor:

- Type of log (Attempt, Result, Hint request, Hint message)
- Date and time of the exercise
- Name of the exercise
- Action of the exercise (e.g. done button pressed)
- Result of the exercise (e.g. correct or incorrect)
- Number of the steps in the exercise (e.g. 17 steps)
- Hint

Whizz:

- Learning objective
- Number of questions in the exercise
- Current question number
- Current percentage of questions answered
- Current questions answered correctly (score)
- Amount of help provided at different levels
- Timing data (e.g. time answering questions)
- Paper-based or Exercise
- Mode (Assessment, Tutor, Replay)
- Credits earned
- Final outcome (pass, fail, static)

Sail ASR:

Our partners, IOE, RUB and Sail, will help developing a vocabulary

- Fillers (yeah, ehm, eeh, uh huh etc.)
- State words (boring, easy, difficult etc.)
- Swear words
- Technical words (numerator, factor, denominator, etc.)
- n-grams: ("It is a piece of cake", "I don't like..", etc.)

Both German and English vocabularies need to be written.

The keywords selected can be used in the different tasks creating new features like:

- Percentage of technical word used
- Coherence between state words and outcome
- Mood retrieval

For other features see the Appendix.

For the **exploratory learning environment** (which is in development) the following data needs to be extracted in order to provide task-dependent support:

- Interaction with and position of the various objects available in a task
- Timing data
- Which representation was selected (at what time)

The iTalk2Learn recommender system will use the aforementioned data to create its internal representation of the student and alternate structured and exploratory activities with and without speech indicators.

4 Conclusion

Different machine learning techniques have been applied for modelling in intelligent tutoring systems and exploratory learning environments. We discussed how these techniques have been applied and what types of data they model. Some of these techniques can be appropriately adapted for the needs of the components to be developed in iTalk2Learn.

Different requirements for adaptive intelligence for robust learning support have been developed. We focused on the target tasks to support, the data available and needed for modelling has been provided.

The requirements provide the basis for the development of the recommender, the task-dependent support for the exploratory learning environment and the task-independent support. A complete list of available data was made and suggestions for the new tasks were done. Finally, we also established testing modalities for these tasks and provided the application constrains of the mentioned algorithms.

References

Beck, Joseph (1997) Modeling the student with reinforcement learning. In *Machine learning for User Modeling Workshop at the Sixth International Conference on User Modeling*.

Beck, Joseph, Beverly Park Woolf, and Carole R. Beal (2000) ADVISOR: A machine learning architecture for intelligent tutor construction. In *AAAI/IAAI* 2000, 552-557.

Beck, Joseph E., and Beverly Park Woolf (2000) High-level student modeling with machine learning. In *Intelligent tutoring systems*. Springer Berlin Heidelberg.

Baker, Ryan SJ, Albert T. Corbett, and Vincent Aleven (2008) More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *Intelligent Tutoring Systems*. Springer Berlin Heidelberg.

Baker, R., Walonoski, J., Heffernan, N., Roll, I., Corbett, A., & Koedinger, K. (2008). Why Students Engage in "Gaming the System. In Behavior in Interactive Learning Environments. Journal of Interactive Learning Research, 19(2), 185-224.

Baker, R.S.J.,Pardos, Z.A., Gowda, S.M., Nooraei, B.B., Heffernan, N.T. (2011) Ensembling predictions of student knowledge within intelligent tutoring systems. In *User Modeling, Adaption and Personalization*. Springer Berlin Heidelberg, 2011. 13-24.

Bunt A. and Conati C. (2003) Probabilistic Student Modelling to Improve Exploratory Behaviour. In User Modeling and User-Adapted Interaction, 3.

Cen, Hao, Kenneth Koedinger, and Brian Junker (2006) Learning factors analysis–a general method for cognitive model evaluation and improvement. In *Intelligent Tutoring Systems*. Springer Berlin Heidelberg..

Chang, M. M. (2007) Enhancing web-based language learning through self-monitoring. In Journal of Computer Assisted Learning, 23 (3), 187-196.

Chang, K., Beck, J., Mostow, J., Corbett, A. (2006) A bayes net toolkit for student modeling in intelligent tutoring systems. In *Intelligent Tutoring Systems*. Springer Berlin Heidelberg.

Chavhan, Yashpalsing, M. L. Dhore, and Pallavi Yesaware (2010) Speech emotion recognition using support vector machine. In *International Journal of Computer Applications* 1.20, 6-9.

Chi, M., VanLehn, K., Litman, D., Jordan, P. (2010) Inducing effective pedagogical strategies using learning context features. In *User Modeling, Adaptation, and Personalization*. Springer Berlin Heidelberg. 147-158.

Chi, M., VanLehn, K., Litman, D., Jordan, P. (2011) Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical strategies. In *User Modeling and User-Adapted Interaction* 21.1-2, 137-180.

Chieu, V.M., Luengo, V., Vadcard, L., Tometti, J. (2010) Student Modeling in Orthopedic Surgery Training: Exploiting Symbiosis between Temporal Bayesian Networks and Fine-grained Didactic Analysis. In *International Journal of Artificial Intelligence in Education,* 20(3), 269-301.

Cocea, M., Gutierrez-Santos, S., Magoulas, D. (2010) Adaptive Modelling of Users' Strategies in Exploratory Learning Using Case-Based Reasoning. In Proceedings of the 14th International Conference of Knowledge-Based and Intelligent Information and Engineering Systems, Lecture Notes in Computer Science, Vol. 6277, 124-134.

Colancies, J.D., Nussbaum, E.M. (2008) Enhancing online collaborative argumentation through question elaboration and goal instructions. Journal of Computer Assisted Learning, 24 (3), 167-180.

Conati, Cristina, Abigail Gertner, and Kurt Vanlehn. (2002) Using Bayesian networks to manage uncertainty in student modeling. In *User modeling and user-adapted interaction,* 12(4), 371-417.

Conati C. and Zhou X. (2002). Modeling Students' Emotions from Cognitive Appraisal in Educational Games. In Proceedings of ITS 2002, 6th International Conference on Intelligent Tutoring Systems, Biarritz, France.

Conati, C., Jaques, N., Muir, M. (2013) Understanding attention to adaptive hints in educational games: an eye-tracking study. In International Journal of Artificial Intelligence in Education, 23.

Corbett, Albert T., and Anderson, John R. (1994) Knowledge tracing: Modeling the acquisition of procedural knowledge. In *User modeling and user-adapted interaction,* 4(4), 253-278.

Crippen, K.J., Earl, B.L. (2007) The impact of web-based worked examples and self-explanation on performance, problem solving, and self-efficacy. In Computers & Education, 49, 809-821.

Davis, E.A., Linn, M.C. (2000) Scaffolding students' knowledge integration: prompts for reflection in KIE. In International Journal of Science Education, 22.

Diziol, D., Walker, E., Rummel, N., Koedinger, K.R. (2010) Using intelligent tutor technology to implement adaptive support for student collaboration. In *Educational Psychology Review,* 22(1), 89-102.

Devolder, A., van Braak, J., Tondeur, J. (2012) Supporting self-regulated learning in computer-based learning environments: systematic review of effects of scaffolding in the domain of science education. In Journal of Computer Assisted Learning, 28(6), 557-573.

Ergazaki, M., Zogza, V., Komis, V. (2007) Analysing students‘ shared activity while modeling a biological process in a computer-supported educational environment. In Journal of Computer Assisted Learning, 23(2), 158-168.

Feng, M., Heffernan, N.T., & Koedinger, K.R. (2009). Addressing the assessment challenge in an Intelligent Tutoring System that tutors as it assesses. *The Journal of User Modeling and User-Adapted Interaction, 19*, 243-266.

Fund, Z. (2007) The effects of scaffolded computerized science problem-solving on achievement outcomes: a comparative study of support programs. In Journal of Computer Assisted Learning, 23 (5), 410-424.

Furberg, A. (2009) Socio-cultural aspects of prompting student reflection in Web-based inquiry learning environments. In Journal of Computer Assisted Learning, 25(4), 397-409.

Gantner, Z., Drumond, L., Freudenthaler, C., Rendle, S., Schmidt-Thieme, L.(2010) Learning attribute-to-feature mappings for cold-start recommendations. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE.

Gong, Y., Beck, J.E., Heffernan, N.T. (2010) Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures. In *Intelligent Tutoring Systems*. Springer Berlin Heidelberg.

González-Brenes, J.P., Mostow, J. (2013) What and When do Students Learn? Fully Data-Driven Joint Estimation of Cognitive and Student Models. In Proceeding of the 6th International Conference on Educational Data Mining, 236-240.

González-Brenes, J.P., Mostow, J. (2012) Dynamic Cognitive Tracing: Towards Unified Discovery of Student and Cognitive Models. *EDM*. 2012.

Graesser, A.C., Chipman, P., Haynes, B.C., Olney, A. (2005) AutoTutor: An intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions on Education,* 48(4), 612-618.

Gurlitt, J., Renkl, A. (2008) Are high-coherent concept maps better for prior knowldge activation? Different effects of concept mapping tasks on high school vs. university students. In Journal of Computer Assisted Learning, 24.

Gutierrez-Santos, S., Mavrikis, M., Magoulas, D. (2012) A Separation of Concerns for Engineering Intelligent Support for Exploratory Learning Environments. In Journal of Research and Practice in Information Technology, 44(3), 347-360.

Harrison, B., and Roberts, D.L. (2012) A Review of Student Modeling Techniques in Intelligent Tutoring Systems. In Proceedings of *Eighth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Iglesias, A., Martinez, P., Fernández, F. (2003) An experience applying reinforcement learning in a web-based adaptive and intelligent educational system. In *Informatics in Education,* 2(2), 223-240.

Iglesias, A., Martinez, P., Aler, R., Fernández, F (2009) Learning teaching strategies in an adaptive and intelligent educational system through reinforcement learning.In *Applied Intelligence* 31(1), 89-106.

Joolingen van, W. (2012) Supporting inquiry learning based on Emerging Learning Objects. In Proceedings of the Intelligent Support for Exploratory Environments workshop at ITS 2012.

Jones, A., Bull, S., Castellano, G. (2013) Teacher Perspectives on the Potential for Scaffolding with an Open Learner Model and an Empathic Robot. In Proceedings of Scaffolding in Open-Ended Learning Environments at AIED 2013.

Joshi, A., Kaur, K. (2013) A Study of Speech Emotion Recognition Methods.In International Journal of Computer Science and Information Technology (IJCSMC), 2(4).

Joshi, D.D., Zalte, M.B. (2013) Speech Emotion Recognition: A Review. In Journal of Electronics and Communication Engineering (IOSR-JECE), 4(4).

Karakostas, A., Demetriadis, S. (2011) Enhancing collaborative learning through dynamic forms of support: the impact of an adaptive domain-specific support strategy. In Journal of Computer Assisted Learning, 27(3), 243-258.

Kardan, S., and Conati, C. (2012) Providing Adaptive Support in an Exploratory Learning Environment Using Interaction Data. In Proceedings of the Intelligent Support for Exploratory Environments workshop at ITS 2012.

Kickmeier-Rust, M.D., Albert, D. (2010) Micro-adaptivity: protecting immersion in didactically adaptive digital educational games. In Journal of Computer Assisted Learning, 26(2), 95-105.

Kim, J., Hill, R.W., Durlach, P., Lane, H.C., Forbell, E., Core, M., Marsella, S., Pynadath, D., Hart, J. (2009) BiLAT: A Game-Based Environment for Practicing Negotiation in a Cultural Context. In International Journal of Artificial Intelligence in Education, 19(3), 289-308.

Koedinger, K.R., Pavlik, P.I., Stamper, J., Nixon, T., Ritter, S. (2010) Avoiding Problem Selection Thrashing with Conjunctive Knowledge Tracing. In *EDM*, 91-100 .

Krothapalli, S.R., Koolagudi, S.G. (2013) Speech Emotion Recognition: A Review. In *Emotion Recognition using Speech Features*. Springer New York, 15-34.

Krohn-Grimberghe, A., Busche, A., Nanopoulos, A. (2011) Active learning for technology enhanced learning. *Towards Ubiquitous Learning*. Springer Berlin Heidelberg, 512-518.

Lallé, S., Mostow, J., Vanda, L., Guin, N. (2013) Comparing Student Models in Different Formalisms by Predicting their Impact on Help Success. In Proceedings of the 16th International Conference, AIED 2013.

Lee, J.I., Brunskill, E. (2012) The Impact on Individualizing Student Models on Necessary Practice Opportunities." In *EDM*.

Liao, C.C.Y., Chen, Z.-H., Cheng, N.H.H., Chen, F.-C., Chan, T.-W. (2011) My-Mini-Pet: a handheld pet-nurturing game to engage students in arithmetic practices. In Journal of Computer Assisted Learning, 27(1), 76-89.

Lindstoem, P., Gulz, A., Haake, M., Sjoeden, B. (2011) Matching and mismatching between the pedagogical design principles of a math game and the actual practices of play. In Journal of Computer Assisted Learning, 27(1), 90-102.

Lintean, M., Rus, V., Azevedo, R. (2011) Automatic Detection of Student Mental Models Based on Natural Language Student Input During Metacognitive Skill Training. In International Journal of Artificial Intelligence in Education, 21 (3), 169-190.

Litman, D.J., Silliman, S. (2004) ITSPOKE: An intelligent tutoring spoken dialogue system. In *Demonstration Papers at HLT-NAACL 2004*. Association for Computational Linguistics.

Manlove, S., Lazonder, A.W., de Jong, T. (2007) Software scaffolds to promote regulation during scientific inquiry learning. In Metacognition and Learning, 2, 141-155.

Manlove, S., Lazonder, A.W., de Jong, T. (2009) Trends and issues of regulative support use during inquiry learning: patterns from three studies. In Computers in Human Behavior, 25.

Martin, K.N., Arroyo, I. (2004) AgentX: Using reinforcement learning to improve the effectiveness of intelligent tutoring systems. In *Intelligent Tutoring Systems*. Springer Berlin Heidelberg.

Mavrikis, M. (2010) Modelling Student Interactions in Intelligent Learning Environments: Constructing Bayesian Networks from Data. In *International Journal on Artificial Intelligence Tools,* 19(6), 733-753.

Mavrikis, M., Gutierrez-Santos, S., Geraniou, E., Noss, R. (in press) *Design Requirements and Validation Metrics for Adaptive Exploratory Learning Environments: From pedagogic strategies to computer-based support*. Journal of Personal and Ubiquitous Computing.

Mao, X., Chen, L., Fu, L. (2009) Multi-level speech emotion recognition based on HMM and ANN. In *Computer Science and Information Engineering, 2009 WRI World Congress,* 7.

van der Meij, J., de Jong, T. (2011) The effects of directive self-explanation prompts to support active processing of multiple representations in a simulation-based learning environment. In Journal of Computer Assisted Learning, 27(5), 411-423.

Muir, M.M.A. (2012) Prime Climb: an analysis of attention to student-adaptive hints in an educational game. MSc Thesis. University of British Columbia.

Moos, D.C., Azevedo, R. (2008) Exploring the fluctuation of motivation and use of self-regulatory processes during learning with hypermedia. In Instructional Science, 36.

Nash, P., Willamson Shaffer, D. (2011) Mentor modeling: the internalization of modeled professional thinking in an epistemic game. In Journal of Computer Assisted Learning, 27(2), 173-189.

Nooraei, B., Pardos, Z.A., Heffernan, N.T., Baker, R.S.J.D (2011) Less is More: Improving the Speed and Prediction Power of Knowledge Tracing by Using Less Data. In *EDM*.

Pavlik, P.I., Cen,H., and Koedinger, K.R. (2009) Performance Factors Analysis-A New Alternative to Knowledge Tracing. In *AIED*.

Pardos, Z.A., Heffernan, N.T. (2010) Modeling individualization in a bayesian networks implementation of knowledge tracing. In *User Modeling, Adaptation, and Personalization*. Springer Berlin Heidelberg. 255-266.

Pardos, Z.A., Heffernan, N.T. (2010) Using HMMs and bagged decision trees to leverage rich features of user and skill from an intelligent tutoring system dataset. In *Journal of Machine Learning Research W & CP*.

Pardos, Z.A., Heffernan, N.T. (2011) KT-IDEM: introducing item difficulty to the knowledge tracing model.In *User Modeling, Adaption and Personalization*. Springer Berlin Heidelberg, 243-254.

Gowda, S.M., Baker, R.S.J.D, Pardos, Z., Heffernan, N.T. (2012) The sum is greater than the parts: ensembling models of student knowledge in educational software. In *ACM SIGKDD Explorations Newsletter*, 13(2), 37-44.

Pintrich, P.R. (2000) The role of goal orientation in self-regulated learning. In Handbook of Self-Regulatio, 451-502. Academic Press, San Diego, CA.

Pólya, G. (1945) *How to Solve It*. Princeton University Press. ISBN 0-691-08097-6.

Porayska-Pomsta, K., Mavrikis, M., D'Mello, S., Conati, C., Baker, R.S.J.D. (2013) Knowledge Elicitation Methods for Affect Modelling in Education. In International Journal of Artificial Intelligence in Education, 23.

Qiu, Y., Qi, Y., Lu, H., Pardos, Z.A., Heffernan, N.T. (2011) Does Time Matter? Modeling the Effect of Time with Bayesian Knowledge Tracing. In *EDM*. 2011.

Ravindran, B., Sarma, B.H. (2010) Intelligent Tutoring Systems using Reinforcement Learning to teach Autistic Students." *International Federation for Information Processing Digital Library* 241(1).

Rendle, S. (2010) Factorization machines. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE.

Reye, J. (2004) Student modelling based on belief networks. In *International Journal of Artificial Intelligence in Education* 14(1), 63-96.

Roll, I., McLaren, B.M., Koedinger, K.R. (2011) Improving students' help-seeking skills using metacognitive feedback in an intelligent tutoring system. In *Learning and Instruction,* 21(2), 267-280.

Sabourin, J.L., Shores, L.R., Mott, B.W., Lester, J. (2013) Understanding and Predicting Student Self-Regulated Learning Strategies in Game-Based Learning Environments. In International Journal of Artificial Intelligence in Education, 23.

Schoenfeld, A. H. (1992). Learning to think mathematically: Problem solving, metacognition, and sense-making in mathematics. In D. Grouws (Ed.), Handbook for Research on Mathematics Teaching and Learning (pp. 334-370). New York: MacMillan.

Simons, K.D., Klein, J.D. (2007) The impact of scaffolding and student achievement levels in a problem-based learning environment. In Instructional Science, 35.

Sutton, R.S., Barto, A.G. (1998) *Reinforcement learning: An introduction*. Vol. 1. No. 1. Cambridge: MIT press.

Thai-Nghe, N., Drumond, L., Krohn-Grimberghe, A., Schmidt-Thieme, L. (2010) Recommender system for predicting student performance. In *Procedia Computer Science,* 1(2), 2811-2819.

Thai-Nghe, N., Drumond, L., Horvath, T., Schmidt-Thieme, L. (2011) Multi-relational factorization models for predicting student performance. In *KDD 2011 Workshop on Knowledge Discovery in Educational Data, KDDinED*.

Thai-Nghe, N., Drumond, L., Horvath, T., Krohn-Grimberghe, A., Nanopoulos, A., Schmidt-Thieme, L. (2011) Factorization techniques for predicting student performance. In *Educational Recommender Systems and Technologies: Practices and Challenges (In press). IGI Global* (2011).

Thai-Nghe, N., Drumond, L., Horvath, T., Nanopoulos, A., Schmidt-Thieme, L. (20100) Matrix and Tensor Factorization for Predicting Student Performance. In *CSEDU (1)*.

Thai-Nghe, N. Horvath, T. Schmidt-Thieme, L. (2011) Context-Aware Factorization for Personalized Student's Task Recommendation. In *Proceedings of the International Workshop on Personalization Approaches in Learning Environments*. Vol. 732.

Thai-Nghe, N., Horváth, T., Schmidt-Thieme, L. (2011) Personalized forecasting student performance. In *Advanced Learning Technologies (ICALT), 2011 11th IEEE International Conference on*. IEEE.

Thillmann, H., Kunsting, J., Wirth, J., Leutner, D. (2009) Is it merely a question of 'what' to a prompt or also 'when' to a prompt? The role of point of presentation time of prompts in self-regulated learning. Zeitschrift Fuer Pedagogische Psychologie, 23, 105-115.

Trudel, C., Payne, S.J. (1995) *Reflection and goal management in exploratory learning*. In International Journal of Human-computer Studies, 42(3), 307-339.

VanLehn, K., Jordan, P.W., Rose, C.P., Bhembe, D., Bottner, M., Gaydos, A., Makatchev, M., Pappuswamy, U., Ringenberg, M., Roque, A., Siler, S., Srivastava, R. (2002) The architecture of Why2-Atlas: A coach for qualitative physics essay writing. In *Intelligent tutoring systems*. Springer Berlin Heidelberg.

VanLehn, K., Jordan, P. Litman, D. (2007) Developing pedagogically effective tutorial dialogue tactics: Experiments and a testbed. In *Proceedings of SLaTE Workshop on Speech and Language Technology in Education ISCA Tutorial and Research Workshop*.

Vapnik, V.N. (1998) Statistical learning theory. Wiley, New York. NY.

Ververidis, D., Kotropoulos, C. (2006) Emotional speech recognition: Resources, features, and methods. In *Speech communication,* 48(9), 1162-1181.

Wang, Y., Heffernan, N.T. (2011) Extending Knowledge Tracing to Allow Partial Credit: Using Continuous versus Binary Nodes. Springer-Verlag.

Wang, Y., Heffernan, N.T. (2012) The student skill model. In *Intelligent Tutoring Systems*. Springer Berlin Heidelberg.

Webber, C., Pesty, S., Balacheff, N. (2002) A multi-agent and emergent approach to learner modelling. In Proceedings of the 8th Iberoamerican Conference on Artificial Intelligence.

Whitehill, J., Bartlett, M., Movellan, J. (2008) Automatic facial expression recognition for intelligent tutoring systems. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*. IEEE.

Worsley, M., Blikstein, P. (2011) What's an Expert? Using Learning Analytics to Identify Emergent Markers of Expertise through Automated Speech, Sentiment and Sketch Analysis. In *EDM*.

Xu, Y., Mostow, J. (2011) Logistic Regression in a Dynamic Bayes Net Models Multiple Subskills Better!. In *EDM*.

Xu, Y., Mostow, J. (2012) Comparison of methods to trace multiple subskills: Is LR-DBN best?. In *EDM*. 2012.

Xu, Y., Mostow, J. (2013) Using Item Response Theory to Refine Knowledge Tracing. In EDM.

Appendix

State-of-the-art resume of paper Ververidis, Kotropoulus (2006) Fig 10 and 11

D. Ververidis, C. Kotropoulos / Speech Communication 48 (2006) 1162–1181

Table 1
Emotional speech data collections (in alphabetical ordering of the related references)

Reference	Language	Subjects	Other signals	Purpose	Emotions	Kind
Abelin and Allwood (2000)	Swedish	1 Native	–	Recognition	Ar, Fr, Jy, Sd, Se, Dt, Dom, Sy	Simulated
Alpert et al. (2001)	English	22 Patients, 19 healthy	–	Recognition	Dn, Nl	Natural
Alter et al. (2000)	German	1 Female	EEG	Recognition	Ar, Hs, Nl	Simulated
Ambrus (2000), Interface	English, Slovenian	8 Actors	LG	Synthesis	Ar, Dt, Fr, Nl, Se	Simulated
Amir et al. (2000)	Hebrew	40 Students	LG,M, G,H	Recognition	Ar, Dt, Fr, Jy, Sd	Natural
Ang et al. (2002)	English	Many	–	Recognition	An, At, Nl, Fd, Td	Natural
Banse and Scherer (1996)	German	12 Actors	V	Recognition	H/C Ar, Hs, Sd, ...	Simulated
Batliner et al. (2004)	German, English	51 Children	–	Recognition	Ar, Bm, Jy, Se	Elicited
Bulut et al. (2002)	English	1 Actress	–	Synthesis	Ar, Hs, Nl, Sd	Simulated
Burkhardt and Sendlmeier (2000)	German	10 Actors	V, LG	Synthesis	Ar, Fr, Jy, Nl, Sd, Bm, Dt	Simulated
Caldognetto et al. (2004)	Italian	1 Native	V, IR	Synthesis	Ar, Dt, Fr, Jy, Sd, Se	Simulated
Choukri (2003), Groningen	Dutch	238 Native	LG	Recognition	Unknown	Simulated
Chuang and Wu (2002)	Chinese	2 Actors	–	Recognition	Ar, Ay, Hs, Fr, Se, Sd	Simulated
Clavel et al. (2004)	English	18 From TV	–	Recognition	Nl, levels of Fr	Simulated
Cole (2005), Kids' Speech	English	780 Children	V	Recognition, Synthesis	Unknown	Natural
Cowie and Douglas-Cowie (1996), Belfast Structured	English	40 Native	–	Recognition	Ar, Fr, Hs, Nl, Sd	Natural
Douglas-Cowie et al. (2003), Belfast Natural	English	125 From TV	V	Recognition	Various	Semi-natural
Edgington (1997)	English	1 Actor	LG	Synthesis	Ar, Bm, Fr, Hs, Nl, Sd	Simulated
Engberg and Hansen (1996), DES	Danish	4 Actors	–	Synthesis	Ar, Hs, Nl, Sd, Se	Simulated
Fernandez and Picard (2003)	English	4 Drivers	–	Recognition	Nl, Ss	Natural
Fischer (1999), Verbmobil	German	58 Native	–	Recognition	Ar, Dn, Nl	Natural
France et al. (2000)	English	70 Patients, 40 healthy	–	Recognition	Dn, Nl	Natural
Gonzalez (1999)	English, Spanish	Unknown	–	Recognition	Dn, Nl	Elicited
Hansen (1996), SUSAS	English	32 Various	–	Recognition	Ar, Ld eff., Ss, Tl	Natural, simulated
Hansen (1996), SUSC-0	English	18 Non-native	H,BP,R	Recognition	Nl, Ss	A-stress
Hansen (1996), SUSC-1	English	20 Native	–	Recognition	Nl, Ss	P-stress
Hansen (1996), DLP	English	15 Native	–	Recognition	Nl, Ss	C-stress
Hansen (1996), DCIEM	English	Unknown	–	Recognition	Nl, Sleep deprive	Elicited
Heuft et al. (1996)	German	3 Native	–	Synthesis	Ar, Fr, Jy, Sd, ...	Simulated, elicited
Iida et al. (2000), ESC	Japanese	2 Native	–	Synthesis	Ar, Jy, Sd	Simulated

Fig.10: Table 1 state-of-the-art resume of paper Ververidis, Kotropoulus (2006).

D. Ververidis, C. Kotropoulos / Speech Communication 48 (2006) 1162–1181

Iriondo et al. (2000)	Spanish	8 Actors	–	Synthesis	Fr, Jy, Sd, Se,...	Simulated
Kawanami et al. (2003)	Japanese	2 Actors	–	Synthesis	Ar, Hs, Nl, Sd	Simulated
Lee and Narayanan (2005)	English	Unknown	–	Recognition	Negative–positive	Natural
Liberman (2005), Emotional Prosody	English	Actors	–	Unknown	Anxty, H/C Ar, Hs, Nl, Pc, Sd, Se,...	Simulated
Linnankoski et al. (2005)	English	13 Native	–	Recognition	An, Ar, Fr, Sd,...	Elicited
Lloyd (1999)	English	1 Native	–	Recognition	Phonological stress	Simulated
Makarova and Petrushin (2002), RUSSLANA	Russian	61 Native	–	Recognition	Ar, Hs, Se, Sd, Fr, Nl	Simulated
Martins et al. (1998), BDFALA	Portuguese	10 Native	–	Recognition	Ar, Dt, Hs, Iy	Simulated
McMahon et al. (2003), ORESTEIA	English	29 Native	–	Recognition	Ae, Sk, Ss	Elicited
Montanari et al. (2004)	English	15 Children	V	Recognition	Unknown	Natural
Montero et al. (1999), SES	Spanish	1 Actor	–	Synthesis	Ar, Dt, Hs, Sd	Simulated
Mozziconacci and Hermes (1997)	Dutch	3 Native	–	Recognition	Ar, Bm, Fr, Jy, Iy, Nl, Sd	Simulated
Niimi et al. (2001)	Japanese	1 Male	–	Synthesis	Ar, Jy, Sd	Simulated
Nordstrand et al. (2004)	Swedish	1 Native	V, IR	Synthesis	Hs, Nl	Simulated
Nwe et al. (2003)	Chinese	12 Native	–	Recognition	Ar, Fr, Dt, Jy,...	Simulated
Pereira (2000)	English	2 Actors	–	Recognition	H/C Ar, Hs, Nl, Sd	Simulated
Petrushin (1999)	English	30 Native	–	Recognition	Ar, Fr, Hs, Nl, Sd	Simulated, Natural
Polzin and Waibel (2000)	English	Unknown	–	Recognition	Ar, Fr, Nl, Sd	Simulated
Polzin and Waibel (1998)	English	5 Drama students	LG	Recognition	Ar, Fr, Hs, Nl, Sd	Simulated
Rahurkar and Hansen (2002), SOQ	English	6 Soldiers	H, R, BP, BL	Recognition	5 Stress levels	Natural
Scherer (2000b), Lost Luggage	Various	109 Passengers	V	Recognition	Ar, Hr, Ie, Sd, Ss	Natural
Scherer (2000a)	German	4 Actors	–	Ecological	Ar, Dt, Fr, Jy, Sd	Simulated
Scherer et al. (2002)	English, German	100 Native	–	Recognition	2 Tl, 2 Ss	Natural
Schiel et al. (2002), SmartKom	German	45 Native	V	Recognition	Ar, Dfn, Nl	Natural
Schröder and Grice (2003)	German	1 Male	–	Synthesis	Soft, modal, loud	Simulated
Schröder (2000)	German	6 Native	–	Recognition	Ar, Bm, Dt, Wy,...	Simulated
Slaney and McRoberts (2003), Babyears	English	12 Native	–	Recognition	Al, An, Pn	Natural
Stibbard (2000), Leeds	English	Unknown	–	Recognition	Wide range	Natural, elicited
Tato (2002), AIBO	German	14 Native	–	Synthesis	Ar, Bm, Hs, Nl, Sd	Elicited
Tolkmitt and Scherer (1986)	German	60 Native	–	Recognition	Cognitive Ss	Elicited
Wendt and Scheich (2002), Magdeburger	German	2 Actors	–	Recognition	Ar, Dt, Fr, Hs, Sd	Simulated
Yildirim et al. (2004)	English	1 Actress	–	Recognition	Ar, Hs, Nl, Sd	Simulated
Yu et al. (2001)	Chinese	Native from TV	–	Recognition	Ar, Hs, Nl, Sd	Simulated
Yuan (2002)	Chinese	9 Native	–	Recognition	Ar, Fr, Jy, Nl, Sd	Elicited

Abbreviations for emotions: The emotion categories are abbreviated by a combination of the first and last letters of their name. At: Amusement, Ay: Antipathy, Ar: Anger, Ae: Annoyance, Al: Approval, An: Attention, Anxty: Anxiety, Bm: Boredom, Dfn: Dissatisfaction, Dom: Dominance, Dn: Depression, Dt: Disgust, Fd: Frustrated, Fr: Fear, Hs: Happiness, Ie: Indifference, Iy: Irony, Jy: Joy, Nl: Neutral, Pc: Panic, Pn: Prohibition, Se: Surprise, Sd: Sadness, Ss: Stress, Sy: Shyness, Sk: Shock, Td: Tiredness, Tl: Task load stress, Wy: Worry. Ellipses denote that additional emotions were recorded.
Abbreviations for other signals: BP: Blood pressure, BL: Blood examination, EEG: Electroencephalogram, G: Galvanic skin response, H: Heart beat rate, IR: Infrared Camera, LG: Laryngograph, M: Myogram of the face, R: Respiration, V: Video.
Other abbreviations: H/C: Hot/cold, Ld eff.: Lombard effect, A-stress, P-stress, C-stress: Actual, Physical, and Cognitive stress, respectively, Sim.: Simulated, Elic.:Elicited, N/A: Not available.

Fig.11: Table 2 state-of-the-art resume of paper Ververidis, Kotropoulus (2006).

Other speech recognition features:

Speech Features	Availability
Mel-Frequency Cepstrum Coefficients	feasible
Linear Predictive Cepstrum Coefficients	no
duration	From the platform
filled pause, fillers (ehm, uhm, hä, eh)	yes, consider they are cultural dependent
Restarts	yes
Word Usage Analysis in order to understand the mood	yes
N-gram analysis, i.e. if patterns like "I don't know", "well, you know",etc. are present	yes, a list will be collected
The number of times technical word has occurred in the current dialogue. This feature reflects the students' familiarity with the current topic.	yes

The average number of words per student turn. This reflects the student's level of activity and verbosity.	yes
Pitch analysis	no
Number of harmonics	no
Vocal tract features: Linear prediction Coefficients for vocal tract resonance, multi-tube lossless model for the cross section area and length, MFCC for band energy	feasible
Speech energy	feasible
Mean	feasible
Range	feasible
Variance	feasible
Pitch contour trends	no
Mean and range of the intensity contour	no
Rate of speech and transmission duration between utterances	feasible

Other Whizz features:
This sheet describes all columns provided within the datasets:

- lesson_history_id.csv
- Starting Topics.xlsx
- Student info.xlsx
- Teachers.xlsx

Lesson history ID

column name	**value range**	**example values**	**description / notes**
lesson_history_id	int		primary key of the dataset
pupilid	int		id of a single pupil
engineid	int	0; 1; 2	id of the Whizz System used
curriculumid	int	1 - 1223	curriculum id
regression	int	-1; 0; 1	Was the difficulty of the task regressed
topicage	int	Used with age and exerciseid to identify a single exercise	Difficulty evel of the task
topicid	character(2)	AA, BA, CA, DA, EA, FA, GA, HA, JA, KA, LA, MA, NA, PA, QA, RA, SA, TA, UA, VA, ZA	id of a topic
age	int	Used with topicage and exerciseid to identify a single exercise	Niveau of the student
exerciseid	int	Used with age and topicage to identify a single exercise	ID of an exercise

exercisetype	Character(1)	p, x	Type of an exercise: exercise or test
run_mode	character(1)	u, x, a, p, r, s	Different modalities exercises can be run
mark	int	0 - 100	Mark
score	double	0-100	percentage of correct answers
Total questions	int / NULL		
Time taken	text	HH:mm:ss	Time a student has needed for solving a Task
marked	text	YYYY-MM-dd HH:mm:ss	Date and time a task was marked
help1	int		Number of times hint 1 was asked
help2	int		Number of times hint 2 was asked
help3	int		Number of times hint 3 was asked
credit	int	Credits for a game	-
coins	int	Coins to be used in a game	-

Starting Topics

Statistical information available.

column name	**description / notes**
pupilid	Id of a single pupil
topicid	Id of a topic

topicIdNext	Id of the next topic
exerciseIdNext	Id of the next exercise
exerciseTypeNext	Type of the next exercise

Student info

column name	**description / notes**
pupilid	ID of a pupil
credit	(Int) Credits a pupil has earned
credit_spent	(int) Credits a pupil has spent
challenge_coins	(int) Coins a pupil has earned
challenge_copins_spent	(int) Coins a pupil has spent
assessmentStarted	Date the student started his evaluation in the Whizz System , to receive an initial math age
assessmentCopleted	Date the student finished his evaluation in the Whizz System , to receive an initial math age
assessment_reset	IIn case an assessment needs to start again for an error

currentAge	Current knowledge level
AssessmentMaxAge	Maximal knowledge in a topic
AssessmentMinAge	Minimal knowledge in a topic
dateOfBirth	Birthday of a pupil
classid	ID of a class a pupil has participated
schoolid	ID of the school
Gender	(int) 0,1,2
Parented	ID of the parents
schoolterretoryid	ID of the territory the school belongs to
schoolterretoryname	Name of the school territory
parentterretoryid	ID of the territory parents belong to
parentterretoryname	Name of the parent territory

Teachers

column name	**description / notes**
Id	ID of a Teacher
Schooled	ID of the school a Teacher belongs to
Active	Boolean value (1, 0) describing if a teacher is active,

	i.e. if he is teaching