Multimodal Affect Recognition for Adaptive Intelligent Tutoring Systems

Ruth Janning
Information Systems and Machine Learning Lab (ISMLL)
University of Hildesheim
Marienburger Platz 22, 31141 Hildesheim, Germany
janning@ismll.uni-hildesheim.de

Carlotta Schatten
Information Systems and Machine Learning Lab (ISMLL)
University of Hildesheim
Marienburger Platz 22, 31141 Hildesheim, Germany
schatten@ismll.uni-hildesheim.de

Lars Schmidt-Thieme
Information Systems and Machine Learning Lab (ISMLL)
University of Hildesheim
Marienburger Platz 22, 31141 Hildesheim, Germany
schmidt-thieme@ismll.uni-hildesheim.de

ABSTRACT

The performance prediction and task sequencing in traditional adaptive intelligent tutoring systems needs information gained from expert and domain knowledge. In a former work a new efficient task sequencer based on a performance prediction system was presented, which only needs former performance information but not the expensive expert and domain knowledge. In this paper we aim to support this approach by automatically gained multimodal input like for instance speech input from the students. Our proposed approach extracts features from this multimodal input and applies to that features an automatic affect recognition method. The recognised affects shall finally be used to support the mentioned task sequencer and its performance prediction system. Consequently, in this paper we (1) propose a new approach for supporting task sequencing and performance prediction in adaptive intelligent tutoring systems by affect recognition applied to multimodal input, (2) present an analysis of appropriate features for affect recognition extracted from students speech input and show the suitability of the proposed features for affect recognition for adaptive intelligent tutoring systems, and (3) present a tool for data collection and labelling which helps to construct an appropriate data set for training the desired affect recognition approach.

Keywords

multimodal input, affect recognition, feature analysis, speech, adaptive intelligent tutoring systems

1. INTRODUCTION

Learning management systems like intelligent tutoring systems are an important tool for supporting the education of students for instance in learning fractional arithmetic. The main advantages of intelligent tutoring systems are the possibility for a student to practice any time, as well as the possibility of adaptivity and individualisation for a single student. An adaptive intelligent tutoring system possesses an internal model of the student and a task sequencer which decides which tasks in which order are shown to the student. Originally, the task sequencing in adaptive intelligent tutoring systems is done using information gained from expert and domain knowledge and logged information about the performance of students in former exercises. In [12] a new efficient sequencer based on a performance prediction system was presented, which only uses former performance information from the students to sequence the tasks and does not need the expensive expert and domain knowledge. This approach applies the machine learning method matrix factorization (see e.g. [1]) for performance prediction to former performance information. Subsequently, it uses the output of the performance prediction process to sequence the tasks according to the theory of Vygotsky's Zone of Proximal Development [14]. That is the sequencer chooses the next task in order to neither bore nor frustrate the student or in other words, the next task should not be too easy or too hard for the student.

In this paper we propose to support the task sequencer and performance prediction system of the approach in [12] in a new way by further automatically to get and process multimodal information. One part of this multimodal information, which is investigated in this paper, is the speech input from the students interacting with the intelligent tutoring system while solving tasks. A further part will be the typed input or mouse click input from the students, which will be reported in upcoming works. The approach proposed in this paper extracts features from the mentioned multimodal information and applies to that features an automatic affect recognition method. The output of the affect recognition method indicates, if the last task was too easy, too hard or appropriate for the student. This information matches the theory of Vygotsky's Zone of Proximal Development, hence it is obviously suitable for supporting the performance prediction system and task sequencer of the approach in [12]. However, for the proposed approach we need a large amount of labelled data. For this reason we developed a tutoring tool which (a) records students speech input as well as typed input and mouse click input and (b) allows the students to label by themselves how difficult they perceived the shown tasks. This tool is presented in the second part of this paper and will be used to conduct further studies to gain the desired labelled data.

The main contributions of this paper are: (1) presentation of a new approach for supporting performance prediction and task sequencing in adaptive intelligent tutoring systems by affect recognition on multimodal input, (2) identification and analysis of appropriate and statistically significant features for the presented approach, and (3) presentation of a new tutoring tool for multimodal data collection and self-labelling to gain automatically labelled data for training appropriate affect recognition methods.

In the following, first we will present some preliminary considerations along with state-of-the-art in section 2. Subsequently, we will describe in section 3 the real data set used for the feature analysis and investigate in section 4 for the data set the correlation between students affects and their performance. In section 5 we will propose and analyse appropriate features for affect recognition and in section 6 we will explain how to support performance prediction and task sequencing in intelligent tutoring systems by affect recognition applied to multimodal input. Before we conclude, we will describe in section 7 the mentioned tool for multimodal data collection and self-labelling.

2. PREPARATION AND RELATED WORK

Before an automatic affect recognition approach can be applied, one has to clarify three things: (1) What kind of features shall be used, (2) what kind of classes shall be used and (3) which instances shall be mapped to features and labelled with the class labels. After deciding which features, classes and instances shall be considered, one can apply affect recognition methods to these input data. In the following subsections we will present possible features, classes, instances and methods for affect recognition supporting performance prediction and task sequencing in adaptive intelligent tutoring systems along with the state-of-the-art.

2.1 Features

The first step before applying automatic affect recognition is to identify useful features for this process. For the purpose to recognise affect in speech one can use two different kinds of features ([13]): acoustic and linguistic features. Further, one can distinct linguistics (like n-grams and bag-of-words) and disfluencies (like pauses). If linguistics features are used, a transcription or speech recognition process has to be applied to the speech input before affect recognition can be conducted. Subsequently, approaches from the field of sentiment classification or opinion mining (see e.g. [10]) can be applied to the output of this process. However, the methods of this field have to be adjusted to be applicable to speech instead of written statements.

Another possibility for speech features is to use disfluencies features like it was done in [17], [7] and [4] for expert identification. The advantage of using such features is that instead of a full transcription or speech recognition approach only for instance a pause identification has to be applied before. That means that one does not inherit the error of the full speech recognition approach. Furthermore, these features are independent from the need that students use words related to affects. For using this kind of features one has to investigate, which particular features are suitable for the special task of affect classification in adaptive intelligent tutoring systems. Because of the mentioned advantage of disfluencies features in this work we focus on features extracted from information about speech pauses as one part of the multimodal input for affect recognition.

As mentioned in the introduction the other part of the multimodal input will be features which are gained from information about typed input or mouse click input from the students. This kind of features is similar to the keystroke dynamics features used in [2]. In [2] emotional states were identified by analysing the rhythm of the typing patterns of persons on a keyboard.

2.2 Classes

The second step before applying automatic affect recognition is to define the classes corresponding to emotions and affective states, which shall be recognised by the used affect recognition approach. According to [6], [5] and [16] it is possible to recognise in intelligent tutoring systems students affects like for instance confusion, frustration, boredom and flow. As mentioned above, we want to use the students behaviour information gained from speech and from typed input or mouse click input for supporting the performance prediction system and task sequencer of the approach in [12], which is based on the theory of Vygotsky's Zone of Proximal Development [14]. That means that the goal is to neither bore the student with too easy tasks nor to frustrate him with too hard tasks, but to keep him in the Zone of Proximal Development. Accordingly, we want to use the output of the automatic affect recognition to get an answer to the question "Was this task too easy, too hard or appropriate for the student?", or with other words we want to find out if the student felt under-challenged, over-challenged or like to be in a flow. However, the mapping between confusion, frustration, boredom and under-challenged, over-challenged is not unambiguous as one can infer e.g. from the studies mentioned in [16]. Hence, we will use instead of the above mentioned affect classes three other classes for supporting performance prediction and task sequencing by automatic affect recognition: under-challenged, over-challenged and flow. One could summarise these classes as *perceived task-difficulty classes*, as we aim to recognise the individual perceived task-difficulty from the view of the student.

2.3 Instances

The third step before applying automatic affect recognition is deciding which instances shall be mapped to features and labelled with the class labels. If the goal of the affect recognition is to provide a student motivation or hints according to his affective state like e.g. in [16], then instances can be utterances. For supporting performance prediction and task sequencing by affect recognition instead one needs at the end of a task the information, if the task overall was too easy, too hard or appropriate for the student. The reason is that this information shall help to choose the next task shown to the student. Hence, an instance for supporting performance prediction and task sequencing by affect recognition has to be instead of an utterance the whole speech input of a student for one task.

2.4 Methods

The possible methods for an automatic affect recognition depend on the kind of the features used as input. As mentioned above, for speech we distinct two kinds of features: linguistics features and disfluencies. Linguistics features are gained by a preceding speech recognition process and can be processed by methods coming from the areas sentiment analysis and opinion mining ([10]). Especially methods from the field of opinion mining on microposts seem to be appropriate if linguistics features are considered. State-of-the-art approaches in opinion mining on microposts use methods for instance based on optimisation approaches ([3]) or Naive Bayes ([11]).

The process of gaining disfluencies like pauses is different to the full speech recognition process. For extracting for instance pauses usually an energy threshold on the decibel scale is used as in [4] or an SVM is applied for pause classification on acoustic features as in [9]. Appropriate state-of-the-art methods for automatic emotion and affect recognition on disfluencies features as well as on features from information about typed input or mouse click input are – as proposed e.g. in [13] and [6] – classification methods like artificial neural networks, SVM, decision trees or ensembles of those.

3. REAL DATA SET

After identifying features, classes, instances and methods for affect recognition for supporting performance prediction and task sequencing like above one can collect data for a concrete feature analysis and a training of the chosen affect classification method. We conducted a study in which the speech and actions of ten 10 to 12 years old German students were recorded and students affective states as well as the perceived task-difficulties were reported. The labelling of these data was done on the one hand concurrently by the tutor and on the other hand retrospectively by a second reviewer. Furthermore, a labelling per exercise (consisting of several subtasks) and an overall labelling per student as an aggregation of the labels per exercise was done. During the study a paper sheet with fraction tasks was shown to the students and they were asked to paint (with the software Paint) and explain their observations and answers. We made a screen recording to record the painting of the students and an acoustic recording to record the speech of the students. The screen recordings were used for the retrospective annotation. The speech recordings shall be used to gain the input for affect recognition. The mentioned typed input or mouse click input information we will collect and investigate in further studies with the self-labelling and multimodal data collection tutoring tool described in section 7.1. In this paper we focus on speech features and hence in section 5 we will propose and analyse possible features extracted from speech pauses. But first we will investigate in the following section 4 the correlation between perceived task-difficulty labels and the performance of the students in the real data set.

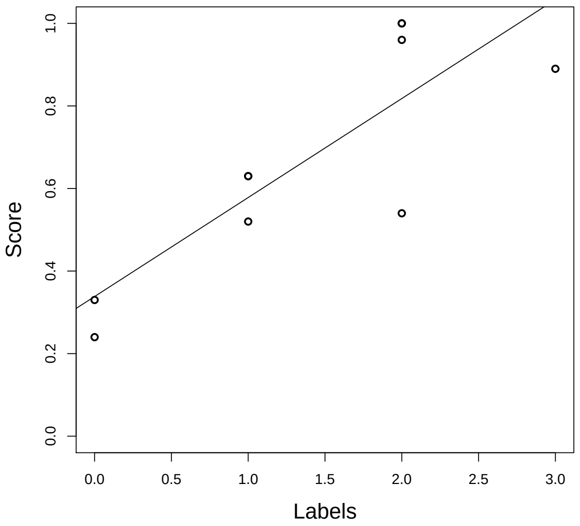


Figure 1: Mapping of the perceived task-difficulty labels to the scores of the students in the real data set.

4. CORRELATION OF PERCEIVED TASK-DIFFICULTY LABELS AND SCORE

Before we present speech features for recognising perceived task-difficulty, we want to show that there is a correlation between the proposed perceived task-difficulty labels and the performance of the students, to underline the suitability of supporting performance prediction and task sequencing by the proposed affect recognition approach. Hence, we mapped the overall perceived task-difficulty labels to the overall score of the students (see figure 1). For this mapping we encoded the different overall perceived task-difficulty class labels as follows:

- 0 = over-challenged
- 1 = over-challenged/flow
- 2 = flow
- 3 = flow/under-challenged
- 4 = under-challenged

The overall score of a student i is computed by

$$\frac{n_{c_i}}{n_{t_i}} \, , \qquad (1)$$

where n_{c_i} is the number of correctly solved tasks of student i and n_{t_i} is the number of tasks shown to student i. In figure 1 one can see that there is a clear correlation between perceived task-difficulty labels and score. To substantiate this observation we applied a statistical test by conducting a linear regression and measuring the p-value, indicating the statistical significance, as well as the R^2 and Adjusted R^2 value, indicating how well the regression line can approximate the real data points. This approach delivers a p-value of 0.0027,

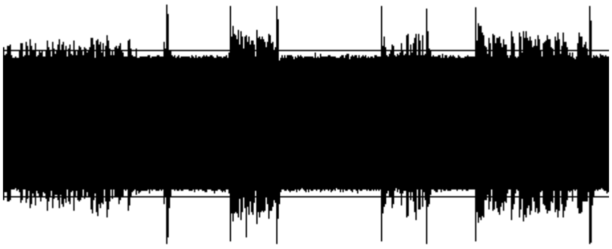


Figure 2: Graphic of the decibel scale of an example sound file of a student. The two straight horizontal lines indicate the threshold.

a R^2 value of 0.6966, and an Adjusted R^2 value of 0.6586. The small p-value indicates a strong statistical significance. The significant correlation between perceived task-difficulty labels and scores, which demonstrate the performance, indicates that it makes sense to support performance prediction and task sequencing by perceived task-difficulty classification.

5. SPEECH FEATURE ANALYSIS

The features we propose and analyse in this section are gained from speech pauses. Hence, first one has to identify pauses within the speech input data. The most easy way is to define a threshold on the decibel scale as done e.g. in [4]. For our preliminary study of the data we also used such a threshold, which we adjusted by hand. More explicitly, we extracted the amplitudes of the sound files and computed the decibel values. Subsequently, we investigated which decibel values belong to speech and which ones to pauses (see figure 2). In larger data and in the application phase later on, one has to learn automatically the distinction between speech and pauses by either learn the threshold or train an SVM, which classifies speech and pauses.

5.1 Single Feature Analysis

Before we can introduce the features we want to investigate, we have to define some measurements:

- m: number of students
- p_i: total length of pauses of student i
- s_i: total length of speech of student i
- n_{p_i}: number of pause segments of student i
- n_{s_i}: number of speech segments of student i
- $p_i^{(x)}$: xth pause segment of student i
- $s_i^{(y)}$: yth speech segment of student i
- n_{t_i}: number of tasks shown to student i
- n_{c_i}: number of correctly solved tasks by student i
- Overall score for student i: $\frac{n_{c_i}}{n_{t_i}}$

Table 1: p-value, R^2 and Adjusted R^2 for the feature *Length of maximal pause segment* mapped to score as well as to label.

Mapped to	p-value	R^2	Adjusted R^2
Score	0.1156	0.2802	0.1902
Label	0.0678	0.3577	0.2774

Our data set exists of acoustic recordings from m students, each of which saw n_{t_i} tasks and solved n_{c_i} tasks correctly. The overall score of a student i in this case is the number of correctly solved tasks n_{c_i} divided by the number of seen tasks n_{t_i}. After applying the above mentioned threshold to the data, we get for each student i the total length of pauses p_i and the total length of speech s_i in his acoustic recoding. Furthermore, we can count connected pause and speech segments to get the number of pause segments n_{p_i} and speech segments n_{s_i} of a student i. The xth pause segment is then $p_i^{(x)}$ and the yth speech segment $s_i^{(y)}$. By means of these measurements and their combination we can create a set of features useful for affect recognition supporting performance prediction and task sequencing:

- Ratio between pauses and speech ($\frac{p_i}{s_i}$)
- Frequency of speech pause changes ($\frac{n_{p_i}+n_{s_i}}{\max_j(n_{p_j}+n_{s_j})}$)
- Percentage of pauses of input speech data ($\frac{p_i}{(p_i+s_i)}$)
- Length of maximal pause segment ($\max_x(p_i^{(x)})$)
- Length of average pause segment ($\frac{\sum_x p_i^{(x)}}{n_{p_i}}$)
- Length of maximal speech segment ($\max_y(s_i^{(y)})$)
- Length of average speech segment ($\frac{\sum_y s_i^{(y)}}{n_{s_i}}$)
- Average number of seconds needed per task ($\frac{(p_i+s_i)}{n_{t_i}}$)

The ratio between the total length of pauses and the total length of speech indicates, if one one them is notable larger than the other one, i.e. if the student made much more speech pauses than speaking or vice versa. The frequency of speech and pause segment changes indicates, if there are many short speech and pauses segments or just a few large ones and it is normalised by dividing it by the maximal sum of pause and speech segments over all students. From the percentage of pauses one can see if the total pause length was much larger than the total speech part, i.e. the student did not speak much but was more thinking silently. The length of maximal pause or speech segment indicates if there was e.g. a very long pause segment where the student was thinking silently or a very long speech segment where the student was in a speech flow. The length of average pause or speech segment give us an idea of how much on average the student was in a silent thinking phase or a speech flow. The average number of seconds needed per task indicates how long a student on average needed for solving a task.

To investigate, if these features are suitable to describe perceived task-difficulty as well as performance in our real data

Table 2: p-value, R^2 and Adjusted R^2 for the best combinations of features (with a p-value smaller than 0.05) of a set with 6, 5, 4 or 3 features mapped to the score.

#	Features	p-val.	R^2	Adj. R^2
6	Frequency of changes, seconds per task, max. length of pause, average length of pause, max. length of speech average length of speech	0.0439	0.9516	0.8548
5	Frequency of changes, seconds per task, max. length of pause, average length of pause, average length of speech	0.0105	0.9496	0.8867
4	Frequency of changes, seconds per task, average length of pause, average length of speech	0.0415	0.8207	0.6773
3	Frequency of changes, frequency of changes, average length of speech	0.0431	0.719	0.5786

set, we mapped the values of each feature to the score as well as to the perceived task-difficulty labels. Subsequently, we applied a linear regression to measure the p-value as well as the R^2 and Adjusted R^2 value. However, as expected, single features are not very significant. The feature with the best values for p-value, R^2 and Adjusted R^2 – mapped to score as well as to labels – is the *Length of maximal pause segment*. The statistical values for this feature are shown in table 1. These values are not very satisfactory, as one would desire a p-value smaller than 0.05 and values for R^2 and Adjusted R^2 which are closer to 1. A more reasonable approach is to combine several features instead of considering just one feature. Hence, in the following section we will investigate different combinations of features.

5.2 Feature Combination Analysis

We analysed different combinations of features by applying a multivariate linear regression to them to gain the p-value, R^2 and Adjusted R^2 for these combinations. The investigated combinations are combinations where all features are not strongly correlated, i.e. whenever we had two correlated features we put just one of them into the feature set for that combination. In further steps we removed from the considered feature sets feature by feature. Furthermore, in the multivariate linear regression we mapped the features on the one hand to the score and on the other hand to the labels. The results of the best combinations, i.e. such with a p-value at least smaller than 0.05, are shown in table 2 and 3. For the score there were no combinations with only 2 features with a p-value smaller than 0.05, hence in table 2 we just listed the best combinations with 3 up to 6 features. For the labels instead there were no such combinations, which have a p-value smaller than 0.05, with 6 features, so that in table 3 we only listed the best combinations of 2 up to 5 features. For both (score and labels) there are statistically significant feature combinations. That means that our proposed features are able to describe the score as well as the labels.

Table 3: p-value, R^2 and Adjusted R^2 for the best combinations of features (with a p-value smaller than 0.05) of a set with 5, 4, 3 or 2 features mapped to the labels.

#	Features	p-val.	R^2	Adj. R^2
5	Ratio pause speech, frequency of changes, seconds per task, average length of pause, average length of speech	0.0284	0.9158	0.8106
4	Ratio pause speech, frequency of changes, average length of pause, average length of speech	0.0154	0.8818	0.7872
3	Ratio pause speech, frequency of changes, average length of speech	0.0117	0.8207	0.7311
2	Frequency of changes, average length of speech	0.0327	0.6238	0.5163

6. SUPPORTING PERFORMANCE PREDICTION AND SEQUENCING

As mentioned in the introduction, our goal is to support the performance prediction system and task sequencer of the approach in [12] by affect recognition, or by multimodal input respectively. Hence, in the following we will propose how to realise this support. In figure 3 a block diagram of the approach of supporting performance prediction and task sequencing by means of affect recognition is presented. The approach in [12] is represented in figure 3 by the non-dotted arrows: the performance prediction gets input from former performances and computes by means of the machine learning method matrix factorization predictions for future performances, which are the input for the task sequencer. The task sequencer decides based on the theory of Vygotsky's Zone of Proximal Development from the performance prediction input which task shall be shown next to the student. This process can be supported by the multimodal input as follows:

(1) The additional input for the performance predictor can be the output of the affect recognition, i.e. the perceived task-difficulty labels. In this case the performance predictor can take the perceived task-difficulty of the last task ($T^{(t)}$) to use the following rules for deciding how difficult the next task ($T^{(t+1)}$) should be:

   - If $T^{(t)}$ was too easy (label *under-challenged* or *flow/under-challenged*), then $T^{(t+1)}$ should be harder.
   - If $T^{(t)}$ was appropriate (label *flow*), then $T^{(t+1)}$ should be similar difficult.
   - If $T^{(t)}$ was too hard (label *over-challenged* or *over-challenged/flow*), then $T^{(t+1)}$ should be easier.

(2) The values of the features gained by feature extraction from speech, typed input and mouse click input

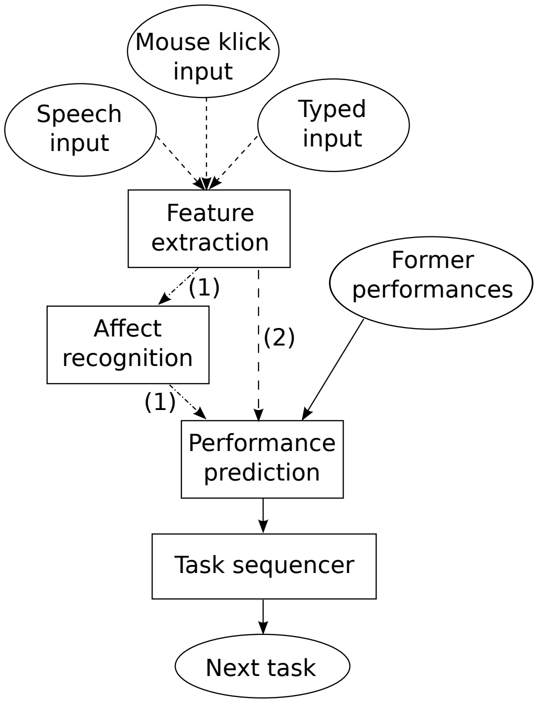


Figure 3: Approach for supporting performance prediction and task sequencing by means of multimodal input and affect recognition.

can be fed directly into the performance prediction without applying an affect recognition. That means that the features are mapped to scores instead of perceived task-difficulty classes. That this makes sense was shown in section 4 and 5. The performance predictor can then compare e.g. the differences between performances, expressed as *score*, and the scores computed by means of the features ($\widehat{score}$). This difference indicates outliers like if a student felt to be in a flow or under-challenged but his score is worse, i.e. $\widehat{score} > score$. In this case the student may not fully understand the principles of the considered task although he thinks so. Hence, next the system should show the student rather tasks which explain the approach of solving such kind of tasks.

In our studies we observed the behaviour of students described in (2), i.e. the student was labelled as to be in a flow or under-challenged, although he performed worse, as he just thought to understand how the tasks should be solved but he was wrong. In figure 4 this behaviour is indicated by the outliers.

7. LABELLING AND DATA COLLECTION

As mentioned in section 3 the labels of our real data set come from two sources: (a) a concurrent annotation by the tutor and (b) a retrospective annotation by another external reviewer on the basis of the tasks sheet, the sound files and the screen recording. However, in the literature one can find further labelling strategies like self-labelling of the students (see e.g. [5], [6], [8]). The advantage of self-labelling is that one can gain automatically a labelled data set for a subsequent training of an affect recognition method. Furthermore, as we want to recognise the perceived task-difficulty from the view of the student, a label from the student himself seem to be more appropriate than labels from another person only reviewing the behaviour of the student. Hence, for further studies we developed a tool for collecting speech data and typed input and mouse click input data, labelled automatically with the task-difficulty perceived by the student. This tool will be further described in the following section.

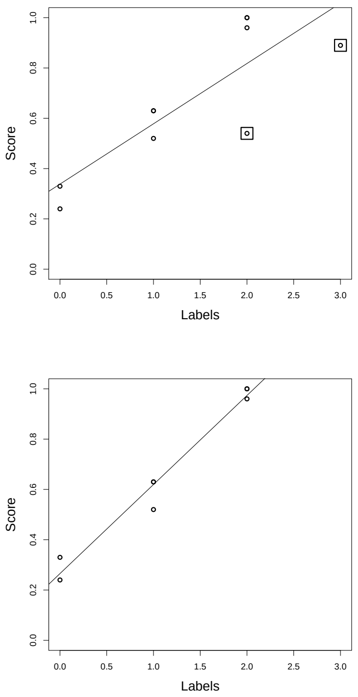


Figure 4: Mapping of the perceived task-difficulty labels to the scores of the students in the real data set (a) with outliers indicated by surrounding rectangles (top) and (b) without outliers (bottom).

7.1 Self-Labelling Fractional Arithmetic Tutor for Multimodal Data Collection

To be able to conduct studies in which the students themselves label the task-difficulty which they perceived, we developed a tutoring tool (*self* - **se**lf-**l**abelling **f**ractional arithmetic tutor for multimodal data collection) written in Java. However, for little children it might be difficult to analyse themselves (see e.g. [8]). Hence, self-labelling is often applied in experiments with at least college students as for instance in [5]. Therefore, we will conduct the experiments with this tool first with older students and more challenging tasks. Later on we will investigate if there is a way to adapt the tool so that a self-labelling is possible also with younger students. Nevertheless, conducting experiments with older students has several advantages besides the possibility of a reasonable self-labelling: older students are able to focus on the tasks longer than young students and the privacy issues are not such strong as for younger students. Both facts lead to more data. Hence, besides investigating the possibility of adapting *self* for younger students, we have to identify differences and similarities of the data from older and younger students to find out how to exploit older students data to recognise affects from multimodal input from younger students.

In figure 5 one can see the graphical user interface of our self-labelling multimodal data collection tool *self*. To gain more background information, in the beginning *self* asks some information from the students as course of studies, number of terms, age and gender. Subsequently, an instruction with hints how to behave is shown to the students, which they can have a look at also while interacting with the tool (button ”Anleitung“). *self* speaks to the students to motivate them to speak with the system and records the speech input of the students. The speech output of *self* is generated by means of *text to speech* realised by the library MARY developed at the DFKI ([18]). While interacting with the system, the student can type in numbers, ask for a hint (button ”Hilfe”), skip the task because it is too easy or because it is too hard (left buttons) or submit the solution (button ”Endergebnis überprüfen”). Every action of the student, like asking for a hint or submitting the answer, is written – together with a time stamp – into a log file immediately after the action, enabling also the extraction of typed input or mouse click input features. Also a score depending on the number of requested hints h_r and the number of incorrect inputs w is computed according to the approach in [15] and written into the log file. The formula for this score is

$$1 - (\frac{h_r}{h_t} + (w \cdot 0.1)) \ , \qquad (2)$$

where h_t is the total number of available hints for the considered task. The meaning behind the formula is that each wrong input $w^{(j)}$ is punished with a factor of 0.1 and every request of a hint $h_r^{(k)}$ is punished with a factor of $\frac{1}{h_t}$, so that if every hint was seen the score will be 0. After the student submitted the correct answer, he is asked to evaluate, if this task was too easy, too hard or appropriate for him (see pop-up window in figure 5). The tasks implemented in *self* for older students cover the following areas:

- Reducing fractions with numbers and variables
- Fraction addition with and without intermediate steps and with numbers and variables
- Fraction subtraction with and without intermediate steps and with numbers and variables
- Fraction multiplication with and without intermediate steps and with numbers and variables
- Fraction division with and without intermediate steps and with numbers and variables
- Distributivity law with and without intermediate steps
- Finite sums of unit fractions
- Rule of Three

After developing *self*, the next step will be to conduct further studies with students to collect an adequate amount of automatically labelled speech input, typed input and mouse click input data for training an affect recognition method and supporting performance prediction and task sequencing. Furthermore, we will investigate if there is a way to adapt *self* so that also younger students can label themselves.

8. CONCLUSIONS

We proposed a new approach for supporting performance prediction and task sequencing in adaptive intelligent tutoring systems by affect recognition on features gained from multimodal input like students speech input. For this approach we proposed and analysed appropriate speech features and showed that there are statistically significant feature combinations which are able to describe students affect, or perceived task-difficulty respectively, as well as the performance of a student. Furthermore, we proved the possibility of supporting performance prediction and task sequencing by perceived task-difficulties by demonstrating that there is a correlation between perceived task-difficulty and performance. Next steps will be to conduct more studies with students by means of the presented self-labelling and multimodal data collection tool to enable a training of an appropriate affect recognition method for supporting performance prediction and task sequencing in adaptive intelligent tutoring systems.

9. ACKNOWLEDGMENTS

The research leading to the results reported here has received funding from the European Union Seventh Framework Programme (FP7/2007 – 2013) under grant agreement No. 318051 – iTalk2Learn project (www.italk2learn.eu). Furthermore, we thank our project partner Ruhr University Bochum for realising the study and data collection as well as the IMAI of the University of Hildesheim for support for the tutoring tool and preparation for future studies.

10. REFERENCES

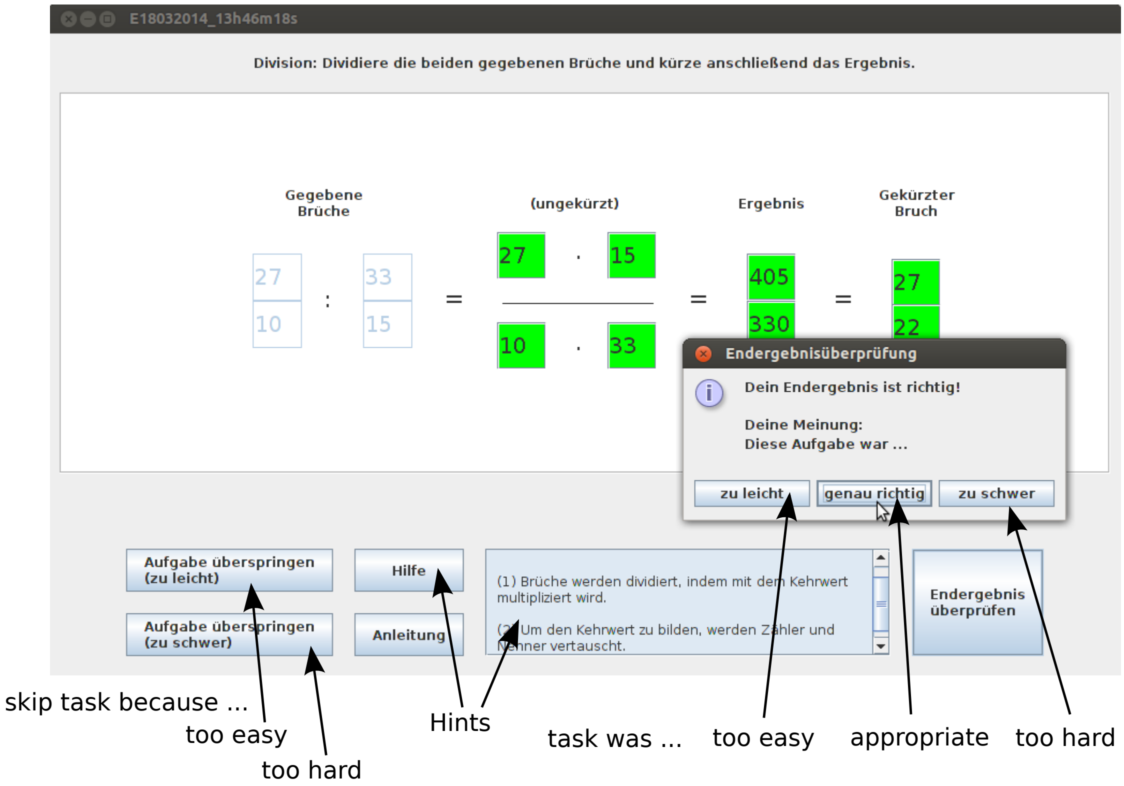
[1] Cichocki, A., Zdunek, R., Phan, A. H. and Amari, S.I. 2009. Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation, Wiley.

[2] Epp, C., Lippold, M., Mandryk, R.L. 2011. Identifying Emotional States Using Keystroke Dynamics. In Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems (CHI 2011), Vancouver, BC, Canada, pp. 715–724.

[3] Hu, X., Tang, L., Tang, J. and Liu, H. 2013. Exploiting Social Relations for Sentiment Analysis in Microblogging. In Proceedings of the Sixth ACM WSDM Conference (WSDM ’13).

[4] Luz, S. 2013. Automatic Identification of Experts and Performance Prediction in the Multimodal Math Data Corpus through Analysis of Speech Interaction. Second International Workshop on Multimodal Learning Analytics, Sydney Australia, December 2013.

[5] D’Mello, S., Picard, R. and Graesser, A. 2007. Towards An Affect-Sensitive AutoTutor. Intelligent Systems, IEEE, Vol. 22, Issue 4, pp. 53–61.



E18032014_13h46m18s

Division: Dividiere die beiden gegebenen Brüche und kürze anschließend das Ergebnis.

Endergebnisüberprüfung

Dein Endergebnis ist richtig!

Deine Meinung:
Diese Aufgabe war ...

(1) Brüche werden dividiert, indem mit dem Kehrwert multipliziert wird.

(2) Um den Kehrwert zu bilden, werden Zähler und Nenner vertauscht.

Figure 5: Graphical user interface of the developed fractional arithmetic tutoring tool *self* for self-labelling as well as for speech data and typed input or mouse click input data collection.

[6] D'Mello, S.K., Craig, S.D., Witherspoon, A., McDaniel, B. and Graesser, A. 2008. Automatic detection of learner's affect from conversational cues. User Model User-Adap Inter, DOI 10.1007/s11257-007-9037-6.

[7] Morency, L.P., Oviatt, S., Scherer, S., Weibel, N. and Worsley, M. 2013. ICMI 2013 grand challenge workshop on multimodal learning analytics. In Proceedings of the 15th ACM on International conference on multimodal interaction (ICMI 2013), pp. 373–378.

[8] Porayska-Pomsta, K., Mavrikis, M., D'Mello, S., Conati, C. and Baker, R.S.J.d. 2013. Knowledge Elicitation Methods for Affect Modelling in Education. International Journal of Artificial Intelligence in Education, ISSN 1560-4292.

[9] Qi, F., Bao, C., Liu, Y. 2004. A novel two-step SVM classifier for voiced/unvoiced/silence classification of speech. International Symposium on Chinese Spoken Language Processing, pp. 77 – 80.

[10] Sadegh, M., Ibrahim, R., Othman, Z.A. 2012. Opinion Mining and Sentiment Analysis: A Survey. International Journal of Computers & Technology, Vol. 2, No. 3.

[11] Saif, H., He, Y. and Alani, H. 2012. Semantic Sentiment Analysis of Twitter. In Proceedings of the 11th International Semantic Web Conference (ISWC 2012).

[12] Schatten, C. and Schmidt-Thieme, L. 2014. Adaptive Content Sequencing without Domain Information. In Proceedings of the Conference on computer supported education (CSEDU 2014).

[13] Schuller, B., Batliner, A., Steidl, S. and Seppi, D. 2011. Recognising realistic emotions and affect in speech: State of the art and lessons learnt from the first challenge. Speech Communication, Elsevier.

[14] Vygotsky, L.L.S. 1978. Mind in society: The development of higher psychological processes. Harvard university press.

[15] Wang, Y. and Heffernan, N. 2011. Extending Knowledge Tracing to allow Partial Credit: Using Continuous versus Binary Nodes. Artificial Intelligence in Education, Lecture Notes in Computer Science, Vol. 7926, pp. 181–188.

[16] Woolf, B., Burleson, W., Arroyo, I., Dragon, T., Cooper, D. and Picard, R. 2009. Affect-aware tutors: recognising and responding to student affect. Int. J. of Learning Technology, Vol. 4, No. 3/4, pp. 129–164.

[17] Worsley, M. and Blikstein, P. 2011. What's an Expert? Using Learning Analytics to Identify Emergent Markers of Expertise through Automated Speech, Sentiment and Sketch Analysis. In Proceedings of the 4th International Conference on Educational Data Mining (EDM '11), pp. 235–240.

[18] The MARY Text-to-Speech System, http://mary.dfki.de/