Formative Feedback in an Interactive Spoken CALL System

Nikos Tsourakis, Manny Rayner, Claudia Baur

TIM/FTI, University of Geneva, Switzerland

{Nikolaos.Tsourakis, Emmanuel.Rayner, Claudia.Baur}@unige.ch

Abstract

By definition spoken dialogue CALL systems should be easy to use and understand. However, interaction in this context is often far from unhindered. In this paper we introduce a formative feedback mechanism in our CALL system, which can monitor interaction, report errors and provide advice and suggestions to users. The distinctive feature of this mechanism is the ability to combine information from different sources and decide on the most pertinent feedback, which can also be adapted in terms of phrasing, style and language. We conducted experiments at three secondary schools in German-speaking Switzerland and the obtained results suggest that our feedback mechanism helps students during interaction and contributes as a motivating factor.

Index Terms: formative feedback, CALL, children, German

1. Introduction

Voice user interfaces have become extremely popular in the education of children. What is easier to use than voice, the most natural means of intuitive interaction? Especially for children it can contribute to a fun, exciting and engaging way to learn [1]. Besides the known problems of recognizing children’s speech [2], providing feedback in terms of pronunciation and grammar errors [3], reading fluency [4], speech scoring [5], etc., is becoming a central issue.

As stated in [6] “formative feedback is defined as information communicated to the learner that is intended to modify his or her thinking or behavior for the purpose of improving learning”. Numerous research articles have appeared in the literature focusing on how feedback affects knowledge and skill acquisition. There are, however, contradictory finding on whether it is helpful or not [6], making the application of formative feedback in a CALL application a difficult task.

On the other hand, interaction in this context involves a software system, which imposes several technical challenges per se. Software installation, microphone configuration, learning how to use the interface, etc., are typical impediments. The interaction can become even more problematic if the student is overwhelmed by the functionalities offered by the system. A typical example is the latest trend in CALL applications to incorporate gamification techniques in order to increase student motivation [7], [8]. Gamified applications are based on elements like scores, badges and leaderboards [9], but the rules that govern interaction may not always be transparent to end-users.

For the reasons stated above we decided to implement a feedback module and embed it in CALL-SLT, our computer assisted language learning system [10]. The CALL-SLT tool offers students of English the opportunity to train their receptive and productive language skills in various conversational dialogue scenarios. The interactive element of this approach allows the students not only to train their grammar and vocabulary knowledge, but more importantly real-life conversational skills.

The current interface already provides feedback based on user interaction, e.g. reporting failed recognition, informing on total score, etc. The question that immediately arises is why we need to introduce an extra feedback mechanism. After several user studies with CALL-SLT [10], [11], [12], we identified the following obstacles when using the system:

1. Users may be overwhelmed by too much feedback, leading to information overload.
2. Users may not be aware of certain system functionalities.
3. Users may not understand what triggered certain events.
4. It is difficult to unify the feedback messages in terms of phrasing, style and language.

As a remedy to the aforementioned problems, we introduced a parallel feedback mechanism to CALL-SLT. The output on the screen remained the same, except for an extra message coming from the newly introduced module. By utilizing information from three different sources: (a) the client application, (b) the server side component and (c) the waveform, the module tries to produce the most pertinent feedback after each interaction. The decision is based on a set of priority rules defined externally, which makes their maintenance an easy task even by non-experts. The client interface receives the message and is responsible for presenting it in predefined locations on the screen according to the nature of the event that triggered it. We conducted experiments at three secondary schools in German-speaking Switzerland using both versions of the system, with and without formative feedback. Our findings suggest that the feedback module helps and motivates students during their interaction with the system.

The rest of the paper is organized as follows. Section 2 gives a brief summary of the technical aspects of the system, and Section 3 describes the feedback module. Section 4 presents the results of our user studies and the final section concludes.

2. CALL-SLT System

2.1. Basic architecture

CALL-SLT is a prompt-response system based on speech recognition and machine translation technology, in which the system and the user take alternate turns; CALL applications of this general kind can be traced back to the “spoken translation game” described in [13].

Each exchange begins with the system giving the student a prompt, which in the present version is a combination of an English-language multimedia file and a piece of German text; for example, at the beginning of the Hotel lesson, the system plays a cartoon clip of a desk clerk asking “How many nights
would you like to stay at our hotel?”), simultaneously showing the German text “Frag: Zimmer für 6 Nächte” (“ask: room for 6 nights”). The student then gives a spoken response; the intent is that this should be reasonably free, so here one can for instance answer “I would like to stay for six nights”, “I want a room for six nights”, “A room for six nights please”, and several other variants.

The system decides whether to accept or reject the response by first performing speech recognition, then translating to a language-neutral (interlingual) representation, and finally matching this representation against the language-neutral representation of the prompt. A “help” button allows the student, at any time, to access a correct sentence in both written and spoken form; spoken help examples are collected from previous successful interactions. When the system has decided whether to accept or reject, it moves to a new dialogue state; the choice of state is determined by an XML-based script written by the course designer, which specifies various options. Continuing the example, an “accept” moves to a state where the desk clerk’s next question is “What type of room would you like?”; a “reject” stays in the same state, with the desk clerk saying that he did not understand; and a sequence of three rejects moves to a state where the clerk says he did not understand, but asks whether a room for one night will be okay. Figure 1 shows the user interface.

The system is deployed over the web using a scalable architecture designed for cloud-based computing. In common with similar platforms, like WAMI [14] and Nuance’s Mobile Developer Platform, it uses a client/server approach in which speech recognition is carried out on the server side. The overhead due to web deployment is a few hundred milliseconds per recognition operation, compared to execution on a desktop machine. Full details are presented in [15]. Speech and language processing use a grammar-based framework embodied in the Regulus platform [16], which in turn sits on top of the commercial Nuance Toolkit package.

### 2.2. Content

The course content was formulated in close collaboration with an English teacher working at a secondary school in German-speaking Switzerland, who provided the subject matter expertise required to guarantee the usefulness of the content for the target audience. On the basis of this collaboration, we decided to link the content to the school’s curriculum and the standard English textbook *Ready For English I* used in Swiss German schools [17], as well as to use a communicative approach to second language learning, including multimedia elements to simulate a conversation partner for the language learner.

This resulted in a dialogue-based system, loosely covering the first year’s class content. We created the eight coherent lessons or dialogues, which can be put together to simulate a virtual trip to London. This approach allows the students to practice their receptive skills (by listening to the videos recorded by native English speakers), as well as their productive skills (by engaging in the conversation and responding to the questions), two closely linked components of second language acquisition according to the CEFR [18]. Table 1 shows examples of typical content.

Table 1. Lessons and typical content.

<table>
<thead>
<tr>
<th>Train station:</th>
<th>name, nationality, numbers, locations, time expressions</th>
</tr>
</thead>
<tbody>
<tr>
<td>I would like two tickets to London - I need to leave on Tuesday afternoon</td>
<td></td>
</tr>
<tr>
<td>Introducing yourself:</td>
<td>name, nationality, siblings, capitals</td>
</tr>
<tr>
<td>I am from Italy - I have two sisters</td>
<td></td>
</tr>
<tr>
<td>Tube station:</td>
<td>numbers, locations, prices</td>
</tr>
<tr>
<td>Can I pay by credit card? - Thank you</td>
<td></td>
</tr>
<tr>
<td>Hotel:</td>
<td>numbers, room types, prices, payment types, where-questions</td>
</tr>
<tr>
<td>I would like to stay for two nights - Is there a swimming pool?</td>
<td></td>
</tr>
<tr>
<td>Tourist Information Office:</td>
<td>numbers, cultural knowledge of London, time expressions, ordinal numbers</td>
</tr>
<tr>
<td>I would like tickets for Wicked - I want to sit in the third row</td>
<td></td>
</tr>
<tr>
<td>Restaurant:</td>
<td>food and beverages, payment types</td>
</tr>
<tr>
<td>I want my steak well done - I did not order this</td>
<td></td>
</tr>
<tr>
<td>Asking and giving directions:</td>
<td>where-questions, directions, distances</td>
</tr>
<tr>
<td>Where is the Science Museum? - Which bus goes there?</td>
<td></td>
</tr>
<tr>
<td>Shopping:</td>
<td>clothing, colours, numbers, like/dislike expressions</td>
</tr>
<tr>
<td>I am a small - This is too expensive</td>
<td></td>
</tr>
</tbody>
</table>

The eight lessons use a combined vocabulary of about 450 words. A dialogue typically contains between 10 and 20 states (average of 14.4); in each state, there are typically between 5 and 15 possible prompts (average of 10.8) that can be issued. Random choices are made both for prompt and state transitions, so students get a different interaction each time they do a lesson. The dialogue flow is structured so that students are given at most three attempts at each prompt, after which the system backs off to a yes/no question: the intention is to make sure that the student cannot get stuck at a difficult step.

### 3. Feedback module

#### 3.1. Basic architecture

For the reasons stated in the introduction of the paper we decided to extend the current system with a feedback module. This extra component monitors interaction by utilizing three sources of knowledge. Specifically: the client application, the server-side dialogue manager and the user waveform (Figure 3). An expert decision is performed after analyzing the input from the three sources in order to provide the most pertinent feedback as output.
The average overhead due to this processing is 914 msec (sd = 221 msec) and in the subsequent sections we will highlight the functionality offered by each one of the three sub-components.

3.2. Dialogue Manager & Client application

The dialogue management framework implements a version of Update Semantics [19]. The central concepts are those of dialogue move, information state and dialogue action. At the beginning of each turn, the dialogue manager is conceptually in an information state that may include the current state of the application, the history of the dialogue, which participant is considered to hold the dialogue initiative, and so on. Inputs to the dialogue manager are by definition dialogue moves, and outputs are dialogue actions. Thus if a dialogue move is applied in a given information state, the result is a new information state and a dialogue action.

The client application, on the other hand, is a lightweight process developed using Flash 11 in combination with ActionScript 3.0. When the user types the application’s link in the browser, the remote web server delivers the requested page that includes the ”.swf” file of our flash client. The latter takes control and communicates directly using Remote Procedures Calls (RPC) with the remote Flash Media Server, which is the entry point of any client’s request.

Table 2. Information from the two sources

<table>
<thead>
<tr>
<th>Source</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>DM</td>
<td>Points gained/lost</td>
</tr>
<tr>
<td>DM</td>
<td>Total score</td>
</tr>
<tr>
<td>DM</td>
<td>Number of correct/wrong responses</td>
</tr>
<tr>
<td>DM</td>
<td>Number of consecutive correct/wrong responses</td>
</tr>
<tr>
<td>DM</td>
<td>Score distance to get current badge</td>
</tr>
<tr>
<td>DM</td>
<td>Bonus phrase acquired</td>
</tr>
<tr>
<td>DM</td>
<td>Badge already acquired</td>
</tr>
<tr>
<td>CA</td>
<td>Video prompt was interrupted</td>
</tr>
<tr>
<td>CA</td>
<td>The video prompt was played back again</td>
</tr>
<tr>
<td>CA</td>
<td>Help requested in the wrong context</td>
</tr>
<tr>
<td>CA</td>
<td>Access to the microphone was denied</td>
</tr>
</tbody>
</table>

Each of the two sources reports information to the feedback module using JSON-formatted messages. The latter is responsible for parsing each message in order to derive a list of attributes. The focus of the dialogue manager is more on issues related to user score and recognition result, whereas the client reports information based on user interaction. Table 2 summarizes the most important information reported by each source.

3.3. Audio analysis

Results from our previous user studies have identified specific issues related to the usage of the microphone. Silent speech or high background noise are typical problems. Another deficiency of our interface is that students are forced to use a push-and-hold button to record their voice; it presents various usability problems, in particular that users release the button too early.

Thanks to being deployed on the web, CALL-SLT can function as an efficient corpus collection tool. We already had at our disposal an English-L2 child learner speech corpus, produced by 15 Swiss German-L1 students in their third year of learning [20]. The corpus consists of 814 analyzed and annotated utterances. Most of the utterances are spontaneous speech, recorded while interacting with the language game as explained in section 2; there was also a small amount of read speech included (66 utterances; 7.7%). Read speech utterances are typically longer than spontaneous speech and count between 7 and 19 words (average of 15.3 words). The length of spontaneous speech utterances is much shorter, with almost a quarter of all utterances (23%) consisting of as little as one word. Utterances with up to six words are common, but the frequency drops sharply after eight words. Among the different annotations of the utterances, we were particularly interested in audio quality. This was identified using the following tags: low background noise, high background noise, low volume, high volume, cut-off, discard and non-linguistic interruption (e.g. cough). The corpus was extended with 10 utterances consisting of silence and 20 null utterances (very brief press on the recording button).

We used the software program praat [21] to extract 27 typical voice analysis measurements for each utterance, including: the mean pitch, mean intensity, jitter, shimmer, waveform duration, etc. (Table 3). Using the WEKA Toolkit [22], we applied different machine learning algorithms in order to classify robustly an utterance to each one of the aforementioned tags. The initial error rate, however, was too high for real experimentation (around 75%). Moreover, some of those tags were underrepresented in the corpus. Thus, we restricted our study to the most common problems encountered in real interactions, namely, high background noise, silent speech and instant press of the recording button. Incorporating a correlation-based feature subset selection method [23], we restricted the subsequent study using a space of 8 features (Table 3).

Classification was performed using Naive Bayes, Ensembles of Nested Dichotomies [24], Multilayer Perceptron with back propagation (one hidden layer with 10 hidden nodes, learning rate 0.3 and momentum 0.2, 500 epochs sigmoid for activation), Decision Trees implementing C4.5 pruned algorithm, Random Forest of 10 trees considering 4 random features classifiers [25], Functional Trees [26], SVM (polynomial kernel and trade-off between training error and margin 5,000), and Nearest-neighbor using non-nested generalized exemplars [27]. The results, using 10-fold cross-validation, are shown in Table 4. The rates measure classification accuracy for the 3 error tags plus 1 class for utterances where there is no problem. As shown in the table the accuracy is high across all methods.
that looks similar to Java source code. It can be written either in XML or using the so-called mvel dialect to integrate. The framework is based on Java and the rules can be error-prone and generally difficult to maintain. On the other hand, expressing it with external rules can alleviate these issues."
4. User Study

In this section we describe the findings of our experiment. In general, CALL systems evaluations need to incorporate a control group during the experiment, otherwise they run the risk of misinterpreting the obtained results [11]. Determining whether a CALL tool really helps students learn involves allowing a substantial group of students to use it over an extended period, tracking their progress, performing in-depth interviews with the subjects, and comparing the students in different conditions [29]. It is also clear that the efficacy of a given tool depends on the level of the students and the environment in which it is used. An interesting consequence of the fact that speech-based systems can now be put on the internet is that it enables use of crowdsourcing methods [30].

The experiment described here steers an intermediate approach. An easily accessible web-based application is offered to students, who, however, are chosen in the context of regular school classrooms. A formative feedback version is compared to a non-feedback one, in terms of motivation and problem solving, and less in terms of learning. Three classrooms in three different secondary schools in German-speaking Switzerland started to use one of the two versions. Each experiment was organized around an approximately one month period. Students were asked to use the system at home like a homework assignment; however, it did not affect their final grade. Table 6 summarizes some key parameters for the three schools, including the classroom id, the number of participants, their age distribution, the approximate level of the classroom compared to the average in the canton, the duration of the experiment and the availability of formative feedback.

Table 6. Experiment set-up for the three classrooms.

<table>
<thead>
<tr>
<th>Id</th>
<th>Participants</th>
<th>Age (sd)</th>
<th>Level</th>
<th>Duration</th>
<th>Feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>17</td>
<td>13.9 (0.5)</td>
<td>mixed</td>
<td>4 weeks</td>
<td>No</td>
</tr>
<tr>
<td>S2</td>
<td>13</td>
<td>15.1 (0.6)</td>
<td>weak</td>
<td>4 weeks</td>
<td>No</td>
</tr>
<tr>
<td>S3</td>
<td>9</td>
<td>13.8 (0.4)</td>
<td>strong</td>
<td>4 weeks</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For the subsequent analysis we decided to keep sessions that included 5 interactions at least, while the number of discarded sessions was evenly distributed among the three schools. Initially, we quantify the number of times students in S3 received some kind of feedback. As we can observe in Figure 5, all students received a positive or a negative message approximately after half of their interactions. This was actually a design consideration during the creation of the priority rules (using the heuristics of section 3.4) in order to avoid overwhelming students with too much feedback.

Figure 5: Distribution of feedback type for each user in S3.

Figure 6 shows the distribution of messages in terms of frequency. The acquired bonus points message is the most frequent feedback presented to students. On the other hand, reporting consecutive interaction mistakes (misrecognitions) and informing users not to barge-in while the video prompt is playing are the most frequent negative messages. In general, problems related to the microphone or how to use the interface have low frequency. This indicates that even when these problems occur, they can be easily fixed and they are not frequently repeated.

The analysis continues with some quantitative comparisons among the participants of schools S1, S2 and S3. The comparisons (Figure 7) are based on five different dimensions: (a) student score per session, (b) interactions per session, (c) total number of interactions, (d) total number of sessions and badges acquired (e). A single-factor ANOVA identified significant differences among schools on (b) (F(2, 34) = 11.7, p<0.001) and (e) (F(2, 21) = 16, p<0.001), without being able to verify if the feedback mechanism was the sole factor for this difference and if the diversity in the level of each classroom was more decisive. Plot (a) shows no significant differences between the average scores of S1 and S3, which means that they performed equally well in the speech recognition task. Moreover, the number of total sessions is practically the same across configurations (plot (d)). Conversely, students at S3 did much more interactions per session (plot (b)), which indicates that they were motivated only during the game play. This led to a greater number of total interactions (plot (c)) and subsequently to the acquisition of more score badges (plot (e)).

Figure 8: Frequency of feedback presented in the subsequent interaction after an error. Interrupted video prompt (a), many consecutive mistakes (b) and instant press of the recording button (c).
Besides motivating students, this mechanism seems to help students to resolve interaction problems. The spider charts of Figure 8 show three cases of error conditions. We try to quantify if the error has been resolved by examining the subsequent interaction. For example, an instant press of the recording button followed by an interaction with successful recognition indicates that the student understood what caused the problem. From the charts we can observe low values of error repetition in all cases, so it can be reasonably argued that students were indeed helped.

Finally, we are also interested in the subjective opinion of students concerning this type of interaction. A post-experiment questionnaire was distributed to all students at the end of each experiment. Three of the corresponding answers are presented in Figure 9, and show no evident consensus towards our system. Students at S2 exhibited the lowest performance during the experiment and this was reflected in their subjective opinion. S1 students expressed the most positive comments and S3 users were equally split in two halves. Interestingly, the majority of the students reported that their motivation to use CALL-SLT came from the fact that it was part of their homework assignment, which was technically obligatory but no grade was assigned to it.

5. Conclusions

We have described a formative feedback mechanism in the context of a CALL system that uses information from different sources. We performed experiments at three secondary schools with and without this mechanism and from the findings it can be reasonably argued that a parallel feedback mechanism in L2 actually does help students during interaction and contributes as a motivation factor.

The design of the current module opens interesting research directions, as it is straightforward to customize and adapt the formative feedback in terms of phrasing, style and language. Mainstream research suggests that humans form social partnerships with computers during interaction [31]. We can easily see the current module being transformed into a virtual advisor, tutor or a friend. Interaction in this context may reveal interesting implications, especially for children.

On the other hand we can immediately identify some limitations of the current implementation. Feedback is significantly more effective when it provides details of how to improve the answer rather than just indicating whether the student’s response is correct or not. In this respect the rules can be split into a verification part (help users to understand what went wrong) and an elaboration part (help them to decide what to do next). Additionally, with the current architecture it is not possible to provide timely feedback, for example postpone the response after several interactions.

I would use the tool again in future.

All in all I liked practising English with the tool.

What was your motivation to use CALL-SLT?

Figure 9: Subjective opinion of students

Figure 6: Frequency of positive and negative feedback.
6. References


