Linguistic **I**ssues in **L**anguage **T**echnology – **LiLT**

CALL-SLT: A Spoken CALL System

based on grammar and speech recognition

Manny Rayner
Nikos Tsourakis
Claudia Baur
Pierrette Bouillon
Johanna Gerlach

Submitted, May 2014
Revised, October 2014
Published by CSLI Publications

LiLT volume 10, issue 2 October 2014

CALL-SLT: A Spoken CALL System

based on grammar and speech recognition

Manny Rayner, *Geneva University,* Nikos Tsourakis, *Geneva University,* Claudia Baur, *Geneva University,* Pierrette Bouillon, *Geneva University,* Johanna Gerlach, *Geneva University*

Abstract

We describe CALL-SLT, a speech-enabled Computer-Assisted Language Learning application where the central idea is to prompt the student with an abstract representation of what they are supposed to say, and then use a combination of grammar-based speech recognition and rule-based translation to rate their response. The system has been developed to the level of a mature prototype, freely deployed on the web, with versions for several languages. We present an overview of the core system architecture and the various types of content we have developed. Finally, we describe several evaluations, the last of which is a study carried out over about a week using 130 subjects recruited through the Amazon Mechanical Turk, in which CALL-SLT was contrasted against a control version where the speech recognition component was disabled. The improvement in student learning performance between the two groups was significant at $p < 0.02$.

LiLT Volume 10, Issue 2, October 2014.
CALL-SLT: A Spoken CALL System.

1 Introduction and background

People have been building Computer-Assisted Language Learning (CALL) applications for several decades, and more recently it has become popular to include speech recognition as one of the components. The intuitive rationale is obvious: if the system has some ability to understand what the student is saying, then it may be better able to help them improve their spoken skills. The most common type of speech-enabled CALL system constructed to date has focussed exclusively on pronunciation practice. Many variants exist (an impressive and well-documented example is the EduSpeak® system (Franco et al., 2010)), but the basic scheme is simple: the system plays the student a recorded sentence, asks them to imitate it, and then rates them on the accuracy of their imitation, giving advice if appropriate on how to improve pronunciation or prosody. It is easy to believe that this is useful, but it is also very limited in scope: the student is given no opportunity to practice spontaneous spoken generation skills.

A more ambitious approach is to design an application where the student can respond flexibly to the system's prompts. The project we describe in this paper, CALL-SLT (Rayner et al., 2010), is based on an idea originating with (Wang and Seneff, 2007); a related application, described in (Johnson and Valente, 2009), is TLTCS. The system prompts the user in some version of the L1, indicating in an abstract or indirect fashion what they are supposed to say; the student speaks in the L2, and the system provides a response based on speech recognition and language processing. We have built several prototypes on this basic pattern, exploring different language-pairs and strategies. For example, in a minimal version configured to teach French to English-speaking students, a prompt might be the text string:

```
REQUEST HAMBURGER
```

and the student will be allowed to respond with any of the following alternatives:

Je voudrais un hamburger
J'aimerais un hamburger
Puis-je avoir un hamburger ?
Je prendrai un hamburger
...

In more complex versions, the prompt may be presented in multimedia form, or appear as part of a simple dialogue between the student and the application. Irrespective of the modality used, the student, in order to respond correctly, must be able to pronounce the French words well enough to be understood by the speech recogniser; they also need to

be able to construct a spoken French sentence whose meaning matches the content of the prompt.

In the rest of the paper, we first describe the core system (§2) and the various types of content we have developed (§3). We then present a number of evaluations (§4), where the central question addressed is whether the application is in fact capable of helping students improve their language skills. The final section concludes.

2 System architecture

2.1 Overview of architecture

The CALL-SLT system is based on two main components: a grammar-based speech recogniser and an interlingua-based machine translation system, both developed using the Regulus platform (Rayner et al., 2006). Each turn begins with the system giving the student a prompt, typically formulated in a simplified or telegraphic version of the L1, to which the student gives a spoken response; as already noted, it is in general possible to respond to the prompt in more than one way.

The system decides whether to accept or reject the response by first performing speech recognition, then translating to 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. The text forms come from the initial corpus of sentences, which is supplied by the course designer; the associated audio files are collected by logging examples where users registered as native speakers got correct matches while using the system. Prompts are grouped together in "lessons" unified by a defined syntactic or semantic theme, and may optionally be linked together by a script to form a simple interactive dialogue.

The student thus spends most of their time in a loop where they are given a prompt, optionally listen to a spoken help example, and attempt to respond. If the system accepts, they move on to a new prompt; if it rejects, they will typically listen to the help example and repeat, trying to imitate it more exactly. On reaching the end of the lesson, the student either exits or selects a new lesson from a menu.

In the rest of this section, we describe the main components: grammar-based recognition and rule-based translation, lesson structure, user interface and web deployment.

2.2 Grammar-based speech and language processing

The speech and language processing components of the system are entirely rule-based. The same grammar is used both to provide the lan-

guage model which constrains the speech recogniser, and to parse the result of performing speech recognition. The intent is to ensure that all speech recognition results will be within the coverage of the language processing components, which greatly simplifies the architecture by obviating the need for robust parsing and semantic interpretation.

The underlying platform is the Nuance Toolkit[1], which provides facilities for defining context-free (CFG) grammars that can be compiled into language models and also used for parsing. It would be possible to write these grammars by hand, though doing so involves the usual problems, given that CFG is not a very expressive formalism; it is hard to model any non-trivial linguistic phenomena without producing an extremely large grammar that is in practice almost impossible to extend or maintain. A well-known method for addressing these issues is to specify the grammar in some higher-level formalism — most obviously, some kind of feature grammar — and compile this down to CFG form (Stent et al., 1999). This time, the problem is in another direction. Even if a substantial, linguistically motivated feature grammar can in principle be expanded out to a CFG grammar (this is of course by no means guaranteed), the resulting grammar will probably be so large that it exceeds the practical resource limits imposed by the speech recognition framework. For these reasons, grammars that can actually be used as language models need to be domain-specific; this, unfortunately, conflicts with the natural desire to make the grammars general and reusable.

The Regulus system steers a middle course between these alternatives. For each language, we construct a central resource grammar implemented using a feature-grammar formalism, but we do not compile this directly into a CFG language model. Instead, we first extract a domain-specific subgrammar using Explanation-Based Learning (EBL) methods driven by small corpora typically containing a few hundred examples. The scheme is explained in detail in (Rayner et al., 2006), which also includes a thorough description of the English resource grammar. Similar grammars have since been developed for French, German and Japanese. These have been further extended to cover related languages by a parameterization process. In particular, the French grammar has been extended into a shared grammar which covers French, Spanish and Catalan (Bouillon et al., 2007), and the English grammar has been similarly extended to cover both English and Swedish (Rayner et al., 2012b).

The Regulus resource grammars are also parameterized to support

[1] `www.nuance.com`

multiple types of semantic representation. In all the work reported here, the semantic formalism used is Almost Flat Functional Semantics (AFF; (Rayner et al., 2008)), a minimal formalism where clauses are represented as unordered lists of elements tagged with functional roles. For example, "Could you give me directions to the zoo?" is represented as the structure

```
[null=[utterance_type, ynq],
 agent=[pronoun, you],
 null=[modal, could],
 null=[action, give],
 null=[voice, active],
 object=[abstract, directions]
 indobj=[pronoun, i],
 to_loc=[loc, zoo]]
```

2.3 Using interlingua to display prompts

The AFF representations produced by speech recognition and parsing are translated into a language-neutral form, also expressed using AFF. The minimal list-based format means that translation rules can be very simple: basically, they map tagged lists onto tagged lists, possibly conditional on the surrounding context. The details are provided in (Rayner et al., 2008).

The space of possible interlingual forms is defined using another Regulus grammar, which associates each valid interlingual AFF representation with a surface form. In the context of the CALL-SLT system, the intention is that these interlingual representations form a reduced, simplified and abstracted version of the English grammar, while the surface form is used as part of the prompt given to the student. Thus, continuing the previous example, suppose that the system is configured with English as L1 and German as L2, i.e. to teach English to a German-speaking student. The AFF form corresponding to "Could you give me directions to the zoo?" is converted into the interlingual representation

```
null=[utterance_type, request]
arg2=[abstract, directions],
to=[loc, zoo]
```

using a set of translation rules of which the least trivial is the one which maps the elements corresponding to "Could you give me..." to the single element `null=[utterance_type, request]`.

We have experimented with several strategies for defining interlingua grammars: as usual, there are a number of competing requirements. On

the one hand, we want the grammar to be structurally simple, so that the process of converting interlingual AFF representations to surface forms can be fast as possible. A second requirement is that the surface forms should be fairly natural-looking L1 expressions. A third is that it should be easy to port the system to support a new L1; in practice, this task is often carried out by people who have domain and L1 expertise but little knowledge of computational linguistics.

The compromise we have found best in practice is to define a minimal interlingua grammar parameterized in two ways. First, there are hooks allowing generic changes to the default Subject-Verb-Object word-order: for example, if the L1 is Japanese, we want the verb to be preceded by its complements, and for German we want the modifiers to come before the verb if it is not the main verb. The grammar is kept very simple to make it feasible to allow this kind of flexibility: for example, we do not have any kind of moved question construction, so the word order is "You want what?" rather than "What do you want?". The only concession to grammatical agreement is to add a formal affix to the verb, making it agree with the subject.

The second type of parameterization is to handle surface forms. The plain grammar produces surface forms using English words; a final processing step uses a set of surface rewriting rules to map English words into the final L1.

Although the scheme cannot be called elegant, it is easy to implement and maintain, and performs well in practice. Table 1 presents examples of some typical prompts taken from the English L2/German L1 version, showing the original English example, and the translated prompt both before and after surface rewriting.

i would like a double room ask-for : double room frag : Doppelzimmer
can i pay by credit card say : i want-to 1-SING with credit-card pay INF sag : ich möchte mit Kreditkarte bezahlen

TABLE 1 Examples of prompts in the English L2/German L1 version of the system, showing the original English corpus example and the German prompt before and after surface rewriting.

2.4 Providing help examples

When the student is uncertain how to answer a prompt, the system can provide them with an example of a correct response. Since every prompt is derived from an example taken from the L1 corpus, at least one example is always available. Given that the interlingua grammar used to define the prompts has less structure than the L1, and maps many L1 sentences onto the same prompt, there are typically several possible help examples available.

By recording successful interactions made by users registered as native speakers of the L1, the system can also store spoken help examples. When a course is being created, the course designer usually arranges for a native speaker to cycle through all the examples until they have successfully completed each one, ensuring that both written and spoken help are always available. The user interface supports a special "recording mode" designed to support this task, where the system only offers the user prompts for which recorded examples examples do not yet exist. Each speech example is tagged with the words found by the speech recogniser when it was recorded, so that students can be offered both text and speech help. Since slightly incorrect responses can still be counted as successful matches (most often, an article is inserted or deleted), a second pass is required to correct erroneous transcriptions. This is done efficiently by creating an editable HTML table which includes both transcriptions and links to the associated speech files.

2.5 Lesson structure

For pedagogical reasons, it is desirable to group the examples into thematic units; we call these units "lessons". The most straightforward alternative is to divide up the corpus into a number of possibly overlapping sets, each set corresponding to a lesson. The unifying theme of a lesson will typically be either the language required to carry out a concrete task (booking a ticket, ordering at a restaurant, buying clothes), or one or more types of grammatical construction (numbers, dates, simple questions). In the latter case, it is also possible to define the content of each lesson by listing the semantic properties (grammar rules used, subtrees in the parse-tree, or lexical items).

Grouping examples into lessons creates structure and makes the activity of practicing with the system feel more focussed and meaningful. This suggested to us that introducing further structure might be worthwhile. To this end, the most recent version of the system allows the course designer to add a simple dialogue script to the lesson. The script, written in a minimal XML notation, defines a number of steps, typically about 10 to 20; the specification of each step includes a unique

```
<!-- Ask for number of nights -->
<step>
   <id>ask_for_number_nights</id>
   <multimedia>how_many_nights</multimedia>
   <group>room_for_number_of_nights</group>
   <repeat>ask_for_number_nights</repeat>
   <limit>is_one_night_okay</limit>
   <success probability="25">not_available</success>
   <next_success>ask_type_of_room</next_success>
</step>
```

FIGURE 1 Example step from a lesson whose theme is "booking a hotel room". When the step is executed, the multimedia file `how_many_nights` shows a clip of a cartoon desk clerk asking the student how many nights they wish to stay for; the associated prompt will be taken from the group `room_for_number_of_nights`, and, rendered in the L1, will mean something like "request: room for three nights". The `repeat` tag says to repeat the step if the student's response is not accepted. If it is not accepted three times, the `limit` tag says to move to the step `is_one_night_okay`, where the student is asked a simple yes-no question. Conversely, if the response is accepted, the two `success` tags say to move either to the step `not_available` (25% probability) or otherwise to `ask_type_of_room`. The step definition has been slightly simplified for expositional reasons.

ID, a group of prompts, a recorded multimedia file, and the steps to move to next depending on different conditions. Figure 1 shows a typical step.

Execution of a step proceeds as follows. The system plays the multimedia file and displays one of the prompts. The student responds (possibly after asking for help), and the system performs recognition, translation and matching to decide whether the response is accepted or rejected. Depending on the result, it either repeats the step or moves to a new one.

2.6 User feedback

The user feedback in the versions of CALL-SLT which we will describe here is the minimal possible: the system either accepts or rejects the student's response.

It would obviously be desirable in principle to provide more feedback, giving the student some idea of what they have done wrong when the system rejects. Unfortunately, experience shows that this information must be very reliable in order to be helpful, and that unreliable feedback is worse than useless. Simply rejecting is less confusing than rejecting

and adding an explanation based on something that the student didn't actually say; highly reliable recognition of incorrect responses is, for obvious reasons, a challenging task.

Since doing the work reported in the current paper, we have carried out experiments with other versions of the system, which leave us guardedly optimistic that it may be possible to recognize certain specific types of incorrect response with high enough reliability. This idea is still at a preliminary stage, and will be reported elsewhere.

2.7 Web deployment and user interface

CALL-SLT is deployed over the web using a scalable architecture, developed by Paideia Inc, California and particularly designed for cloud-based computing. In common with similar platforms, like WAMI ((Gruenstein et al., 2008); `http://wami.csail.mit.edu`) and the Nuance Mobile Developer Platform (NMDP; `http://dragonmobile.nuancemobiledeveloper.com`), it uses a client/server approach in which speech recognition is carried out on the server side; the Paideia architecture, however, goes further than these systems by performing dialogue management, application integration, and large-scale grammar-based language processing on the server, rather than just returning the results of recognition to the client. Another important difference is that speech is passed to the recognition processes in the form of files, rather than using streaming audio. Although this goes against the currently prevailing wisdom, we have found that there are compensating advantages, and that the performance hit, with a little care, can be reduced to only a couple of hundred milliseconds per recognition operation. Full details are presented in (Fuchs et al., 2012).

By moving almost all processing to the server, the client can be kept very simple. It only needs to be responsible for the graphical user interface, maintaining a small amount of state logic, performing recording and playback of audio, and requesting services from the remote peer. Versions of the client for standard browsers have been developed using Flash 11 in combination with ActionScript 3.0.

An important aspect of the GUI is the way the recognition button is used. Due to the limitations of the target platform (lack of an endpointing mechanism), we have adopted a push-and-hold solution, where the user has to keep the button pressed while speaking. The recorded audio packets are streamed to the server until the button is released. The latter signifies the end of the user speech, triggering the transition to the next step of the processing chain; recognising using the remote audio file, processing the result and returning a response to the client.

3 Designing content

We briefly describe the various types of content we have developed for use in CALL-SLT.

Early work, during the first year and a half of the project, was centred on the restaurant domain. We developed courses for several L2s, with most of the work focussing on English, French and Japanese. The courses were divided up into lessons by syntactic theme: for example, the French course (the most elaborate one), included lessons for singular and plural nouns, numbers, location expressions, different kinds of question construction, etc (Bouillon et al., 2011b).

Subsequently, work has diverged in several different directions. A basic tourist Japanese course was built together with Future University, Hakodate; here, the unusual idea was to focus on adjectives ("excellent", "difficult", "expensive" etc) to rapidly build up a basic communicative vocabulary. The course was tested over the Amazon Mechanical Turk (AMT), with some success (Rayner et al., 2011).

At the same time, a set of elementary French lesson was developed together with the University of Bologna, and tested in conjunction with one of their courses (Bouillon et al., 2011a); the topics covered included "greeting", "talking about my family" and "scheduling a meeting". A modified and abbreviated version of this course was adapted for use on mobile devices, and also tested over AMT; this work is described in detail in the next section.

Most recently, we have been developing an interactive multimedia course for teaching English to beginner German-speaking school students; some examples are shown earlier in §2.5. A first evaluation with real students was carried out in late 2013 and early 2014 (Baur et al., 2014, Tsourakis et al., 2014).

In the next section, we describe in detail some of the more substantial evaluations we have carried out using the French L2 courses.

4 Evaluations

The central question we consider is an apparently simple one: does the system help students improve their generative language skills? Here, we summarise from this point of view the results of three concrete experiments carried out between 2011 and 2013.

One-day experiment using French/Chinese system

A typical early study was the one described in (Bouillon et al., 2011b), where the subjects were 10 Chinese-speaking computer science students spending an exchange year in Tours, France. The students, who had previously done between one and two years of French in China and

spent five months in France, were asked to use the French-for-Chinese version of the system, loaded with five sample lessons. They took part in two sessions, totalling about three hours in duration and yielding a total of 5245 recorded spoken interactions. Each spoken response was stored in recorded form, together with meta-data including the associated system prompt.

In the paper, we argued that the results provided evidence that subjects had learned from using the system. First, students had a higher proportion of utterances accepted by the system in the later utterances than in the earlier ones, this difference being statistically significant. Second, grammar and vocabulary tests carried out before and after the experiment showed large differences; most of the students appeared to have picked up some vocabulary, and there was also reason to believe that they had consolidated their knowledge of grammar.

Looking critically at the design, we can advance various objections against the validity of our conclusions. One obvious question is whether the fact that students have more utterances accepted by the system after they have used it for a while really does mean that they have improved their generative spoken language skills. Other explanations are a priori quite possible. In particular, they may only have become more skillful at using the interface, learning to speak in a way that is better adapted to the machine, but not necessarily better in itself.

Contrastive judging of data from first experiment

The follow-on experiments described in (Rayner et al., 2012a) were designed to investigate how much substance there was to these potential criticisms. To this end, we collated the data so as to find cases where a) the same student had responded more than once to the same prompt, and b) at least one example had been accepted, and at least one rejected. For each such group, we randomly selected one recorded file which had been accepted and one which had been rejected, giving us 413 pairs.

An initial sampling of the data quickly revealed that, in many cases, the most important characteristic was that one or both files had been badly recorded. (Among other things, the experiment had been carried out in a small room with bad acoustics). We consequently divided judging into two rounds. During the first round, two system experts listened to all the pairs, and marked ones which exhibited recording problems: this accounted for 243 pairs, about 56% of the data.

The remaining 170 pairs were then judged by three French native speakers, all of whom had worked as French language teachers. None of them had previously been associated with the project or knew the

exact point of the evaluation exercise: in particular, we were careful not to tell them that each pair consisted of one successful and one unsuccessful recognition match. Judges were asked to mark each pair to say which element, if either, was better in terms of speech (pronunciation/prosody), vocabulary and grammar. Judging was performed using AMT, with the judges paid a zero fee. This allowed us to distribute work efficiently over the Web and also simplified the task of writing the judging interface, which could be specified straightforwardly in HTML.

Judge	Agree	Disagree	Null
All judgements			
1	82	40	48
2	87	31	52
3	99	51	20
at-least-1	134	80	7
majority	90	30	50
unanimous	44	12	114

TABLE 2 Agreement between system responses and human judgements on 170 well-recorded contrastive pairs. "Agree" means the judge(s) marked the element of the pair accepted by the system as better; "Disagree" means they marked it as worse; "Null" means no preference.

The results are shown in Table 2. For each judge, we list the number of pairs on which they explicitly agree with the system (i.e. the judge considered that the accepted element of the pair was better) and the number where they explicitly disagree (the judge preferred the rejected element). If the judge did not express a preference with respect to any of the specified criteria, we counted the pair as being neither an agreement nor a disagreement. We also list results for aggregated judgements that are "unanimous" (all three judges), "majority" (at least two out of three judges) and "at-least-1" (at least one judge).

The notion of "quality of spoken response" is slippery; since we refrained from giving detailed guidelines, we were not surprised to see a fair degree of disagreement between the three judges. Even with respect to vocabulary and grammar, which one might expect to be reasonably uncontroversial, we found many differing judgements. For example, one judge thought a full sentence was grammatically better than a nominal phrase, while the other two considered them equally good. However, when we look at the "majority" judgements, we find a reassuring correlation between the human and mechanical evaluations; the judges

agreed with the recogniser three times as often as they disagreed with it (90 versus 30). It is also worth noting that there are few cases of unanimous disagreements, and that, even when all the judges unanimously disagree with the recogniser, they often do not disagree for the same reasons: for example, one judge may think that the rejected utterance was better due to pronunciation, and another due to grammar.

4.1 AMT-based evaluation using a control group

With the above results in mind, we decided to attempt a more ambitious evaluation that would address some of the issues that had arisen. We start by outlining the methodological problems, then describe the experiment itself.

Evaluation methodology

The experiment described immediately above suggests that there is a reasonable correlation between the quality of the students' speech and the frequency with which the system accepts their utterances; if students get more utterances accepted, this can thus reasonably be taken as evidence that they have improved their generative language abilities. Unfortunately, even if the results unambiguously show that the student has improved over a given period, it is still not clear what has caused the improvement. This problem is particularly acute when use of the system is integrated into a formal language course, as in (Bouillon et al., 2011a, 2012); given that the student is also receiving other kinds of instruction, it is obviously possible that any improvement measured is independent of use of the system. Even if the student is only learning through use of the system, at least over the duration of the experiment, it is still unclear which aspects of the system are responsible for the improvement. In an application like CALL-SLT, the student spends a large part of their time listening and repeating, which may well be helpful for them. It remains to be shown that any of the more sophisticated system functionalities are useful in practice. In particular, the obvious question is whether we can find concrete evidence to show that speech recognition feedback, which can never be more than partially reliable, is actually assisting the learning process. The only straightforward way to demonstrate this is to find some way to compare a version of the system which has recognition feedback against one that lacks it, and find a clear difference.

The question is how to organise an experiment to perform a comparison of this kind. A natural idea is to separate the students into two groups, a main group and a control; the first group uses the system with recognition, and the second the one without. Unfortunately, ex-

perience has shown that there are many practical problems with this design, partly because motivation is always an important factor in language learning. For example, suppose, as in e.g. (Coyne et al., 2011), that we pick subjects randomly from one class, assigning half of them to the group using the system and the other half to the control. The two groups of students will talk to each other. If the system is perceived as useful, which the authors claim in the cited study, it is reasonable to wonder whether students in the control group felt correspondingly unmotivated; it is methodologically better if no subject is aware that any version exists except the one they are using.

If, on the other hand, we take the two groups from two different classes that have no contact with each other, not mixing them, it is impossible to know whether the classes are comparable. Most teachers we have asked say their experience suggests high variability between classes. Yet another possibility is to use a crossover methodology, letting students in the same class alternate between the two groups. Some clear successes have been claimed for this methodology, in particular by the LISTEN project (Poulsen, 2004, Reeder et al., 2007, Korsah et al., 2010); if the learning effect from using the system is large enough, as appears to be the case there, it is reasonable to hope for a clear result. There are however many known problems with crossover, since it is difficult to account correctly for the effect of using the main system and the control version in different orders. In the context of CALL, students may once again be disappointed if they like the main system and are then forced to use the inferior control, and react accordingly.

For the kinds of reasons just given, it has often been argued that experiments with control groups are unproductive in CALL (Kulik et al., 1980), and that single-case design methodologies (Kennedy, 2005) are more appropriate. The central idea of the single-case methodology is that the student acts as their own control. There is an initial period where the student is exposed to the baseline condition; this needs to continue long enough for a rate of progress to be reliably estimated. The new condition is then introduced, and the critical measure is the difference in the student's rate of progress between the baseline period and the period where the new condition applies. Here, too, the experimental design is fraught with difficulties if we wish to apply it to a test of the kind we are considering. We would have to develop sufficient course material to allow students to use the baseline system for a period long enough to estimate their initial rate of progress; lack of uniformity of the material would mean that it had to be tried in many different orders; and it is not obvious how to correct for the fact that students who have become used to the baseline version are then obliged to reac-

climatize to the enhanced one when they reach the crossover point. Note that the "AB" design (baseline followed by intervention) is the best case; versions in which intervention is followed by baseline are even more problematic, since students are likely to be disappointed by going back to a version they perceive as inferior.

It may seem that there is no satisfactory solution, and that the best we can do is rely on informal assessments based on subjective teacher impressions. Recently, however, the introduction of crowdsourcing platforms has opened up new possibilities (Jurčiček et al., 2011, Eskenazi et al., 2013). In a large, diverse online community, it is not unreasonable to hope that most subjects will have no contact with each other; under circumstances like these, the design with two separated groups becomes more like the "randomized clinical trial" design typical of medical studies, and is correspondingly more plausible.

Experiment

In the remainder of this section, we describe a crowdsourced experiment with two separated groups, carried out in early 2013 using a multimedia-enabled Android phone version of the French CALL-SLT system. We only set ourselves the modest goal of establishing that use of speech recognition feedback produced a significant difference in short-term learning outcomes. A more elaborate experiment would also have investigated the question of how well subjects retained the knowledge they had acquired, but this would have required much greater resources than we had available.

The main content of the course used consisted of four lessons, *about-me* (simple questions about the subject's age, where they live, etc); *about-my-family* (similar questions about family members); *restaurant* (ordering in a restaurant) and *time-and-day* (times and days of the week). Three additional lessons called *overview-1*, *overview-2* and *revision* will be described shortly. The course was designed for students with little or no previous knowledge of French. It covered about 80 words of vocabulary and a dozen or so basic grammatical patterns.

We created four different versions of the basic system. Three of them differed only in the way the multimedia part of the prompt was realised: in **video** it had the form of a recorded video segment of a human speaker, in **avatar** it was an animated avatar, and in **text** it was a piece of text. The fourth version, **no-rec**, was the same as **video**, except that the student was given no feedback to show whether speech recognition and subsequent processing had accepted or rejected their response.

Subjects were recruited through AMT; we requested only workers from the US. After discovering during a previous study (Rayner et al.,

2011) that experiments of this kind can easily attract scammers, we required all workers to have a track record of at least 50 previously completed Human Interface Tasks (HITs), at least 80% of which had been accepted.

The experiment was carried out in two cycles, each of which had the same sequence of eight HITs. In the first HIT, the task was to check that one version of the app (we chose **no-rec**) could be successfully run on an Android phone. Subjects who gave a positive response were then randomly assigned to the four different versions of the system and given different versions of the subsequent HITs. AMT "qualifications" were used so that subjects doing one version of a HIT were unable to see that HITs for other versions existed. The seven HITs were issued at 24-hour intervals; workers were paid $1.00 for the first HIT and $2.00 for each subsequent one, reasonable pay by AMT standards. The HITs had the following content:

Round	Remaining	
	Cycle 1	Cycle 2
Recruit	80	22
Pre-test	36	14
About-me	29	11
My-family	24	10
Restaurant	22	9
Time-and-day	20	8
Revision	18	8
Post-test	17	7

TABLE 3 Number of students left after each round in the two cycles.

Pre-test: The student was asked to do *overview-1* and *overview-2*, each of which consisted of a balanced selection of examples from the other lessons. During *overview-1*, they were encouraged to use the Help function as much as they wished, so the main skill being tested was ability to imitate. In *overview-2*, Help was switched off, so the main skill tested was generative ability in spoken French.

Lessons 1–4: The student was asked to attempt each of the four lessons in turn, one lesson per HIT, with Help turned on. They were told to spend a minimum of 20 minutes practising, and speak to the system at least 25 times.

Revision: The student was warned that the next HIT would be a test (they were not told what it was), and was asked to revise

by doing the *revision* lesson, which contained the union of the material from the four main lessons, for at least 20 minutes.

Post-test: The student was asked to do *overview-1* and *overview-2* again. They were told that the intent was to measure how much they had learned during the course, and were asked to do the test straightforwardly without cheating.

rec			**no-rec**		
ID	B-S-W	Signif	ID	B-S-W	Signif
1	1 7-13-8	—	13	6-18-3	—
2	4-14-2	—	14	4-13-1	—
3	9-6-1	$p < 0.05$	15	2-18-2	—
4	9-18-1	$p < 0.05$	16	7-15-6	—
5	8-19-0	$p < 0.02$	17	7-19-2	—
6	10-12-5	—	18	14-9-4	$p < 0.05$
7	6-12-1	—	19	18-7-3	$p < 0.01$
8	8-5-0	$p < 0.02$	20	4-22-1	—
9	6-15-3	—	21	5-15-6	—
10	5-14-9	—	22	10-15-2	$p < 0.05$
11	9-12-2	—	23	5-17-6	—
12	12-11-5	—	24	9-17-2	—

TABLE 4 Improvement between pre-test and post-test for **rec** and **no-rec** versions, broken down by student. "B-S-W" shows the number of prompts on which the student performed BETTER, SAME and WORSE. "Signif" gives the significance of the difference between BETTER and WORSE according to the McNemar test. Students who described themselves as beginners are underlined.

The purpose of the pre- and post-tests was to measure the progress the students had made during the main course of the experiment by comparing their results across the two rounds. The mode of comparison will be described shortly.

In the first cycle, we started with 100 subjects. The second column of Table 3 shows the number of students left in play after each round of HITs. At the end of the cycle, there were 17 students who had completed both the pre- and post-tests. A preliminary examination of the results suggested that students performed similarly on the three versions which gave recognition feedback, but worse on **no-rec**; there was not, however, sufficient data to be able to draw any significant conclusions.

We decided that the most interesting way to continue the experiment was to collect more data for **no-rec**; in the second cycle, we consequently started with 30 subjects, assigning all of them to the **no-rec** group. The third column of Table 3 shows the number left after each round. At the end of the cycle, we had adequate data for 12 subjects in **no-rec** and 12 in the union of the three groups which included recognition feedback, which we will call **rec**.

The analysis in the next section focusses on exploring the difference between **no-rec** and **rec**, and our basic strategy is as follows. For each of the two versions, we compare student performance in the pre- and post-tests; we wish to determine whether this difference is significantly larger in **rec** than in **no-rec**.

Prompt	BETTER-SAME-WORSE, score			
	rec		**no-rec**	
With help				
P1	**7-1-1**	**66.7**	4-7-1	25.0
P2	**2-8-0**	**20.0**	3-7-3	0.0
P3	1-6-0	14.3	**5-5-2**	**25.0**
P4	**1-7-0**	**12.5**	1-8-4	–23.1
P5	3-3-3	0.0	**4-7-2**	**15.4**
P6	**5-6-3**	**14.3**	5-7-4	6.2
P7	4-3-5	–8.3	2-7-3	–8.3
P8	**3-2-2**	**14.3**	2-5-3	–10.0
P9	**3-2-2**	14.3	**6-7-3**	**18.8**
P10	4-4-1	33.3	**5-5-1**	**36.4**
P11	**4-6-1**	**27.3**	3-5-4	–8.3
P12	**6-6-2**	**28.6**	4-7-2	15.4
P13	**3-4-0**	**42.9**	4-6-2	16.7

TABLE 5 Improvement between pre-test and post-test for **rec** and **no-rec** versions on examples *with* help (i.e. testing pronunciation only), broken down by prompt. The version with the larger improvement is marked in **bold**.

The pre- and post-tests are the same[2] and consist of two halves. The first half ("with-help", i.e. online help was available) contains 13

[2]We wondered if it was methodologically sound to use the same items for the pre- and post-tests. Students were however going to take the two tests at least a week apart, during which they would practice many similar examples. We felt it was unlikely that they would remember the specific sentences from the pre-test, and that it was more important to give ourselves the option of performing a clear item-by-item comparison.

Prompt	BETTER-SAME-WORSE, score			
	rec		**no-rec**	
Without help				
P14	**3-8-0**	**27.3**	4-7-1	25.0
P15	**3-7-1**	**18.2**	1-10-1	0.0
P16	3-5-2	10.0	**4-6-0**	**40.0**
P17	4-4-3	9.1	3-6-2	9.1
P18	1-10-0	9.1	**2-9-0**	**18.2**
P19	5-4-1	40.0	**6-5-1**	**41.7**
P20	**2-8-0**	**20.0**	3-7-2	8.3
P21	**6-3-2**	**36.4**	3-6-2	9.1
P22	**4-6-2**	**16.7**	2-7-1	10.0
P23	**2-7-0**	**22.2**	1-9-1	0.0
P24	**3-7-1**	**18.2**	1-8-1	0.0
P25	2-7-1	10.0	**2-9-0**	**18.2**
P26	3-7-1	18.2	**2-8-0**	**20.0**
P27	**7-4-0**	**63.6**	7-5-0	58.3
P28	**3-6-0**	**33.3**	3-9-0	25.0

TABLE 6 As Table 5, but examples *without* help, i.e. testing both pronunciation and recall.

prompts; the second ("without-help", i.e. online help was not available) contains 15 prompts. We compare a given student's performance on each prompt by determining whether the system accepts the student's response or not. As already noted, this correlates reasonably with human judgements. Students can get BETTER (not recognised in pre-, recognised in post-), WORSE (recognised in pre-, not recognised in post-), or stay the SAME (identical outcomes in both tests).

We can compare either across students or across prompts. The simplest way to compare across students is to take each student and count how many examples of BETTER/WORSE/SAME (B/W/S) they get. We can then look at the difference between BETTER and WORSE using the McNemar test to find how significant it is (Table 4); note that $B + S + W$ does not always total to 28, since students sometimes omitted a few items from one or both tests. The comparison turns up four students in the **rec** group who get a significant difference, against three in **no-rec**; in the right direction, but obviously not strong evidence that **rec** is better. Other more complex tests also failed to show a statistically significant difference when we compared across all students, though some were close. It is however worth noting that we do get a significant difference on the two-tail t-test ($t = 1.7$, $df = 11$,

$p < 0.01$) when we use only the subset of students, underlined in the table, who described themselves as beginners. The fact that some students in the control group improve should not be regarded as surprising, since one expects simple listening and repeating to be useful; this is, indeed, exactly why we need the control group (or some other kind of controlled methodology) in order to make claims about the utility of speech recognition feedback.

Comparing across prompts produces a convincing result even when we use all the students (Tables 5 and 6). This time, we look at all the B/W/S scores for a given prompt and version, using the measure $(B - W)/(B + W + S)$. The value will be 100% if every example is BETTER, zero if BETTER and WORSE are equal, and –100% if every example is WORSE.

We can now perform a prompt-by-prompt comparison of **rec** and **no-rec**, contrasting the scores. For example, looking at prompt P11, we have under **rec** $B = 4$, $S = 6$ and $W = 1$, giving a score of $(4 - 1)/(4 + 6 + 1) = 3/11 = 27\%$. Under **no-rec**, we have $B = 3$, $S = 5$ and $W = 4$, giving a score of –8.3%. Applying the Wilcoxon signed-rank test to the whole set of prompts, the comparison between **rec** and **no-rec** on the above measure yields a difference significant at $p < 0.02$.

5 Summary and discussion

We have presented an overview of CALL-SLT, a spoken CALL application implemented using grammar and rule-based translation technologies. The current system has the status of a mature prototype; there are versions for several languages, over 50 lessons of content, and a stable web deployment. We have been able to use it to carry out substantial evaluations, which yield reasonable evidence that students who use it actually do learn compared with a simpler control system that lacks speech recognition capabilities.

When we talk to people who have used the system, our impression is that there are three main problems. First, recognition is currently too unforgiving; students often feel that they have pronounced the response adequately, but the system has still rejected them. This agrees well with comments we have seen from other people building spoken CALL systems, who generally report users as preferring false positives to false negatives. It is probably advisable to move the balance in this direction.

Second, there is not enough content; enthusiastic students (usually something between a quarter and a half of the whole group) finish all the existing lesson modules, and say they would happily do more if it was there. Without implementing more content, it is impossible to

determine how useful the tool could be to them in the long term. The process of creating new lessons is still too slow and requires too much system expertise; as usual with grammar-based systems, some steps can only be performed by personnel who are intimately familiar with the grammar. The most feasible way to address this problem is to build up a large enough stock of examples that further content can be added using a copy-and-edit strategy. It is not yet clear how large the set of examples would need to be, but a reasonable guess is that it should at any rate be less than an order-of-magnitude increase compared to the current set.

Finally, the application is still experienced by many students, particularly younger ones, as boring and monotonous; it is largely due to feedback from evaluations on the 10 to 12 year old age group that we have started experimenting with script-based lessons that add a narrative flow to the learning experience. Initial results here have been very positive, and are reported in (Baur et al., 2014, Tsourakis et al., 2014).

Acknowledgements

The core CALL-SLT system was developed under funding from the Swiss National Science Foundation and the Boninchi Foundation. The web deployment architecture was developed by Paideia Inc, Palo Alto. French lesson content was developed in collaboration with the University of Bologna. We would like to thank Nuance for making their software available to us for research purposes.

References

Baur, C., M. Rayner, and N. Tsourakis. 2014. Using a serious game to collect a child learner speech corpus. In *Proceedings of LREC 2014*. Reykjavik, Iceland.

Bouillon, P., C. Cervini, A. Mandich, M. Rayner, and N. Tsourakis. 2011a. Speech recognition for online language learning: Connecting CALL-SLT and DALIA. In *Proceedings of the International Conference on ICT for Language Learning*. Florence, Italy.

Bouillon, P., J. Gerlach, C. Baur, C. Cervini, and R.B. Gasser. 2012. Intégration d'un jeu de traduction orale sur le web pour l'apprentissage d'une langue seconde. In *Proceedings of TICE 2012*. Lyon, France.

Bouillon, P., M. Rayner, B. Novellas, M. Starlander, M. Santaholma, Y. Nakao, and N. Chatzichrisafis. 2007. Une grammaire partagée multitâche pour le traitement de la parole: application aux langues romanes. *TAL* .

Bouillon, P., M. Rayner, N. Tsourakis, and Q. Zhang. 2011b. A student-centered evaluation of a web-based spoken translation game. In *Proceedings of the SLaTE Workshop*. Venice, Italy.

Coyne, B., C. Schudel, M. Bitz, and J. Hirschberg. 2011. Evaluating a text-to-scene generation system as an aid to literacy. In *Speech and Language Technology in Education*.

Eskenazi, M., G.-A. Levow, H. Meng, G. Parent, and D. Suendermann, eds. 2013. *Crowdsourcing for Speech Processing: Applications to Data Collection, Transcription and Assessment*. John Wiley & Sons.

Franco, H., H. Bratt, R. Rossier, V. Rao Gadde, E. Shriberg, V. Abrash, and K. Precoda. 2010. Eduspeak®: A speech recognition and pronunciation scoring toolkit for computer-aided language learning applications. *Language Testing* 27(3):401.

Fuchs, M., N. Tsourakis, and M. Rayner. 2012. A scalable architecture for web deployment of spoken dialogue systems. In *Proceedings of LREC 2012*. Istanbul, Turkey.

Gruenstein, A., I. McGraw, and I. Badr. 2008. The WAMI toolkit for developing, deploying, and evaluating web-accessible multimodal interfaces. In *Proceedings of the 10th international conference on Multimodal interfaces*, pages 141–148. ACM.

Johnson, W.L. and A. Valente. 2009. Tactical Language and Culture Training Systems: using AI to teach foreign languages and cultures. *AI Magazine* 30(2):72.

Jurčiček, F., S. Keizer, M. Gašić, F. Mairesse, B. Thomson, K. Yu, and S. Young. 2011. Real user evaluation of spoken dialogue systems using Amazon Mechanical Turk. In *Proceedings of Interspeech 2011*. Florence, Italy.

Kennedy, C. H. 2005. *Single-Case Designs for Educational Research*. Allyn and Bacon.

Korsah, G.A., J. Mostow, M.B. Dias, T.M. Sweet, S.M. Belousov, M.F. Dias, and H. Gong. 2010. Improving child literacy in Africa: Experiments with an automated reading tutor. *Information Technologies and International Development* 6(2):1–19.

Kulik, J., C.Kulik, and P. Cohen. 1980. Effectiveness of computer-based college teaching : A meta-analysis of findings. *Review of Educational Research* 50:177–190.

Poulsen, R. 2004. *Tutoring Bilingual Students With an Automated Reading Tutor That Listens: Results of a Two-Month Pilot Study*. DePaul University, Chicago, IL: Masters Thesis.

Rayner, M., P. Bouillon, and J. Gerlach. 2012a. Evaluating appropriateness of system responses in a spoken CALL game. In *Proceedings of LREC 2012*. Istanbul, Turkey.

Rayner, M., P. Bouillon, B.A. Hockey, and Y. Nakao. 2008. Almost flat functional semantics for speech translation. In *Proceedings of COLING-2008*. Manchester, England.

Rayner, M., P. Bouillon, N. Tsourakis, J. Gerlach, M. Georgescul, Y. Nakao, and C. Baur. 2010. A multilingual CALL game based on speech translation. In *Proceedings of LREC 2010*. Valetta, Malta.

Rayner, M., I. Frank, C. Chua, N. Tsourakis, and P. Bouillon. 2011. For a fistful of dollars: Using crowd-sourcing to evaluate a spoken language CALL application. In *Proceedings of the SLaTE Workshop*. Venice, Italy.

Rayner, M., J. Gerlach, M. Starlander, N. Tsourakis, A. Kruckenberg, R. Eklund, A. Jönsson, and A. McAllister. 2012b. A web-deployed Swedish spoken CALL system based on a large shared English/Swedish feature grammar. In *Proceedings of the SLTC Workshop on NLP for CALL*. Lund, Sweden.

Rayner, M., B.A. Hockey, and P. Bouillon. 2006. *Putting Linguistics into Speech Recognition: The Regulus Grammar Compiler*. Chicago: CSLI Press.

Reeder, K., J. Shapiro, and J. Wakefield. 2007. The effectiveness of speech recognition technology in promoting reading proficiency and attitudes for Canadian immigrant children. In *15th European Conference on Reading*.

Stent, A., J. Dowding, J. Gawron, E. Bratt, and R. Moore. 1999. The CommandTalk spoken dialogue system. In *Proceedings of the Thirty-Seventh Annual Meeting of the Association for Computational Linguistics*, pages 183–190.

Tsourakis, N., M. Rayner, and C. Baur. 2014. Formative feedback in an interactive spoken CALL system. In *Proceedings of The Second Workshop on Child-Computer Interaction*. Singapore.

Wang, C. and S. Seneff. 2007. Automatic assessment of student translations for foreign language tutoring. In *Proceedings of NAACL/HLT 2007*. Rochester, NY.