A Shared Task for Spoken CALL?

Claudia Baur[1], Johanna Gerlach[1], Manny Rayner[1], Martin Russell[2], Helmer Strik[3]
(1)FTI/TIM, University of Geneva, Switzerland
(2) Department of Electronic, Electrical and Systems Engineering, University of Birmingham
(3) Centre for Language Studies (CLS), Radboud University Nijmegen
Claudia.Baur@unige.ch, Johanna.Gerlach@unige.ch, Emmanuel.Rayner@unige.ch
m.j.russell@bham.ac.uk, w.strik@let.ru.nl

Abstract

We argue that the field of spoken CALL needs a shared task in order to facilitate comparisons between different groups and methodologies, and describe a concrete example of such a task, based on data collected from a speech-enabled online tool which has been used to help young Swiss German teens practise skills in English conversation. Items are prompt-response pairs, where the prompt is a piece of German text and the response is a recorded English audio file. The task is to label pairs as "accept" or "reject", accepting responses which are grammatically and linguistically correct to match a set of hidden gold standard answers as closely as possible. Initial resources are provided so that a scratch system can be constructed with a minimal investment of effort, and in particular without necessarily using a speech recogniser. Training data for the task will be released in June 2016, and test data in January 2017.

Keywords: CALL, shared tasks, speech recognition, metrics

1. Introduction

The history of human language technology shows that the introduction of a shared task[1] often has a positive effect. Friendly competition motivates people, and the ability to make direct comparisons between different approaches to solving the same problem makes it easier to identify the ideas that work, so that effort can be focused more productively. A prominent series of examples are the various tasks based on the Wall Street Journal corpus, including speech recognition (Bahl et al., 1995), parsing (Riezler et al., 2002) and several types of semantic analysis (Pradhan et al., 2007). Perhaps even more importantly, work on machine learning during the 21st century has to a considerable extent been driven by the handwritten digit recognition task (Goodfellow et al., 2016). Other well-known examples of shared tasks include ATIS in the early 90s (Zue et al., 1994), which had a strong effect on interactive spoken language systems; the Named Entity Recognition task (Tjong Kim Sang and De Meulder, 2003), which similarly influenced work on information extraction; and the Recognizing Textual Entailment task (Dagan et al., 2006), which has influenced work on question answering.

In all these cases, introduction of the shared task created a new community with frequent productive interactions between many groups, and substantially advanced a whole subfield inside the space of a few years. The sociology of the process has become familiar to many researchers. A shared task forces each group to look closely at what other groups are doing, and in particular to study methods which are achieving high scores in the competitions. It encourages development of a common vocabulary of concepts. Above all, it introduces widely accepted evaluation procedures and metrics that permit objective comparisons, both between systems developed by different groups and between different versions of single systems. It is easier to achieve progress when people agree on what "progress" consists of, and how it can be measured.

[1] Another common term is "competitive-collaborative task".

As the series of 'Speech and Language Technology in Education' (SLaTE) workshops[2] testifies, speech recognition for CALL has become an established field. The purpose of this paper is to suggest that it has now reached the point where a shared task might be useful. We propose a task of this kind, which we will be making available as a challenge shortly after the LREC 2016 conference. For concreteness, we describe a specific instantiation, but we welcome suggestions about minor changes to the format.

2. A Shared Task for Spoken CALL

One of the most common types of spoken CALL exercise is prompt-response: the system gives the student a prompt, the student responds, and the system either accepts or rejects the response, possibly giving some extra feedback. The prompt can be of various forms, including L2 text ("read the following sentence"), L1 text ("translate the following sentence into the L2"), multimedia ("name this object") or some kind of combination. Prompt-response exercises are for example used heavily in the popular Duolingo application.[3]

We propose a minimal spoken prompt-response task based on data collected from CALL-SLT (Rayner et al., 2010), a spoken CALL system which has been under development at Geneva University since 2009[4]. The prompt is a piece of text; the response is a recorded audio file; the task is to accept linguistically correct responses, and reject others. In §2.1., we briefly sketch CALL-SLT and the data that has been collected using it; next, in §2.2., we introduce and motivate the task in intuitive terms. The rest of the paper describes the task in more detail.

2.1. CALL-SLT

CALL-SLT is an online CALL tool based on speech recognition, web and language processing technology. In the ver-

[2] http://hstrik.ruhosting.nl/slate/
[3] https://www.duolingo.com/
[4] http://callslt.unige.ch/demos-and-resources/

sion used to collect the data for the proposed shared task[5], each prompt is a combination of a multimedia file in the L2 (here, English) and a written text instruction in the L1 (here, German). To give a typical example, the system plays a short animated clip with an English native speaker asking the question, "How many nights would you like to stay at our hotel?" and simultaneously displays the German text, "Frag: Zimmer für 3 Nächte" (Ask: room for 3 nights). The text indicates how the student is supposed to answer in the L2. In this case, an acceptable response would be something like "I want a room for three nights", "Do you have a room for three nights?" or "I would like to stay for three nights". The intention is that a reasonably wide variety of grammatically and linguistically correct utterances are accepted, as long as they correspond to the meaning of the German prompt, so the student is able to practise spontaneous generative language skills. A response can be rejected for a variety of reasons, including incorrect use of vocabulary, grammatical incorrectness, incorrect use of the user interface, bad pronunciation, bad recognition due to insufficient recording quality, etc.

Once the student has answered, by speaking into the headset, the system performs speech recognition and then matches the recognised utterance against the prompt's specification of what should be counted as a correct answer. If there is a match, the system gives positive feedback by displaying a green frame around the text prompt, and moves on to the next dialogue state. If the utterance is rejected, a red frame (negative feedback) is shown and the student is asked to repeat or reformulate their response. The screenshot in figure 1 illustrates the process.

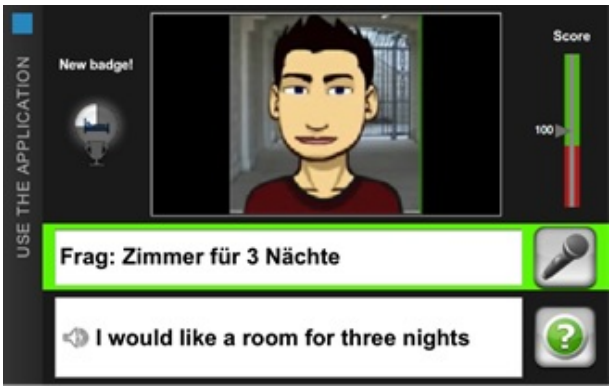

Figure 1: CALL-SLT interface.

The data was collected using an English course developed for German-speaking Swiss teenagers doing their first to third year of English (Baur et al., 2013); the course is based on a textbook commonly used in German-speaking Switzerland and consists of eight lessons ((1) at the train station, (2) getting to know someone, (3) at the tube station, (4) at the hotel, (5) shopping for clothes (6) at the restaurant, (7) at the tourist information office, (8) asking/giving directions). Each lesson offers an interactive dialogue permitting many variations, which allows the students to practise their oral conversational skills. The course focuses on a communicative approach to second language acquisition, putting more weight on achieving a successful interaction than on small grammatical or pronunciation flaws in the utterances. Corpus data has been logged in the form of prompt-response pairs, which have been annotated to specify the correctness or incorrectness of the student's response along the dimensions of grammar, vocabulary, pronunciation and fluency (Baur, 2015).

[5] http://www.issco.unige.ch/en/research/projects/callslt/content/production/english_course/english_course.html

2.2. Overview of the CALL-SLT task

The task we propose is to simulate the ideal behavior of the CALL-SLT system on logged data, with results scored against a gold standard. Each item in the test-set is a pair consisting of a text prompt and a recorded audio file. The pair is to be labelled as either "accept" (the audio file represents a linguistically correct response to the text prompt), or "reject" (it does not). A few examples will clarify the nature of the challenges involved.

Let us assume, to keep things simple, that the system which performs the labelling consists of three components: a speech recogniser, which converts an audio file into a text string; a grammar, which lists possible responses for each prompt; and a matcher, which compares the text string with the items that the grammar associates with the current prompt. Continuing the example above, suppose that the prompt is, again, "Frag: Zimmer für 3 Nächte", which the grammar associates with the three possible responses "I would like a room for three nights", "I want a room for three nights", "A room for three nights"[6]. We now consider some specific cases.

- The speech in the audio file is the words "A room for three nights"; the recogniser gets all the words right; the string is in the grammar. Evidently this is an accept.
- The speech in the audio file is the words "I don't understand"; the recogniser gets all the words right; they do not resemble anything in the grammar. Evidently this is a reject.

Unfortunately, things are not always so simple, as the next few cases show:

- The speech in the audio file is the words "I want room for three nights". The recogniser, however, produces the string "I want **a** room for three nights" — the language model predisposes it towards expecting an article in this position, and a reduced "a" is hard to hear. The system matches the string with the grammar and produces a false accept. This isn't terrible, but it will be more helpful if the system rejects, pushing the student towards a better understanding of how to use indefinite articles.
- The speech in the audio file is the words "A room for three nights please"; the recogniser gets all the words right; the string is not in the grammar. If the system

[6] A realistic grammar would of course be much larger.

is using a simple-minded matching method, it will incorrectly reject because the grammar was incomplete. This is bad, since the student is being given misleading feedback which may discourage them from using politeness phrases.

- The student, who is teasing the system, says "A broom for free fights", but the system misrecognises this as "A room for three nights", perhaps because its language model weight is set too high, and incorrectly accepts. This is catastrophic. The student will probably carry on teasing the machine rather than trying to learn from the exercise.

These examples suggest a few immediate conclusions:

- It is straightforward to develop a system which usually gets things right in the easy cases (well-pronounced correct response/incorrect response not close to any correct response).
- It is challenging to write a system which has a low error rate for the difficult cases, where the response is close to the dividing line between correct and incorrect. Unfortunately, these cases are often the most pedagogically important ones.
- Some incorrect system decisions are more serious than others.

3. Corpus and Other Resources

The core resource for the task proposed here is an English speech corpus collected with the CALL-SLT dialogue game. In total, the corpus contains 38,771 spontaneous speech acts in the form of students' interactions with the dialogue system. The data was collected in 15 school classes at 7 different schools in Germanophone Switzerland during a series of experiments in 2014 and early 2015. All interactions are logged and contain the following information: (1) subject ID, (2) prompt, (3) link to recorded file, (4) transcription, (5) whether help was accessed, (6) whether the student's response was accepted by the system. In addition, human annotators judge each interaction on various factors in order to determine whether or not the utterance should have been accepted by the system.
As described below, a subset of this information is released as data for the shared task.

3.1. Training and test corpus

For the proposed shared task, we will make available a subset of the corpus that has been annotated by three native English speakers. All interactions have been annotated on their linguistic correctness and on their appropriateness given the initial prompt. For linguistic correctness, both vocabulary and grammar are annotated on a 2-point scale, indicating whether they are judged correct or incorrect. The third annotation criterion specifies whether the answer is meaningful or not in the context of the provided prompt. This category is also annotated on a 2-point scale, labelling an utterance as "sense" or "nonsense". Accepted "nonsense" utterances will be more heavily penalised, as discussed at the end of §4.1.. Table 1 gives some examples of annotated utterances.
The training corpus contains 5,000 utterances and the test corpus will contain 1,000 utterances. The utterances in the training and test data sets are selected based on the following criteria with decreasing level of importance: 1) student's total number of interactions, 2) pre-placement test score, 3) gender, 4) age. This methodology allows us to have a representative selection of interactions in both the training and test corpora. The two data sets will contain utterances from motivated and less motivated students, from stronger and weaker students, from both male and female students and from students with different ages (ranging between 12 to 15 years). To make the data set more interesting and challenging, short utterances such as "hello", "bye", "yes", "no" and "thanks", which occur very frequently in the corpus and are almost always well pronounced by the subjects, have been dispreferred.

3.2. Other resources

In order to make it easy for groups to attempt the proposed task, we provide a number of other resources. For people who want to experiment with recognition methods, we include acoustic models, language models and scripts for Kaldi (Povey et al., 2011), a state-of-the-art open-source recogniser platform. This material, together with the accompanying documentation, is enough to permit easy construction of a baseline recogniser for British English. The acoustic models and Kaldi scripts are the ones described in (Najafian, 2016). The models have been trained on native accented British English from the Accents of the British Isles (ABI-1) corpus (D'Arcy et al., 2004) and the training part of WSJCAM0 (Robinson et al., 1994). They deliver good performance on a range of accented British English speech, and are expected to perform reasonably well on the current L2 English data. A basic bigram language model, trained on the task data, is included. It is obvious that both the acoustic and language models can be greatly improved, but they give a reasonable starting point for work. For the benefit of groups that only wish to explore the language processing aspects of the task, we will process test and training data through the baseline Kaldi recogniser and include the recognition results in the task metadata (cf. §4.2.)
We also provide a version of the existing CALL-SLT response grammar, which contains 564 prompts with a total of 11,776 possible responses. The grammar is supplied in a minimal XML format, where each item consists of the original German text prompt, an English translation of the prompt, and a list of possible responses. A typical record from the grammar is shown in Figure 2. It is important to note that the response grammar **is not intended to be exhaustive**. The task is open-ended; ideally, the system should accept any grammatically correct, adequately pronounced response which corresponds to the prompt, and the grammar only gives plausible examples of such responses. Since the grammar was automatically derived from the one used to perform the actual data collection, we know that it gives useful coverage, but it can evidently be improved.
In §5., we suggest some concrete ways to use the above resources.

System prompt	Student's response	Vocab	Grammar	Nonsensical
Frag: Zimmer für 6 Nächte	I would like a room for six nights	correct	correct	sense
Frag: Zimmer für 6 Nächte	I wants a room for six nights	correct	incorrect	sense
Frag: Zimmer für 6 Nächte	I want a room for five nights	incorrect	correct	sense
Frag: Zimmer für 6 Nächte	It's raining outside	incorrect	correct	nonsense

Table 1: Annotation examples

```
<prompt_unit>
   <prompt>Frag : Wie viel kostet es ?</prompt>
   <translatedprompt>Ask: How much does it cost?</translatedprompt>
   <response>how much does it cost</response>
   <response>how much does this cost</response>
   <response>how much is it</response>
   <response>how much is this</response>
</prompt_unit>
```

Figure 2: XML reference grammar example.

4. Concrete Structure of the Task

The abstract structure of the task is the same as it is for virtually all shared tasks. A scoring metric determines the measure to be optimised, which constitutes the task. At date-1, a quantity of training data will be made available to groups interested in participating, together with other resources. At date-2, a quantity of test data will be released to the same groups. At date-3, the participants will return the test data with the answers their software system provides. This will be scored against gold standard answers, according to the scoring metric. At date-4, the results will be released. The four time points date-1 to date-4 are defined by the task schedule. In the rest of this section, we specify our current plans for instantiating the metric, data, resources and schedule, which will be finalised based on feedback received during and shortly after the LREC 2016 conference. We now describe each component in turn.

4.1. Metric

Since no generally accepted metric appears to exist for this kind of task, we will spend some time discussing the options available and motivating the choice we have settled on. Going back to first principles, a prompt/spoken response CALL system like the one we are considering here is useful for two main reasons. The first is simply to encourage the student to practise speaking; the second is to give them accurate feedback on the correctness of their language. The second goal is the one that we wish to measure quantitatively, but the first is more important — if the students are discouraged from talking, there will be nothing to measure. Experience shows that it is essential for the system not to reject too many of the student's correct responses; if it does so, they will often give up. Ideally, the system should also fail to accept incorrect responses, but this is less critical.

Next, we consider the abstract nature of the metric. As already noted, its task is to assess the accuracy of the system's feedback, and there are two fundamental intuitions on which it can be based. The first is error rate: the system should make the accept/reject decision correctly as often as possible. The second is differential response: the system's response to correct answers should be as different as possible from its response to incorrect answers. Obviously, the two intuitions overlap to a considerable extent, but it is important to note that they can sometimes give divergent measurements. The divergence between the two intuitions is highlighted when we consider the score obtained by a dummy system which always accepts. If a high proportion of the student responses are correct, the dummy system's error rate will be fairly good; but since correct and incorrect answers yield the same result, its differential response score will be the minimal one.

There is evidently a wide range of possible metrics, and we will concentrate on several examples that the various authors of this paper have used before, where we are familiar with the issues at stake. To make different candidate metrics easy to compare, we will define them in a uniform manner. Following (Kanters et al., 2009), we assume that we are given a set of annotated prompt/response interactions, where in each case the annotations show whether the response was correct or incorrect, and whether it was accepted or rejected. We write CA for the number of of correct accepts, CR for the number of correct rejects, FA for the number of false accepts and FR for the number of false rejects. It will be convenient to set

$$Z = CA + CR + FA + FR$$

then write $C_A = \frac{CA}{Z}$, $C_R = \frac{CR}{Z}$, $F_A = \frac{FA}{Z}$, $F_R = \frac{FR}{Z}$ and define our metrics in terms of the four quantities C_A, C_R, F_A, F_R, which total to unity. In particular, we consider precision:

$$P = \frac{C_A}{C_A + F_A}$$

recall:

$$R = \frac{C_A}{C_A + F_R}$$

F-measure:

$$F = \frac{2PR}{P + R}$$

System	C_A	C_R	F_A	F_R	SA	P	R	F	D
(Kanters et al., 2009)									
(Baseline)	57.8	0.0	42.2	0.0	57.8	57.8	100.0	73.3	1.00
CGN-test	40.3	41.2	8.5	9.8	81.7	82.5	80.4	81.5	4.24
Dutch-CAPT	49.7	31.8	10.4	8.1	81.5	82.7	86.0	84.3	5.38
Dutch-CAPT (optimised)	51.6	36.0	6.3	6.2	87.6	89.2	89.4	89.3	7.93
(Rayner et al., 2015)									
(Baseline)	75.3	0.0	24.7	0.0	75.3	75.3	100.0	85.9	1.00
Plain	65.7	14.6	10.1	9.6	80.3	86.7	87.2	87.5	4.66
Minimal training	67.3	13.7	11.0	8.0	81.0	86.0	89.4	87.7	5.24
Full training	67.7	14.0	10.7	7.6	81.7	86.3	89.9	88.1	5.59

Table 2: Systems from (Kanters et al., 2009) and (Rayner et al., 2015) + baseline "always accept" systems, with values for different metrics. C_A = correct accept, C_R = correct reject, F_A = false accept, F_R = false reject, SA = scoring accuracy, P = precision, R = recall, F = F-measure, D = differential response metric.

and scoring accuracy:

$$SA = C_A + C_R$$

Scoring accuracy SA is related to classification error E by the equation $SA = 1-E$, and maximising SA is equivalent to minimising E; in general, all of these metrics are based on the idea of minimising some kind of error. In contrast, a metric based on differential response is defined in (Rayner et al., 2015). This is the ratio of the relative correct reject rate to the relative false reject rate:

$$D = \frac{C_R/(C_R + F_A)}{F_R/(F_R + C_A)} = \frac{C_R(F_R + C_A)}{F_R(C_R + F_A)}$$

In order to assess the appropriateness of the various metrics for the proposed task, we consider whether they give us results in line with our intuitive feelings about the worth of different prompt/response systems. We can immediately rule out some metrics just by considering the result they give for dummy systems. R cannot be a good metric, because it gives a maximal value to the dummy system which accepts everything. Similarly, P is unlikely to be a good metric either, since the system's best strategy for maximising it is to reject almost everything, accepting only the examples which appear most certain to be correct. The SA, F and D metrics take into account both precision and recall, so are reasonable candidates.

A problem with F and SA is that they treat false positives and false negatives symmetrically. As noted above, this does not accord with experience, since useful systems require a lower threshold for F_R than for F_A. For this reason, (Kanters et al., 2009) do not optimise SA directly, but rather optimise it subject to the restriction $F_R < 10\%$.

Table 2 lists values for the above metrics on the three Dutch pronunciation-training systems described in (Kanters et al., 2009) and three of the four versions of CALL-SLT described in (Rayner et al., 2015), together with baseline systems that always accept.[7] In each case, the first of the real systems is intuitively worst and the third best, with the second somewhere in between. Examining the different columns, F and D are both plausible metrics for the Dutch systems and capture the intuitive ranking. For the Swiss systems, however, only D clearly has this property.

The scores for the baseline "always accept" systems suggest a reason for the differences between the two groups of systems. For the Dutch systems, only 57.8% of the responses are correct, while the corresponding figure for the Swiss systems — the ones from which the data for the prospective task will be taken — is the much higher value of 75.3%. Since the baseline score on the F metric is harder to beat, its value is correspondingly less informative. However, the D metric, which measures discriminative ability rather than error rate, works equally well for both groups of systems.

If one wishes to defend the F metric, one can argue that it does indeed put the different versions of the Swiss system in the right order, even though the separation is very narrow. By slightly adjusting the numerical parameters, it is, however, apparent that F is fragile for this data. For example, if we change the proportion of correct student responses from 75.3% to 80% and keep the relative frequencies of correct and incorrect rejects the same, the F metric's score for the "always accept" baseline system overtakes that for the "plain" version of the system; if it further increases to 82%, it overtakes all three versions. This is a counter-intuitive result, since it would suggest that the system is less useful to students producing higher proportions of correct responses; in fact, the results presented in chapter 7 of (Baur, 2015) suggest the opposite pattern. In contrast, the D metric returns the same value irrespective of the balance between correct and incorrect answers, as long as the relative reject rate on each group stays the same.

We consequently suggest that the D metric is the most appropriate one for the proposed task. A straightforward refinement is to distinguish between "incorrect" and "grossly incorrect" responses, weighting the "grossly incorrect accepts" k times more heavily. We can do this by replacing the quantity FA, the number of false accepts, with the two quantities FA_1 (the number of normal false accepts), and FA_2 (the number of grossly incorrect false accepts). We

[7] The D metric is technically not defined for the baseline system, since C_R and F_R are both equal to zero. However, if we consider the baseline system to be the limit as $\varepsilon \to 0$ of a system which randomly rejects with probability ε, we obtain an intuitively reasonable value of 1.0.

then change the definitions slightly to set

$$Z = CA + CR + FA_1 + k.FA_2 + FR$$

and

$$F_A = \frac{FA_1 + k.FA_2}{Z}$$

keeping everything else the same; the construction can obviously be generalised to allow weighted subdivision of other categories too.

4.2. Resources

The following material will be made available on June 15, 2016, packaged as a zipfile that can be downloaded from `http://callslt.unige.ch/demos-and-resources/`:

1. 5,000 recorded audio files.
2. A metadata file consisting of a five-column CSV spreadsheet, where the first four columns are respectively a prompt, a link to the audio file, the transcription, and an accept/reject annotation, the annotations carried out according to the protocol described in §3.1. above. The final column gives a recognition result produced using the baseline Kaldi recogniser described in §3.2., and is intended for use by groups who only wish to attempt the language processing aspects of the task.
3. Speech and language resources, described in §3.2., that may be useful for groups who intend to compete in the task.

On January 15, 2017, the test data will be made available at the same URL. This will consist of 1,000 utterances of test data, in the same format as the training data but with the "accept/reject" column of the five-column metadata file left blank. All the utterances included in the test set will have been annotated by at least three judges, and will be restricted to examples where the judges' annotations are unanimous.

4.3. Scoring platform

A web platform will allow participants to check their results against the gold standard data by uploading a spreadsheet with their accept/reject results for each prompt/response pair. The platform will compute the score, as well as individual results. This process will be available without limitations for the training data, thereby allowing participants to check progress of their score as well as to test the submission mechanism. For final submission of the test data results, each participant will be allowed only one submission.

4.4. Schedule

- The training material and other resources defined below in §4.3 will be released on June 15, 2016. The date is chosen to allow consultation about the exact form of the task during and shortly after the LREC 2016 conference.
- The test material as defined below in §4.2. will be released on January 15, 2017.
- Participating groups will have one week, i.e. until January 22, 2017, to process the data through their systems and upload the results, in spreadsheet form.
- If enough groups take part, a special session will be organised at the next SLaTE workshop, a satellite of Interspeech 2017 in Stockholm. Papers describing implemented systems will be due at the SLaTE workshop deadline, provisionally fixed for March 30, 2017. Scores for all systems will be published at the workshop.

5. Why is this a worthwhile task?

A good shared task should be a) relevant to the community, b) accessible to a large number of groups c) clearly defined, d) not too hard and e) not too easy. We discuss these points in turn.

Relevant to the community: Prompt-response exercises are widely regarded as important, and developing systems which perform well on this task is of more than academic interest; as noted, many of the spoken language generation exercises on Duolingo are of the same basic form. A substantial improvement in response accuracy would make CALL platforms of this kind far more useful.

Accessible to a large number of groups: The main problem is that the task inherently favors groups with expertise in speech recognition. We have done our best to level the playing field by adding the resources from §3.2. to the distribution, including recognition results from the baseline recogniser.

Clearly defined: Inter-annotator agreement is good enough that we do not think this will be a problem. We have a simple domain, and it is usually obvious whether a response is linguistically correct or not.

Not too hard: It is trivial to put together a scratch system and get started. A minimal baseline system can literally consist of a couple of dozen lines of Python: all that is necessary is to read the CSV metadata file and the XML reference grammar, then check whether the recognition result in the last column of the CSV file matches one of the responses in the relevant record of the XML grammar.

Not too easy: It is easy to get a basic system working, but, based on our own experience, it is very challenging to build a system which is anywhere close to doing what teachers actually want: accept all correct utterances and reject all incorrect ones. If the utterance is correct except for a small grammatical error (missing article, singular/plural mismatch, incorrect choice of preposition), it will often be accepted. In the other direction, many correct responses not within implemented grammar coverage will be rejected.

Following on from the last point, there is a great deal of scope for improving the original system, which is what makes the challenge interesting. Some obvious possibilities include the following:

Creating better response grammars: This is the idea explored in (Rayner et al., 2015), which describes an initial concrete example of performing the task: we developed a simple machine learning algorithm which used the annotated data to expand the existing response grammar. The method yielded a 20% relative improvement on the D metric from §4.1. above.

Performing more intelligent matching: Another obvious approach is to keep the response grammar as it is, and use machine learning methods to create a better way of matching recogniser output against the existing set of allowed responses.

Creating better language models: The Kaldi resources described in §3.2. only include a minimal bigram language model. The easiest way to improve the baseline system's recognition performance is to replace this with a more sophisticated model.

Creating better acoustic models: Yet another obvious way to improve recognition performance is to use the audio files in the training data to tune the Kaldi acoustic models more closely to the peculiarities of English as spoken by young Swiss German teens. Other freely available speech corpus resources can potentially also be used for this purpose.

6. Summary and Further Directions

We have proposed an initial shared task for spoken CALL, which is being made available to the community in June 2016. It is intentionally very simple. Since no such task currently exists, it seemed advisable to start with something straightforward, where annotation criteria are uncontroversial and it is possible to build a scratch system with an effort measured in person-days. If the task proves successful, in terms of being attempted by a reasonable number of groups, there are obvious directions in which it could be extended. Perhaps the most important of these is to make the criteria for acceptance and rejection relative to pronunciation quality.

We hope that groups working with CALL and speech recognition will consider attempting our task; if people do not find this idea interesting enough, we at least hope our proposal will encourage development of a better one. It's time to go mainstream.

7. Acknowledgements

Work at Geneva University was supported by the Swiss National Science Foundation (SNF) under grant 105219_153278/1.

We would like to thank Nuance for making their software available to us for research purposes, and Cathy Chua for helpful suggestions concerning the metric.

8. Bibliographical References

Bahl, L. R., Balakrishnan-Aiyer, S., Bellgarda, J., Franz, M., Gopalakrishnan, P., Nahamoo, D., Novak, M., Padmanabhan, M., Picheny, M. A., and Roukos, S. (1995). Performance of the IBM large vocabulary continuous speech recognition system on the ARPA Wall Street Journal task. In *Proceedings of ICASSP 1995*, pages 41–44. IEEE.

Baur, C., Rayner, M., and Tsourakis, N. (2013). A textbook-based serious game for practising spoken language. In *Proceedings of ICERI 2013*, Seville, Spain.

Baur, C. (2015). *The Potential of Interactive Speech-Enabled CALL in the Swiss Education System: A Large-Scale Experiment on the Basis of English CALL-SLT*. Ph.D. thesis, University of Geneva.

Dagan, I., Glickman, O., and Magnini, B. (2006). The PASCAL recognising textual entailment challenge. In *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*, pages 177–190. Springer.

D'Arcy, S. M., Russell, M. J., Browning, S. R., and Tomlinson, M. J. (2004). The accents of the British Isles (ABI) corpus. *Proceedings Modélisations pour l'Identification des Langues*, pages 115–119.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep learning. Book in preparation for MIT Press.

Kanters, S., Cucchiarini, C., and Strik, H. (2009). The goodness of pronunciation algorithm: a detailed performance study. *SLaTE*, 2009:2–5.

Najafian, M. (2016). *Acoustic model selection for recognition of regional accented speech*. Ph.D. thesis, University of Birmingham.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., et al. (2011). The Kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, number EPFL-CONF-192584. IEEE Signal Processing Society.

Pradhan, S. S., Loper, E., Dligach, D., and Palmer, M. (2007). Semeval-2007 task 17: English lexical sample, SRL and all words. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 87–92. Association for Computational Linguistics.

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

Rayner, M., Baur, C., Chua, C., and Tsourakis, N. (2015). Supervised learning of response grammars in a spoken CALL system. In *Proceedings of the Sixth SLaTE Workshop*, Leipzig, Germany.

Riezler, S., King, T., Kaplan, R., Crouch, R., Maxwell, J., and Johnson, M. (2002). Parsing the wall street journal using a lexical-functional grammar and discriminative estimation techniques. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (demo track)*, Philadelphia, PA.

Robinson, T., Fransen, J., Pye, D., Foote, J., and Renals, S. (1994). WSJ-CAM0: A British English corpus for large vocabulary continuous speech recognition. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*.

Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the CoNLL-2003 shared task: Language-

independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147. Association for Computational Linguistics.

Zue, V., Seneff, S., Polifroni, J., Phillips, M., Pao, C., Goddeau, D., Glass, J., and Brill, E. (1994). Pegasus: A spoken language interface for on-line air travel planning. In *Proceedings of the workshop on Human Language Technology*, pages 201–206. Association for Computational Linguistics.