Utterance Verification in Language Learning Applications

Joost van Doremalen, Helmer Strik, Catia Cucchiarini

Department of Linguistics, Radboud University, Nijmegen, The Netherlands
{j.vandoremalen,h.strik,c.cucchiarini}@let.ru.nl

Abstract

A CALL system for oral proficiency is being developed in which constrained responses are elicited from L2 learners. In the first phase the best matching utterance is selected from a predefined list of possible responses. Since errors may occur and giving feedback on the basis of incorrectly recognized utterances is confusing, we verify the correctness of the utterance in the second phase. In the current paper we focus on the utterance verification process. Combining duration related features with a likelihood ratio (LR) yielded an equal error rate (EER) of 10.3%, which was significantly better than the EER for LR alone, 14.4%, and the EER for the duration-related features, 25.3%

Index Terms: utterance verification, non-native speech processing, computer-assisted language learning

1. Introduction

In second language acquisition research it is widely acknowledged that naturalistic, implicit learning is not always sufficient to achieve high-quality L2 proficiency and that explicit instruction helps overcome some of the problems [1] [2]. In the case of oral proficiency, providing sufficient instruction and feedback is more problematic than in other skills because time-consuming interaction with an individual tutor is usually required. This might explain the increasing interest in applying automatic speech recognition to oral proficiency learning [3]. The overview by Eskenazi [3] also makes it clear that developing good-quality ASR-based language learning applications is fraught with difficulties. One of the problems concerns the relatively poor performance of ASR systems on non-native speech and the consequent need to develop approaches that restrict the search space and make the task easier. A major distinction can be drawn between strategies that are essentially aimed at constraining the output of the learner so that the speech becomes more predictable and techniques that are aimed at improving the decoding of non-native speech.

Within the first category, a possible strategy consists in eliciting constrained output from learners by letting them read aloud an utterance from a limited set of answers presented on the screen or by allowing a limited amount of freedom in formulating responses, as in the *Subarashii* [4] and and the *Let's Go* systems [5]. However, more freedom in user responses is particularly necessary in ASR-based CALL systems that are intended for practicing grammar in speaking proficiency. While for practicing pronunciation it may suffice to read sentences aloud, to practice grammar learners need to have some freedom in formulating answers in order to show whether they are able to produce correct forms. This can be achieved by designing exercises that allow some freedom to the learners in producing answers, but that are predictable enough to be handled by ASR.

In our DISCO project, which is aimed at developing a prototype of an ASR-based CALL application that can provide intelligent feedback on important aspects of L2 speaking such as pronunciation, morphology, and syntax [6], this is achieved by generating a predefined list of possible (correct and incorrect) responses for each exercise.

We intend to use a two-step procedure in which first is determined what was said (content), and subsequently how it was said (form). In the first (recognition) phase the system should tolerate deviations in the way utterances are spoken, while in the second (error detection) phase, strictness is required (see also [7] and [8]).

In the first phase of the two-step procedure two stages can be distinguished, a) utterance selection and b) utterance verification (UV). When learners are allowed some freedom in formulating their responses, there is always the possibility that the learners response is not present in the predefined list and is recognized incorrectly in stage (a) as one of the utterances of the predefined list. Giving feedback on the basis of an incorrectly recognized utterance is confusing and thus should be avoided. Therefore, utterance verification (UV) is carried out in stage (b). An excellent overview of recent work on UV can be found in [9].

In the present paper we focus on the process of verifying the decoded utterance within the framework of a CALL application for oral proficiency. In the remainder of this paper we first describe the speech material used in our experiments and subsequently the speech recognizer and the UV approach adopted in these experiments. The results are presented in section 3. In section 4 we discuss our findings and speculate on possible ways of utilizing our method for UV in the context of a CALL application like DISCO. We end with some concluding remarks in section 5.

2. Method

2.1. Material

The speech material for the present experiments was taken from the non-native component of the JASMIN speech corpus [10], which was collected for the aim of facilitating the development of ASR-based language learning applications and is particularly suited for our purpose. Speakers with different mother tongues and relatively low proficiency levels (A1, A2 and B1 of the Common European Framework) were recorded because this complies with the demand for ASR-based CALL applications. The JASMIN corpus contains read speech and human-machine dialogues. The latter were used for our experiments because they more closely resemble the situation we will encounter in the DISCO application. The JASMIN dialogues were designed such as to elicit typical phenomena of human-machine interaction that are known to be problematic in the development of spoken dialogue systems, i.e. restarts, filled pauses and repetitions.

The material we used for the present experiments consists of speech from 45 speakers, 40% male and 60% female, with 25 different L1 backgrounds. Ages range from 19 to 55, with a mean of 33. The speakers each respond to 39 questions about a journey. We first deleted the utterances that contain crosstalk, background noise and whispering from the corpus. After deletion of these utterances the material consists of 1325 utterances. The mean signal-to-noise-ratio (SNR) of the material is 24.9 with a standard deviation of 5.1.

To simulate the task in the DISCO application of selecting and verifying the utterance that was spoken, we generated language models from the lists of responses given by each speaker to each of the 39 questions. These lists mimic the predicted responses in our CALL application task because they contain a) responses to relatively closed questions and b) morphologically and syntactically correct and incorrect responses. Note that in this set the response that was spoken was always present in the language model. To simulate the case in which the spoken utterance is not present in the list, we also generated language models in which the correct utterance is left out. In this way, our dataset consists of 1650 items, because each utterance is decoded two times: one time when its representation is present in the language model and one time when it is not present.

2.2. Utterance selection

For selecting the spoken utterance from a list, we have used a speech recognizer with a constrained language model and small vocabulary. The speech recognizer we used in this research is SPRAAK [11], an open source HMM ASR package. In the following section we will discuss the setup of this speech recognizer.

2.2.1. Acoustic Preprocessing

Acoustic preprocessing was done by dividing the speech, sampled at 16kHz, into overlapping 32ms Hamming windows with a 10ms shift and pre-emphasis factor of 0.95. 12 Mel-frequency cepstral coefficients (MFCC) plus C_0, and their first and second order derivatives were calculated, and cepstral mean subtraction (CMS) was applied.

2.2.2. Language Model and Pronunciation Lexicon

Constrained language models (LM) were generated based on the responses to each of the 39 questions. These responses were manually transcribed at the orthographic level. Restarts and repetitions were also annotated. The LMs are implemented as Finite State Machines (FSM) with parallel paths containing the word sequences of the responses. A priori each path is equally likely. To be able to decode filled pauses between words, self-loops are added in every node. Filled pauses are represented in the pronunciation lexicon. The pronunciation lexicon contains canonical phonetic representations extracted from the CGN lexicon [12].

2.2.3. Acoustic Models

We trained three-state tied Gaussian Mixture Models (GMM). 47 Baseline triphone models, 46 phoneme and one silence model, were trained on 42 hours of native read speech from the CGN corpus [12]. In total 11,660 triphones were created, using 32,738 Gaussians. These native models were retrained with non-native speech by doing a one-pass Viterbi training with 6 hours of non-native read speech from the JASMIN corpus. The utterances were spoken by the same speakers as those in the test material.

Table 1: Equal error rates (EER) for the individual features *LR*, *nr_shorter_1*, *nr_shorter_5*, *nr_longer_95*, *nr_longer_99* and the combinations *duration_comb* (*nr_shorter_1*,*nr_shorter_5*,*nr_longer_95*, *nr_longer_99*) and all features, *all*.

Features	EER
LR	14.4%
nr_shorter_1	27.3%
nr_shorter_5	27.4%
nr_longer_95	35.8%
nr_longer_99	38.5%
duration_comb	25.3%
all	10.3%

2.3. Utterance verification

A common approach to utterance verification is to extract confidence predictors during decoding and combine these using a machine learning model. This model is then trained to predict whether the utterance is correctly or incorrectly recognized. Confidence predictors that are often used include N-best list counts, hypothesis density, acoustic stability and duration related features [9]. We have also adopted this confidence predictor combination approach and used two types of predictors, acoustic likelihood ratio and duration related features, to train a logistic regression model. Details on the predictors and model are provided below.

2.3.1. Acoustic likelihood ratio

The first confidence predictor, one that has been used in for example [13], is the likelihood ratio:

$$\frac{p(\mathbf{x}|u_1)}{p(\mathbf{x}|u_{FPR})} \tag{1}$$

in which u_1 is the 1-Best decoding result given the signal $\mathbf{x}$ and u_{FPR} is the optimal phone string found using free phone recognition. We call this predictor *LR*. The rationale behind this predictor is that when the input speech is not modelled as a path in the search space, the likelihood $p(\mathbf{x}|u_1)$ is smaller relative to $p(\mathbf{x}|u_{FPR})$ than when it is modelled. This predictor estimates the posterior probability of the utterance given the speech signal $\mathbf{x}$ where $p(\mathbf{x}|u_{FPR})$ is an estimation of the probability of $\mathbf{x}$.

2.3.2. Duration-related features

When the input speech representation is not modelled as a path in the search space and the utterance is recognized as another sequence of words, the phone segmentation of this sequence of words will generally be characterized by deviations in phone durations. A straightforward way to capture this is to count the phones in the segmentation with durations that deviate substantially from the mean phone duration. We have implemented this by using predictors similar to those introduced in [14]. Phone duration distributions were derived from manually verified phonemic transcriptions of 42 hours of read native speech from the CGN corpus [12]. For each of the 46 phonemes the 1st, 5th, 95th and 99th percentile duration was calculated from these distributions. The predictors that were extracted from the segmentation are the number of phonemes in the decoded utterance that are shorter than the 1st (*nr_shorter_1*) and 5th (*nr_shorter_5*) percentile and the number of phonemes that are longer than the 95th (*nr_longer_95*) and 99th (*nr_longer_99*) percentile durations. These predictors were normalized by the total number of phonemes in the recognized utterance.

Table 2: Percentages of correctly and incorrectly classified decoding results of the two different subsets and the total set using the global EER threshold and all predictors. (a) Percentages of decoding result classification on the set where the correct transcription was in the language model. b) Percentages of decoding result classification on the set where the correct transcription was not present in the language model. (c) Percentages of decoding result classification on the whole dataset.

(a)

		actual	
		correct	incorrect
predicted	correct	80.8%	3.0%
	incorrect	9.2%	7.0%

(b)

		actual	
		correct	incorrect
predicted	correct	-	8.3%
	incorrect	-	91.7%

(c)

		actual	
		correct	incorrect
predicted	correct	40.4%	5.6%
	incorrect	4.6%	49.4%

2.3.3. Feature combination

To combine the five predictors, i.e. *LR*, *nr_shorter_1*, *nr_shorter_5*, *nr_longer_95*, *nr_longer_99*, into one confidence measure we used a logistic regression model. In this model it is assumed that the logit of the probability of a binary variable is a linear function of a set of explanatory variables:

$$\mathrm{logit}(p(y|\mathbf{p})) = \frac{p(y|\mathbf{p})}{1-p(y|\mathbf{p})} = \beta_0 + \sum_{i=1}^{N} \beta_i x_i \tag{2}$$

where $p(y|\mathbf{p})$ is the probability of a correctly or incorrectly decoded utterance y given the confidence predicting variables $\mathbf{p}$. The optimal weights β are choosen through Maximum Likelihood Estimation (MLE) in the WEKA machine learning toolkit [15]. We trained and tested the model by using Leave-One-Speaker-Out crossvalidation where the model is trained on all speakers except one and then tested on the utterances of the speaker that was left out during training. This is repeated until all speakers are tested and the results of all speakers are averaged.

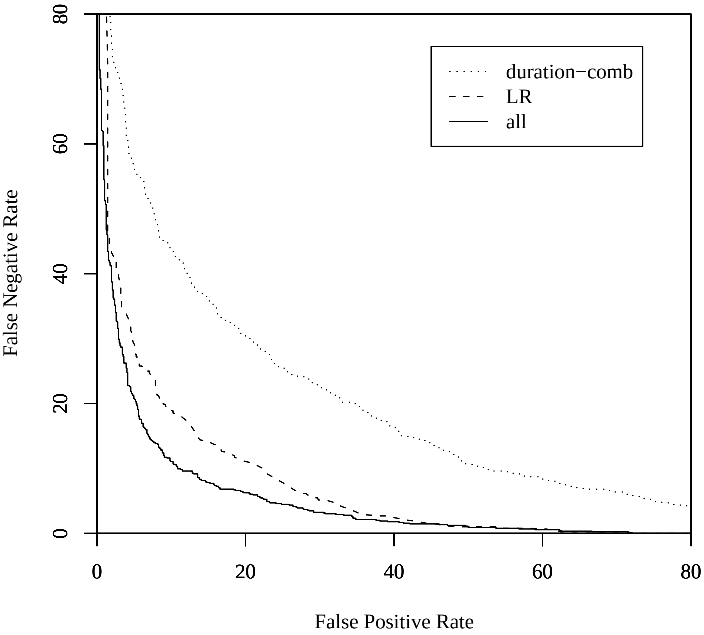
2.4. Evaluation

We have evaluated the discriminative ability of our utterance verifier using Receiver Operator Characteristic (ROC) curves, in which the two types of error rates, i.e. the false positive rate and false negative rate, are plotted for different thresholds. Using the point of the ROC curve where the two error types are equal, the equal error rate (EER), the different confidence indicators and their combinations are evaluated. 95% Confidence intervals were calculated to investigate whether differences between EERs were significantly different.

3. Results

The utterance error rate (UER) of our speech decoder on the set of decoding results where the correct transcription was present in the LM was 10.0%. In this case errors consist of substitutions with competing language model paths. The UER on the set without the correct transcriptions in the LM was of course 100.0%, so 55.0% of all the cases was incorrectly recognized.

Figure 1: ROC curves for the feature *LR* and the combinations *duration_comb* and *all*.



The task for the UV was to discriminate the correctly and incorrectly recognized cases. In Table 1 this ability is shown in terms of EER for the individual predictors and several predictor combinations. ROC curves of the best performing predictor and two combinations are shown in Figure 1.

Within the individual predictors *LR* performs best (14.4%) and all the duration-related predictors perform much worse. When we combined all duration-related predictors, *duration_comb*, the EER relative to the best performing duration-related predictor dropped significantly from 27.3% (with a confidence interval ±1.7) to 25.3%. Finally, by combining the *LR* with *duration_comb*, the EER relative to *LR* decreased significantly by 4.1% from 14.4% to 10.3%.

In Table 2a and 2b percentages are shown using the EER threshold and using all predictors for the two different sets of decoding results, with and without the correct transcription in the LM, respectively. For example, in the set of results with the correct transcription in the LM 80.8% is classified as correct when it indeed was correctly decoded and 9.2% was classified as incorrect (false reject). In the set without the correct transcription in the LM 91.7% was classified as incorrect when it was incorrectly decoded, and 8.3% was classified as correct (false accept). The performance on the whole dataset is shown in Table 2c.

4. Discussion

The duration-related predictors have a weak performance individually, but they still contain additional information relative to the acoustic likelihood ratio *LR*. The duration-related predictor distributions of correctly and incorrectly decoded utterances overlap severely. This was still the case when we normalized these predictors for the speaking rate within the utterance or when we used the probability of the phoneme durations in the utterance as a predictor. The latter we calculated through a kernel density estimation of the duration probability density per phoneme trained on the CGN native read speech data. Using these more complex predictors the model was not able to make substantially better predictions.

By introducing an UV procedure and using the EER threshold we are able to filter out 91.7% of the utterances that are not in the predicted list of responses. This comes with the cost of also rejecting utterances that are correctly decoded and accepting utterances that are incorrectly decoded. Of course, these error rates depend not only on the discriminative performance of the UV, but also on the threshold setting.

In our CALL application this threshold setting has consequences for the learner, because of the potentially misleading feedback he or she gets. Until now we have evaluated the performance of different predictors and combinations using the EER threshold, but this might not be the optimal threshold setting in the actual application.

In our application the recognized utterance will be probably shown to the user so that he/she knows whether the utterance was correctly recognized. If the system makes an error in recognizing the utterance, this will then be clear for the user. The system can make two types of errors: a) a false rejection, in which case a correctly decoded utterance is classified as incorrect by the UV or b) a false acceptance, in which case an incorrectly decoded utterance is classified as correct. To determine which of these errors is more detrimental at this stage of the application, it is necessary to consider how such errors can be handled in the application and what their possible consequences are. In the case of a rejection, and therefore also of a false rejection, it is possible to ask the user to repeat the utterance. In concrete terms then, a false rejection implies that the user is unnecessarily asked to repeat the utterance. In the case of a false acceptance an utterance will be shown to the user that (s)he actually did not produce. This type of error would seem to be more detrimental because it can affect the credibility of the system.

However, the degree of seriousness will depend on the degree of discrepancy between the utterance that was actually produced and the one that was recognized and shown by the system: the larger the deviation the more serious the error. On the other hand, large deviations are less likely than small deviations. On the basis of such considerations we can indicate the seriousness of the two types of errors and therefore the costs that should be assigned to false rejections and false acceptances. More information on this issue can be found in [16].

There are now three different factors that are important in choosing an application-dependent threshold, namely 1) the prior probability of a correct decoding $p_{correct}$, 2) the cost of a false rejection C_{FR} and 3) the cost of a false acceptance C_{FA}. To formalize the idea of taking into account different error costs and different prior distributions in the process of choosing a threshold, we can estimate the total cost of a specific threshold setting with a cost function:

$$C_{total} = p_{FR} C_{FR} p_{correct} + p_{FA} C_{FA} (1 - p_{correct}) \quad (3)$$

where p_{FR} and p_{FA} are the probabilities of false rejection and false acceptance respectively. This kind of cost function is also used in the NIST evaluation of speaker recognition systems [17]. Minimizing C_{total} on a development set will provide us with the optimal threshold setting given the application-dependent parameters C_{FR}, C_{FA} and $p_{correct}$. Using the UV with this application-dependent threshold calibration procedure will make an excellent research vehicle for future experiments with different error costs.

5. Conclusion

We have evaluated several procedures for utterance verification. The best result obtained for a single duration-related feature is an EER of 27.3%. By combining four duration-related features the EER could be reduced significantly to 25.3%. Better results, i.e. an EER of 14.4%, were found for the tested acoustic likelihood ratio, and an extra significant reduction to 10.3% was obtained by combining the likelihood ratio with the four duration-related features.

6. Acknowledgements

The DISCO project is carried out within the STEVIN programme which is funded by the Dutch and Flemish Governments (http://taalunieversum.org/taal/technologie/stevin/).

7. References

[1] Norris, J.M. and Ortega, L., “Effectiveness of L2 instruction: A research synthesis and quantitative meta-analysis”, Language Learning, vol. 50, pp. 417-528, 2000.

[2] Ellis, N.C., Bogart, P.S.H., “Speech and Language Technology in Education: the perspective from SLA research and practice”, In Proceedings ISCA ITRW SLaTE, Farmington PA, 2007.

[3] Eskenazi, M., “An overview of Spoken Language Technology for Education”, Speech Communication, 2009.

[4] Ehsani, F., Bernstein, J. and Najmi, A., “An interactive dialog system for learning Japanese”, Speech Communication, vol. 30, pp. 167-177, 2000.

[5] Raux A. and Eskenazi, M., “Using task-oriented spoken dialogue systems for language learning: potential, practical applications and challenges”, In Proceedings of INSTILL, 2004.

[6] DISCO project website, http://lands.let.ru.nl/ strik/research/DISCO/.

[7] Menzel, W., Herron, D., Morton, R., Pezzotta, D., Bonaventura, P., and Howarth, P., “Interactive pronunciation training”, ReCALL, vol. 13, no. 1, pp. 67-78, 2000.

[8] Cucchiarini, C., Neri, A., and Strik, H., “Oral proficiency training in Dutch L2: The contribution of ASR-based corrective feedback”, Speech Communication, to appear.

[9] Jiang H., “Confidence measures for speech recognition: a survey”, Speech Communication, vol. 45, pp. 455-470, 2005.

[10] Cucchiarini, C., Driesen, J., Van hamme, H. and Sanders, E., “Recording speech of children, non-natives and elderly people for HLT applications: the JASMIN-CGN Corpus”, In Proceedings of LREC, 2008.

[11] Demuynck, K., Roelens, J., Van Compernolle, D. and Wambacq, P., “SPRAAK: an open source SPeech Recognition and Automatic Annotation Kit”, In Proceedings of ICSLP, page 495, 2008.

[12] Oostdijk, N., “The design of the spoken Dutch corpus”, In Peters, P., Collins, P., and Smith, A., (Eds.) New Frontiers of Corpus Research, Rodopi, Amsterdam, pp. 105-112, 2002.

[13] Bouwman, G. and Boves, L., “Utterance verification based on the likelihood distance to alternative paths”, In Proceedings of the 5th International Conference on Text, Speech and Dialogue, pp. 213-220, 2002.

[14] Goronzy, S., Marasek, K., Kompe, R. and Haag, A., “Prosodically Motivated Features for Confidence Measures”, In ASR2000, vol. 1, pp. 207-212, 2000.

[15] Witten, I.H. and Frank, E.,“Data Mining: Practical machine learning tools and techniques”, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

[16] Bachman, L., “Fundamental considerations in language testing”, Oxford University Press, 1990, pp. 214-218.

[17] van Leeuwen, D. and Brümmer, N., “An Introduction to Application-Independent Evaluation of Speaker Recognition Systems”, In Speaker Classification I, Christian Mller (Ed.), Springer, 2007.