Optimizing non-native speech recognition for CALL applications

Joost van Doremalen, Helmer Strik, Catia Cucchiarini

Department of Language and Speech Technology, Radboud University, Nijmegen, The Netherlands

{j.vandoremalen,h.strik,c.cucchiarini}@let.ru.nl

Abstract

We are developing a Computer Assisted Language Learning (CALL) system that gives feedback to grammar and pronunciation that makes use of Automatic Speech Recognition (ASR). However, good quality unconstrained non-native ASR is not yet feasible. Therefore, we use an approach in which we try to elicit constrained responses. The task in the current experiments is to select utterances from a list of responses. The results of our experiments show that significant improvements can be obtained by optimizing the language model and acoustic models. In this way we could reduce the utterance error rate from 29-26% to 10-8%.

Index Terms: non-native speech recognition, computer-assisted language learning

1. Introduction

The increasing demand for innovative applications that support language learning has led to a growing interest in Computer Assisted Language Learning (CALL) systems that make use of Automatic Speech Recognition (ASR) technology. Such systems can address oral proficiency, one of the most problematic skills in terms of time investments and costs, and are seriously being considered as a viable alternative to teacher-fronted lessons. In the Netherlands speaking proficiency plays an important role within the framework of civic integration examinations. In this context automatic systems for improving speaking performance are particularly welcome. Such systems should preferably address important aspects of oral proficiency like pronunciation and grammar.

However, developing ASR-based CALL systems that can provide training and feedback for second language speaking is not trivial, as ASR performance on non-native speech is not yet as good as on native speech [1] [2] [3] [4]. The main problems with non-native speech concern deviations in pronunciation, morphology, and syntax and a relatively high rate of disfluencies, such as filled pauses, repetitions, restarts and repairs. To circumvent the ASR problems caused by these phenomena, some approaches have been proposed to restrict the search space and make the task easier. A major distinction can be drawn between a) strategies that are essentially aimed at constraining the output of the learner so that the speech becomes more predictable and b) techniques that are aimed at improving the decoding of non-native speech.

Within the first category, a well-known strategy consists in eliciting output from learners by letting them choose from a finite set of answers that are presented on the screen. This technique was used in the *ISLE* system [5] and *Tell me More* [6] and *Talk to Me* [7] series developed by *Auralog*. In the Auralog products the learner can engage in dialogues with the computer by answering oral questions that are simultaneously displayed on the screen and can reply by choosing one response from a set of three that are phonetically sufficiently different from each other so that the spoken response can easily be recognized by the ASR system. Although this strategy allows for relatively realistic dialogues and is still applied in language learning applications, different techniques were also explored to allow more freedom in the responses. This would mean that instead of choosing from a limited set of utterances that can be read aloud, the learner has some freedom in formulating his/her answer. This was the case in the *Subarashi* program [8] and in the *Lets Go* system [2].

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 out loud sentences, to practice grammar learners need to have some freedom in formulating answers so that they can show whether they are able to produce correct forms. So, the challenge in developing an ASR-based system for practicing oral proficiency consists in designing exercises that allow some freedom to the learners in producing answers, but that are predictable enough to be handled by ASR.

This is precisely the challenge we face in our DISCO project, which is aimed at developing a prototype of an ASR-based CALL application for practicing speaking performance in Dutch as a second language (DL2). The application aims at optimizing learning through interaction in realistic communication situations and at providing intelligent feedback on important aspects of DL2 speaking, viz. pronunciation, morphology, and syntax. Within this project we are designing exercises that stimulate students to produce utterances containing the required morphological and syntactic forms by using dialogues and displaying words on the screen, without declensions or conjugations, in random order, possibly in combination with pictograms and figures representing scenes.

In these exercises learners are prompted to produce utterances which are subsequently analyzed to detect the errors and provide the appropriate feedback. In the DISCO application we intend to adopt a two-step procedure in which first is determined what was said (content), and subsequently how it was said (form). In the first phase the system should tolerate deviations in the way utterances are spoken, while in the second phase, strictness is required (see also [5] and [9]). The first phase is necessary to establish whether the learner produced an appropriate answer. Only after the incoming utterance has been identified as being an attempt at producing the required answer, does the system proceed to carry out error detection. If the utterance cannot be recognized the system will prompt the user to try again.

In the first phase of the two-step procedure two stages can be distinguished, a) recognition and b) verification. In the present paper we will confine ourselves to the research we carried out to optimize the process of recognizing the intended utterance. In the remainder of this paper we first present the

speech material we used in our experiments. Because we do not have DISCO speech data yet, we resorted to other non-native speech material which seemed particularly suitable for our research purpose, as will be described in the Method section (2). The results of our experiments are presented in section 3. In section 4 we discuss our findings and draw some conclusions.

2. Method

2.1. Material

The speech material for the present experiments was taken from the JASMIN speech corpus [10], which contains speech of children, non-natives and elderly people. Since the non-native component of the JASMIN corpus was collected for the aim of facilitating the development of ASR-based language learning applications, it seemed particularly suited for our purpose. Speech from speakers with different mother tongues was collected, because this realistically reflects the situation in Dutch L2 classes. In addition, these speakers have relatively low proficiency levels, namely A1, A2 and B1 of the Common European Framework (CEF), because it is for these levels that ASR-based CALL applications appear to be most needed.

The JASMIN corpus contains speech collected in two different modalities: 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 collected through a Wizard-of-Oz-based platform and were designed such that the wizard was in control of the dialogue and could intervene when necessary. In addition, recognition errors were simulated and difficult questions were asked to elicit some typical phenomena of human-machine interaction that are known to be problematic in the development of spoken dialogue systems, such as hyperarticulation, restarts, filled pauses, self talk 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 give answers 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.

2.2. Speech Recognizer

The speech recognizer we used in this research is SPRAAK [11], an open source HMM ASR package. The input speech, sampled at 16kHz, is divided 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. The constrained language models and pronunciation lexicons are implemented as finite state machines (FSM).

To simulate the ASR task in the DISCO application, we generated lists of the answers 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.

2.3. Language Modelling

Our approach was to use a constrained language model (LM) to restrict the search space. In total 39 LMs were generated based on the responses to each of the 39 questions. These responses were manually transcribed at the orthographic level. Filled pauses, restarts and repetitions were also annotated.

Filled pauses are common in everyday spontaneous speech and generally do not hamper communication. It seems therefore that students using a CALL application should be allowed to make filled pauses. In our material 46% of the utterances contain one or more filled pauses and almost 13% of all transcribed units are filled pauses.

While restarts, repairs and repetitions can also occur in normal speech, albeit less frequently, we think that in a CALL application for training oral proficiency students should be stimulated to produce fluent speech. On these grounds restarts, repetitions and repairs can be penalized. In our material 11% of the answers contain one or more disfluencies. In this research we do not focus on restarts and repetitions. We included their orthographic transcriptions in the LM and their manual phonetic transcriptions in the lexicon.

The LMs are implemented as FSMs with parallel paths of orthographic transcriptions of every unique answer to the question. A priori each path is equally likely. For example, part of a response list is:

- /ik ga met uh... de vliegtuig/ (/I am going er... by plane/*)
- /ik uh... ga met de trein/ (/I er... am going by train/)
- /met de uh... vliegtuig/ (/by er... plane/*)
- /met het vliegtuig/ (/by plane/)

The baseline LM that is generated from this list, in which filled pauses are left out, is depicted in fig. 1.

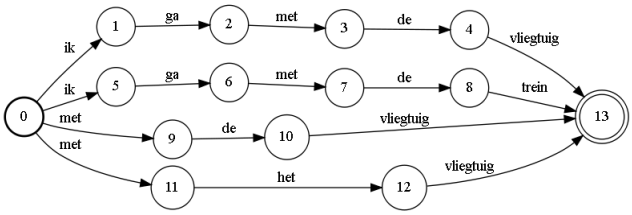

Figure 1: Baseline language model

To be able to decode possible filled pauses between words, we generated another LM with self-loops added in every node. Filled pauses are represented in the pronunciation lexicon as /@/ or /@m/, phonetic representations of the two most common filled pauses in Dutch. The filled pause loop penalty was empirically optimized.

To examine whether filled pause loops are an adequate way of modelling filled pauses, we also experimented with an oracle LM containing the reference orthographic transcriptions (which include the manually annotated filled pauses) without filled pause loops.

2.4. Acoustic Modelling

We trained three-state tied Gaussian Mixture Models (GMM). Baseline triphone models 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.

In several studies on non-native speech processing it has been observed that by adapting or retraining native acoustic models (AM) with non-native speech, decoding performance can be increased [3] [13]. To investigate whether this is also the case in a constrained task as described in this paper, we retrained the baseline acoustic models with non-native speech.

New AMs were obtained by doing a one-pass Viterbi training based on the native AMs with 6 hours of non-native read speech from the JASMIN corpus. These utterances were spoken by the same speakers as those in our test material.

Triphone AMs are the de facto choice for most researchers in speech technology. However, the expected performance gain from modelling context dependency by using triphones over monophones might be minimal in a constrained task. Therefore, we also experimented with non-native monophone AMs trained on the same non-native read speech.

2.5. Lexical Modelling

The baseline pronunciation lexicon contains canonical phonetic representations extracted from the CGN lexicon. It is well-known that non-native pronunciation generally deviates from native pronunciation, both at the phonetic and the phonemic level. To model this pronunciation variation at the phonemic level, pronunciation variants are usually added to the lexicon. Several researchers report a slight performance gain by including non-native pronunciation variants [13].

To derive pronunciation variants, we extracted context-dependent rewrite rules from an alignment of canonical and realized phonemic representations of non-native speech from the JASMIN corpus (the test material was excluded). Prior probabilities of these rules were estimated by taking the relative frequency of rule applications in their context.

We generated pronunciation variants by successively applying the derived rewrite rules to the canonical representations in the baseline lexicon. Variant probabilities were calculated by multiplying the applied rule probabilities. Canonical representations have a standard probability of 1. Afterwards, probabilities of pronunciation variants per word were normalized.

By introducing a cutoff probability, pronunciation lexicons were created that contain only variants above this cutoff, on average 2, 3, 4 and 5 variants per word.

2.6. Evaluation

We evaluated the speech decoding setups by using the utterance error rate (UER), which is the percentage of utterances where the 1-Best decoding result deviates from the transcription. For each UER the 95% confidence interval was calculated to evaluate whether UERs between conditions where significantly different. Filled pauses are not taken into account during evaluatio. That is, decoding results and reference transcriptions were compared after deletion of filled pauses.

As explained in the introduction, we don't expect our method to carry out a detailed phonetic analysis in the first phase. Since it is not necessary to discriminate between phonetically close responses at this stage, a decoding result can be classified as correct when its phonetic distance to the corresponding transcription is below a threshold. The phonetic distance was calculated through an alignment program that uses an adaptation of the standard dynamic programming algorithm to align transcriptions on the basis of distance measures between phonemes represented as combinations of phonetic features [14].

AM	LM	0	5	10	15
native (tri)	without loops	28.9	28.4	26.1	24.6
native (tri)	with loops	14.9	14.6	12.6	11.0
native (tri)	with positions	14.7	14.4	13.1	12.0
non-native(tri)	without loops	22.4	22.0	19.9	18.4
non-native(tri)	with loops	10.0	9.7	7.9	6.9
non-native(tri)	with positions	9.4	9.1	7.8	7.1
non-native(mono)	with loops	11.9	11.5	9.3	8.1

Table 1: This table shows the UERs for the different language models: without FP loops, with FP loops and with FP positions, and different acoustic models: trained on native speech (triphone) and retrained on non-native speech (triphone and monophone). All setups used the baseline canonical lexicon. The columns 0, 5, 10, 15 indicate at what phonetic distance to the reference transcription the decoding result is classified as correct.

Lex	Priors	0	5	10	15
canonical	-	10.0	9.7	7.9	6.9
2 var	no	10.0	9.9	8.2	6.7
2 var	yes	10.0	9.7	8.3	7.0
3 var	no	11.2	10.9	8.5	7.1
3 var	yes	10.6	10.1	8.7	7.2
4 var	no	11.5	11.3	8.9	7.5
4 var	yes	10.4	10.9	9.7	7.2
5 var	no	11.5	11.3	8.9	7.5
5 var	yes	10.4	10.0	8.7	7.2

Table 2: UERs for different lexicons: canonical, 2-5 variants with and without priors. These rates are obtained by using non-native triphone acoustic models and language models with filled pause loops.

3. Results

In table 1 the UER for the different language models and acoustic models can be observed. In all cases, the language model with filled pause loops performed significantly better than the language model without loops. Furthermore, the oracle language model with manually annotated filled pauses did not perform significantly better than the language model with loops.

Decoding setups with acoustic models trained on non-native speech performed significantly better than those with acoustic models trained on native speech.

The performance difference between monophone and triphone acoustic models was not significant. As expected, error rates are lower when evaluating using clusters of phonetically similar responses. To better appreciate the results in table 1 it is important to get an idea of the meaning of these distances. For instance, a phonetic distances between 0 and 5 generally indicates that the utterances differ by 1 or 2 segments; a distance between 5 and 10 usually stands for a discrepancy of a short word, and distances larger than 10 are observed when the differences concern long words. Since there are few responses with a phonetic distance smaller than 5, differences between conditions 0 and 5 are marginal. Performance differences between 0 (equal to transcription) and 10 (one of the answers with a phonetic distance of 10 or smaller to the 1-Best equals the transcription) and between 5 and 15 were significant.

Performance decreased using lexicons with pronunciation variants generated using data-driven methods. The more variants are added, the worse the performance. There is no signifi-

cant difference between using equal priors or estimated priors.

4. Discussion and Conclusions

The results presented in the previous section indicate that large and significant improvements could be obtained by optimizing the language model and the acoustic models. On the other hand, pronunciation modelling at the level of the lexicon did not produce significant improvements. On the contrary, adding variants to the lexicon caused a decrease in performance. Adding estimated prior probabilities to the variants improved the results somewhat, but still the error rates remain higher than those for the canonical lexicon. These results might be surprising because, in general, adding a limited number of pronunciation variants to the lexicon helps improve performance to a certain extent. However, in the case of non-native speech this strategy is not always successful [15]. Possible explanations might be sought in the nature of the variation that characterizes non-native speech. Non-native speakers are likely to replace target language phonemes by phonemes from their mother tongue [4]. When the non-native speech is heterogeneous in the sense that it is produced by speakers with different mother tongues, as in our case, it is extremely difficult to capture the rather diffuse pattern of variation by including variants in the lexicon (see also [16]).

The findings that better results are obtained with non-native acoustic models and with a language model with filled pause loops are not surprising, after all the utterances are spoken by non-natives, recorded in the same environment and contain a lot of filled pauses. In fact, these results do not differ significantly from the results obtained with an oracle language model, in which the exact position of the filled pauses is copied from the manual transcriptions. This is an important result because non-natives are known to produce numerous filled pauses in unprepared, extemporaneous speech [17]. It is therefore to be expected that in the eventual DISCO application we will have to deal with utterances that contain filled pauses, and it is good to see that despite the great number of filled pauses in the material, sequences of words can often still be recognized correctly.

Another reassuring result is that concerning the non-native acoustic models. These were obtained by retraining native models on a relatively small amount (around 8 minutes per speaker) of non-native read speech material. It appears that this was sufficient to obtain significantly better results. In the final application we might then use a relatively short enrolment phase and do acoustic model retraining or adaptation, to obtain better recognition results.

As explained above, the DISCO system will first determine the content of the utterance before proceeding to error detection. In this setup 100% accuracy is not strictly required in the first phase. If it is not clear whether a segment or a (short) word was pronounced or not, this can be ascertained in the second phase through a more detailed analysis [9]. At the moment we think that in the second phase we can handle utterances with a phonetic distance smaller than 5, which usually corresponds to a difference of 1 or 2 segments, or possibly even utterances with a phonetic distance smaller than 10, which often boils down to a deviation by a short word. For the latter category the best result obtained is an error rate of around 8%. This is encouraging, especially if we keep in mind that in a language learning application we can be conservative, in the sense that if we are not sufficiently confident about the recognition result we can always ask the language learner to try again.

In the near future, we will try to improve our results, if only by using more data. We are already collecting more non-native speech material, and once we have a first version of the application we will collect material which is obviously more suitable for improving speech recognition for our system. Error rates will depend on the length of the list (of possible responses), but also on the degree of confusability between the responses in the list, the phonetic distances. We will try to gain more insight into how these factors can affect performance, and will use this knowledge in designing our application.

5. 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/).

6. References

[1] Van Compernolle, D. (2001) Recognizing speech of goats, wolves, sheep and..non-natives. Speech Communication 35, 1-2, 71-79.

[2] Raux A. and Eskenazi, M. (2004) Using Task-Oriented Spoken Dialogue Systems for Language Learning: Potential, Practical Applications and Challenges, In Proceedings of INSTILL, 2004.

[3] Mayfield Tomokiyo L. & Waibel, A. (2001) Adaptation Methods for Non-native Speech. In Proceedings of Multilinguality in Spoken Language Processing, 2001, Aalborg.

[4] Bouselmi, G., Fohr, D., Illina, I., Haton, J.P. (2006) Multilingual Non-Native Speech Recognition using Phonetic Confusion-Based Acoustic Model Modification and Graphemic Constraints, In Proceedings of ICSLP, 2006.

[5] Menzel, W., Herron, D., Morton, R., Pezzotta, D., Bonaventura, P., and Howarth, P. (2000) Interactive pronunciation training, ReCALL, 13 (1), 67-78.

[6] Auralog (2000) Tell me More, Users Manual, Montigny-le-Bretonneux, France.

[7] Talk to Me, the Conversation Method, http://www.auralog.com/en/talktome.html. Last consulted 16/04/2009.

[8] Ehsani, F., Bernstein, J. and Najmi, A. (2000) An interactive dialog system for learning Japanese, Speech Communication, 30, 167-177.

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

[10] Cucchiarini, C., Driesen, J., Van hamme, H. and Sanders, E. (2008) 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. (2008) SPRAAK: An Open Source SPeech Recognition and Automatic Annotation Kit, In Proceedings of ICSLP, page 495, 2008.

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

[13] Kessens, J. (2006) Non-native pronunciation modeling in a command & control recognition task: a comparison between acoustic and lexical modeling, In Proceedings of MULTILING, 2006.

[14] Cucchiarini, C. (1996). Assessing transcription agreement: ethodological aspects. Clinical Linguistics and Phonetics 102: 131155.

[15] Goronzy, S., Rapp, S., and Kompe, R. (2004). Generating non-native pronunciation variants for lexicon adaptation. Speech Communication, 42(1):109-123.

[16] Benzeguiba, M., De Mori, R., Deroo, O., Dupont, S., Erbes, T., Jouvet, D., Fissore, L., Laface, P., Mertins, A., Ris, C., Rose, R., Tyagi, V., Wellekens, C. (2006) In Proceedings of ICASSP, 2006.

[17] Cucchiarini, C., Strik, H. and Boves, L. (2002). Quantitative assessment of second language learners fluency: Comparisons between read and spontaneous speech. JASA 111: 2862-2873.