Optimizing automatic speech recognition for low-proficient non-native speakers

Joost van Doremalen, Catia Cucchiarini, Helmer Strik
{j.vandoremalen, c.cucchiarini, h.strik}@let.ru.nl
Department of Linguistics

September 4, 2009

Abstract

Computer Assisted Language Learning (CALL) applications for improving the oral skills of low-proficient learners have to cope with non-native speech that is particularly challenging. Since unconstrained non-native ASR is still problematic, a possible solution is to elicit constrained responses from the learners. In this paper we describe experiments aimed at selecting utterances from lists of responses. The first experiment on utterance selection indicates that the decoding process can be improved by optimizing the language model and the acoustic models, thus reducing the utterance error rate from 29-26% to 10-8%. Since giving feedback on incorrectly recognized utterances is confusing, we verify the correctness of the utterance before providing feedback. The results of the second experiment on utterance verification indicate that combining duration related features with a likelihood ratio (LR) yields an equal error rate (EER) of 10.3%, which is significantly better than the EER for the other measures in isolation.

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 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. However, developing ASR-based CALL systems that can provide training and feedback for second language (L2) speaking is not trivial.

First of all, because non-native speech is atypical in many respects and, as such, it poses serious problems to ASR systems [1] [2] [3] [4]. Non-native speech may deviate from native speech with respect to pronunciation, morphology, syntax and the lexicon. Pronunciation is considered a difficult skill to learn in a second language (L2), and even highly proficient non-native speakers often maintain a foreign accent [5]. An important limiting factor in acquiring the pronunciation of an L2 is considered to be interference from the first language (L1). As a consequence, the pronunciation of non-native speakers

may deviate in various respects and to different degrees from that of native speakers. Deviations may concern prosodic or segmental aspects of speech or both. At the segmental level the deviations may be limited to phonetic properties without really compromising phonemic distinctions, or they may blur phonemic distinctions and thus have more serious consequences for intelligibility. For instance, non-native speakers may use phonemes from their L1 when speaking the target language [5] or they may have difficulties in perceiving and/or realizing phonetic contrasts that are not distinctive in their mother tongue. Illustrations of this phenomenon are provided by Italian speakers of English who realize English /p/, /t/, /k/, /b/, /d/, and /g/ with voice onset time (VOT) values that differ from those employed by native speakers [5]. Such deviations might cause misunderstandings in certain cases, but do not necessarily hamper communication because the distinction between separate phonemes, i.e /p/ vs /b/ in the target language is preserved, albeit differently realized. Native speakers will probably perceive the difference and consider it as foreign accent. More problematic deviations may arise when the difficulty in perceiving and realizing phonetic features of the target language that are not distinctive in the mother tongue leads non-native speakers to blur the distinction between phonemes in the target language, thus producing one phoneme instead of two distinct ones. This is the case with many non-native speakers of English, for instance Germans [6], who have difficulty in realizing the distinction between the English phonemes /ae/ and /e/ and often produce /e/ when /ae/ should be used, or Japanese speakers of English who have difficulty in distinguishing /l/ and /r/ [7] and may end up producing sounds that are neither an English /l/ nor an English /r/. In such cases confusion may arise because distinct words will be realized in the same way. This can also happen when speech sounds are inappropriately deleted or inserted, which is another common phenomenon in non-native speech [8].

With respect to morphology and syntax the speech of non-natives may also exhibit deviations from that of native speakers. [9]. At the level of morphology they may find it difficult to produce correct forms of verbs, nouns, adjectives, articles etc, especially when the morphological distinction hinges on subtle phonetic distinctions, such as the presence of a plosive or fricative sound in consonant clusters or the distinction between two similar vowels (this vs these). Irregular verbs and nouns may also pose serious problems, resulting in the production of non-existent regularized forms. Deviations in syntax may concern the structure of sentences, the ordering of constituents and their omission or insertion. As to vocabulary, non-native speakers also tend to have a limited and often deviant lexicon. Finally, non-native speech exhibits more disfluencies and hesitation phenomena than native speech and is characterized by a lower speech rate [10] [11] [12] [13][14].

All these problems are compounded when dealing with speech of non-natives that are still in the process of learning the language. In general, the degree of deviation from native speech and the incidence of disfluencies will be in inverse relation to the degree of proficiency in the target language. Considering that ASR-based CALL systems are intended for L2 learners, including beginner and intermediate learners, it follows that the type of non-native speech that has to be handled in this context is, in general, even more atypical and therefore more challenging, than the non-native speech that is usually encountered in other ASR applications that do not have such a teaching function,

like information systems or access interfaces.

To circumvent the ASR problems caused by non-native speech, various techniques have been proposed to 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. Such strategies are often used in combination.

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* [15] and the *Let's Go* systems [16]. 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. Less constrained output is not only problematic because it is more difficult to predict, but also because, in general, it is accompanied by a higher incidence of disfluencies and hesitations. In a study on read and spontaneous speech produced by non-native speakers of Dutch [12], we found that extemporaneous speech contains many more filled pauses and disfluencies than read speech. The more freedom is allowed to the learner, the more complex the recognition task will be. In addition, tasks with more freedom will in general be characterized by a higher cognitive load, which, in turn, is likely to lead to more disfluencies being produced [17], thus making the recognition task even more difficult.

The second category of techniques for dealing with non-native speech, i.e. those that are aimed at improving decoding, comprises methods for optimizing the acoustic models, the lexicon and the language model in order to compensate for the deviations in pronunciation, morphology and syntax.

All the factors mentioned above make it clear that to develop ASR-based CALL systems for oral proficiency it is necessary to take measures at different levels. A first important measure consists in designing exercises that allow some freedom to the learners in producing answers, but that are predictable enough to be handled by ASR. How much freedom can be allowed is of course dependent on the quality of decoding.

These are exactly the problems we face in the DISCO project, which is aimed at developing a prototype of an ASR-based CALL application for practicing oral skills in Dutch as a second language (DL2) and providing intelligent feedback on important aspects of speaking performance such as pronunciation, morphology, and syntax. The application should be able to detect and give feedback on errors that are made by learners of DL2 at the A2 level of the Common European Framework (CEF). This is achieved by generating a predefined list of possible (correct and incorrect) responses for each exercise.

In this project we intend to use a two-step procedure in which first the content of the utterance is determined (what was said), and subsequently the form of the utterance is analysed (how it was said). In the first (recognition) step the system should tolerate deviations in the way utterances are spoken, while in the second (error detection) step, strictness is required (see also [18] and [19]). In the first step of the two-step procedure two phases 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 phase (a) as one of the utterances of the predefined list. And even if the utterance is present in the list, it can also be recognized incorrectly. 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 phase (b).

In this paper we present two experiments we carried out in order to test both utterance selection and utterance verification for our system using state-of-the-art techniques. In the utterance selection phase one of the utterances from the predefined list is selected, and in the utterance verification phase it is determined whether this utterance should be passed on to the following stages of the CALL system (error detection, feedback, etc.). While in the final system both phases should work in tandem, we studied (optimized, evaluated, etc.) the two phases in isolation, for diagnostic purposes, to acquire a better understanding, and thus, finally, to obtain a better functioning system.

In Section 2 we discuss related work on non-native speech recognition and utterance verification. In Section 3 we introduce our system architecture and relate the choices for the experimental settings to previous work. In Section 4 and 5 we present two experiments that are aimed at optimizing and evaluating utterance selection and utterance verification using realistic test material. In Section 6 we discuss the results of the two experiments in combination and consider the implications for our CALL application.

2 Related work

In automatic speech recognition (ASR) the recognition result is often obtained using the *maximum a posteriori* (MAP) decision rule decoder:

$$\hat{w} = \arg\max_{w \in W} p(w|\mathbf{x}) \tag{1}$$

where $p(w|\mathbf{x})$ is the posterior probability of a word sequence w in a set of word sequences W given a sequence of acoustic observations $\mathbf{x}$ and $\hat{w}$ is the recognition result that maximizes the posterior probability.

By using Bayes rule Eq. 1 can be reformulated as Eq. 2, and given that $\mathbf{x}$ is the same for all word sequences in W, it can be rewritten as Eq. 3:

$$\hat{w} = \arg\max_{w \in W} \frac{p(\mathbf{x}|w)p(w)}{p(\mathbf{x})} \tag{2}$$

$$= \arg\max_{w \in W} p(\mathbf{x}|w)p(w) \tag{3}$$

By implementing Eq. 3 we can still find the optimal sequence of words $\hat{w}$ in W. However, it is generally not only important to find the best sequence of words $\hat{w}$ relative to the others sequences (Eq. 3), but also quantitatively assess the confidence in the recognition result in an absolute sense. This number is called the confidence measure (CM) of the recognition result and the problem of accepting or rejecting a recognition result is called utterance verification (UV).

Both (non-native) speech decoding and utterance verification are the key aspects of this research. We will now relate our research on both problems to other recent work.

2.1 Non-native speech decoding

In the ASR community, it has long been known that the differences between native and non-native speech are so pervasive as to degrade ASR performance considerably (e.g. [20] [21] [1]). These differences affect essentially all three components of an ASR system. As explained in Section 1, non-natives often use different words and word orders (language model), produce sounds differently (acoustic models), pronounce words differently (lexicon) (see, for instance [2]), and generally have a lower speech rate and produce more disfluencies ([10] [11] [12]). A short overview of research on the three components of the ASR is provided in this section.

In attempts aimed at improving ASR performance on non-native speech, the acoustic models have received most attention. Various kinds of acoustic models can and have been used. First of all, it is possible to train acoustic models on speech material of the target language (L2). However, the recognition performance obtained with such models is usually not sufficient or at any rate considerably lower than the performance on native speech, because of the various deviations in the speech of non-natives [20] [21]. Models can also be obtained by training exclusively on non-native (L2) speech [22] [23], or on combinations of L1 and L2 speech. Regarding the latter, two different approaches can be adopted: "model merging" and "parallel models". In the "parallel models" approach, acoustic models for both languages are stored, and during decoding the recognizer determines which models fit the data better [24] [25] [26] [27]. In the "model merging" (or model interpolation) approach, acoustic models of both languages are combined, in order to obtain a new set of acoustic models [26]. The obvious disadvantage of these L1-L2 approaches is that they can only be applied to fixed L1-L2 pairs. An alternative approach that can be applied consists in employing adaptation techniques, such as the common Maximum Likelihood Linear Regression (MLLR) and MAP techniques, which have shown to improve recognition performance [20] [23] [21] [28] [26]

Improving ASR performance on non-native speech can also be carried out at the level of the lexicon. An obvious way to model pronunciation variation at the level of the lexicon is by adding pronunciation variants to the lexicon [29] [30]. In the case of non-native speech these variants should reflect possible L1-induced mispronunciations of words L2 learners may produce [31] [32] [18]. These variants can be generated by means of rules obtained from studying non-native speech [32] [18]. Another possibility to generate non-native variants for a L2 lexicon is to apply an L1 phoneme recognizer to L2 speech [31]. The advantage of the latter approach is that no learner data are needed, but a disadvantage is that phoneme recognizers for all source languages (L1s) are needed. [31] also carried out speaker adaptation, and the improvements they obtained with speaker adaptation were much larger than those obtained with lexicon adaptation.

The choices regarding the language model depend to a large extent on the design of the CALL system, the type of items present in the CALL system. In spoken CALL systems use could be made of closed or open items. For instance,

the learner could be asked to repeat an utterance that is spoken by the system, or read an utterance presented on the screen. In these cases the required responses are known, which in turn makes it possible to derive specific language models for every item. Alternatively, in some cases a language model might not be used at all, depending on the approach that is chosen. For more open items in a CALL system (e.g. a question, or a turn in the dialogue), a possibility is to try to elicit constrained responses. This makes it possible to activate a specific language model for every item containing only those utterances that are expected in that given context. In these cases, a 'stricter' language model can be used [33] [34] [35]. In this way, recognition performance can again be maximized without affecting the face validity of the application. This is done, for instance, in the Auralog programs [36]. In spite of the constraints that are introduced to improve ASR performance, the students can still have the feeling that they are interacting with the system and that they have control over the conversation [36].

2.2 Utterance verification

In the literature roughly three approaches for tackling the UV problem can be distinguished: (1) posterior probability estimation, (2) statistical hypothesis testing and (3) confidence predictors. We will now give a short overview of these approaches (see [37] for a more detailed overview).

(1) One approach to CM is to directly estimate the posterior probability of the recognition result $\hat{w}$ given the acoustic observations $\mathbf{x}$:

$$p(\hat{w}|\mathbf{x}) = \frac{p(\mathbf{x}|\hat{w})p(\hat{w}))}{p(\mathbf{x})} \tag{4}$$

and reject the recognition result $\hat{w}$ when it is below a given threshold θ. The greatest challenge with respect to this approach is accurately estimating the denominator $p(\mathbf{x})$. One solution is to estimate it from a word lattice [38], and this generally provides a good result when the lattice contains enough word hypotheses. The lattice-based approach can be viewed as approximating the posterior probability where $p(\mathbf{x})$ is written as $\sum_i p(\mathbf{x}|w_i)p(w_i)$ and i ranges over all sequences of words in a pruned search space.

Another approach to estimating $\sum_i p(\mathbf{x}|w_i)p(w_i)$ is using a free phone recognizer (FPR) [39] [40] and approximate:

$$p(\mathbf{x}) \approx p(\mathbf{x}|u_{FPR})p(u_{FPR}) \tag{5}$$

where u_{FPR} is the optimal phone string found using a free phone recognizer.

(2) Another popular method to UV is statistical hypothesis testing, in which the null hypothesis H_0 states that the recognition result is a correct representation of the speech signal and the alternative hypothesis H_a states that the recognition result is not a correct representation. Then the criterion of accepting the null hypothesis becomes:

$$\frac{p(\mathbf{x}|\hat{w})}{p(\mathbf{x}|\neg\hat{w})} > \theta \tag{6}$$

in which the numerator equals the acoustic likelihood of $\hat{w}$, the denominator equals the acoustic likelihood of all sequences of words other than $\hat{w}$ (usually

called the *anti-model*) and θ a predefined threshold. The main difficulty with this approach is defining and training the anti-model.

(3) Apart from estimating the posterior probability or statistical hypothesis testing, another method to UV is using predictors such as

(1) acoustic stability,

(2) hypothesis density,

(3) duration information

and combine these using a machine learning model. Some machine learning techniques that have been used in the past are artifical neural networks (ANN), linear discriminant analysis (LDA) classifiers and binary decision trees.

Acoustic stability [38] refers to stability of the recognition result given different weightings of the acoustic model and language model scores. When the recognition result remains stable given fluctuations in these weightings it means that we can be more confident that it is correctly recognized. Hypothesis density [41] refers to the average density of the word lattice generated during decoding. When there are a lot of competing hypotheses in a pruned search space at each point in time this means that we can be less confident that the recognition result is correct. Duration modelling for UV usually comes down to capturing the amount of deviation of the phoneme durations in the recognition result from *normal* phone durations [42]. Deviating durations in the recogniton result decreases the confidence that it is recognized correctly.

3 Experimental system

In Fig. 1 the architecture of our CALL system is shown. The input of the system is the learner's speech and a list of predicted responses in the form of transcriptions of sequences of words. Utterance selection is then performed to choose the best fitting (1-Best) response from this list. In the next phase the 1-Best response is verified. If the response is accepted, error detection on this response is carried out. Errors are detected on multiple levels, i.e. syntax, morphology and pronunciation. If the response is not accepted, the user is prompted to try again.

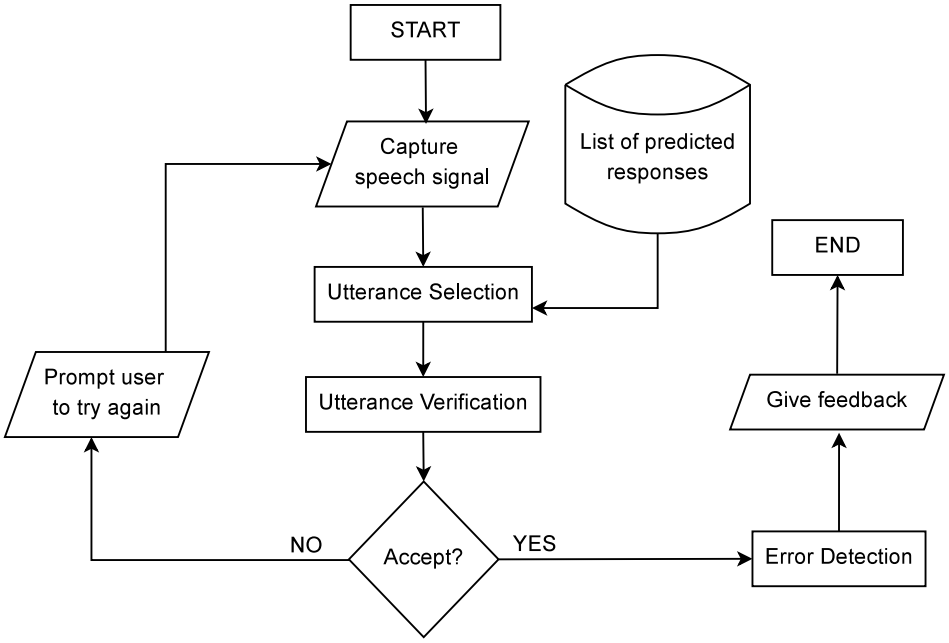
It is difficult for general Hidden Markov modelling methods to discriminate between utterances that are acoustically very similar [43]. Therefore, in the final CALL system we will probably use the following procedure: the output of the first step is a cluster of similar responses (e.g. according to a phonetically-based distance measure), and a more detailed analysis is carried out in the second (error detection) step to determine what was actually uttered and where to give feedback on.

We will now explain the main choices we made for our system regarding utterance selection and utterance verification procedures.

3.1 Utterance selection

In the literature many approaches have already been proposed to improve the performance of speech recognition for non-natives. A large deal of the research

Figure 1: System architecture.



concerned one or a small number of fixed (L1-L2) language pairs. In these approaches material of the source language (L1) or material for specific L1-L2 pairs was employed to enhance ASR for these language pairs. However, since our system is intended for learners of Dutch with different mother tongues, approaches that require material of L1 or specific L1-L2 pairs are not feasible in our case for either of the three components of an ASR system (acoustic models, lexicon, and language model). Consequently, we made the following choices.

For the acoustic models we decided to start with training the acoustic models on Dutch native speech. Next, we used read speech of language learners (DL2 speech) to retrain the acoustic models (see Section 4.1.4). Such retraining of the acoustic models is also possible in a realistic CALL application, albeit not on line, after a so-called enrolment phase, as used in dictation systems. Especially if the system has to be used extensively by a learner, it is possible to make it as suitable as possible for that specific learner. At the level of the lexicon we could not make use of L1 phoneme recognizers, as was done by [31], and thus we added pronunciation variants to the lexicon that were generated by means of data-derived rules (for further details, see Section 4.1.5). Finally, we decided to use specific language models for every item in the CALL system that are based on a list of predicted (correct and incorrect) responses (see Section 4.1.3).

3.2 Utterance verification

In Section 2.2 we have given a short overview of the three key approaches to UV i.e. (1) posterior probability estimation (2) statistical hypothesis testing and

(3) predictor combination. Most of these approaches are aimed at UV in large vocabulary tasks, i.e. posterior probability estimation using word lattices and predictor features like acoustic stability and hypothesis density. Furthermore, training explicit anti-models for statistical hypothesis testing is conceptually and practically difficult for speakers with a large variety of L1 backgrounds [44]. For these reasons, we have chosen a form of predictor combination in which a likelihood ratio similar to Eq. 6 in statistical hypothesis testing is combined with phone durations. The rationale behind this choice is explained in detail in Section 5.1.2.

4 Experiment 1: Utterance selection

To goal of this experiment is to develop a procedure for selecting utterances from a list of predicted responses and to evaluate the effects of different language models, pronunciation lexicons and acoustic models.

4.1 Method

4.1.1 Material

The speech material for the present experiments was taken from the JASMIN speech corpus [45], 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 is particularly suited for our purpose. Speech from speakers with different mother tongues was collected, because this realistically reflects the situation in Dutch L2 classes. 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 our CALL 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.

Considering all these characteristics we can state that the JASMIN non-native dialogues are similar to the speech we will encounter in our CALL ap-

plication for various reasons: 1) they contain answers to relatively constrained questions, 2) they contain semi-spontaneous speech 3) of non-natives with different L1s, 4) which features spontaneous phenomena such as filled pauses and disfluencies. However, since hesitation phenomena were purposefully induced in the JASMIN dialogues, their incidence is probably higher than in typical non-native dialogues.

4.1.2 Speech Recognizer

The speech recognizer we used in this research is SPRAAK [46], an open source hidden markov model (HMM)-based 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 our CALL 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.

4.1.3 Language Modelling

Our approach is 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 produce a limited amount of 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.

11% of the utterances contain one or more other disfluencies such as restarts, repairs and repetitions. While these 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, we decided not to tolerate restarts, repetitions and repairs and to ask the students to try again when one of these phenomena is produced. Therefore, in our research we did not focus on restarts, repairs and repetitions, we only included their orthographic transcriptions in the LM and their manual phonetic transcriptions in the lexicon.

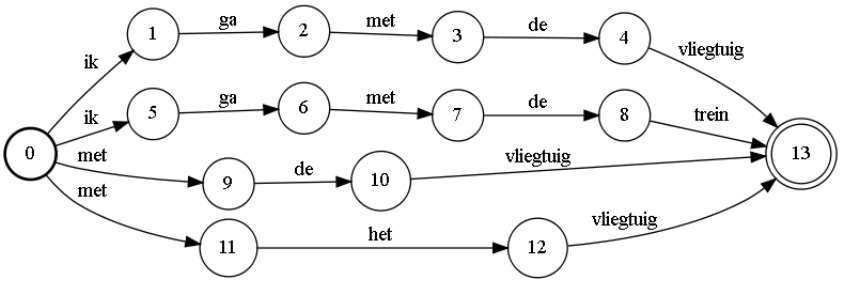
The LMs are implemented as FSMs with parallel paths of orthographic transcriptions of every unique answer to the question. A priori each path is equally likely. An example of such a question is “Hoe wilt u naar deze stad reizen?” (“How do you want to travel to this city?”) and a small part of the responses is:

1. /ik gaat met de vliegtuig/ (/I am going by plane/*)

2. /ik ga met de trein/ (/I am going by train/)
3. /met de vliegtuig/ (/by plane/*)
4. /met het vliegtuig/ (/by plane/)

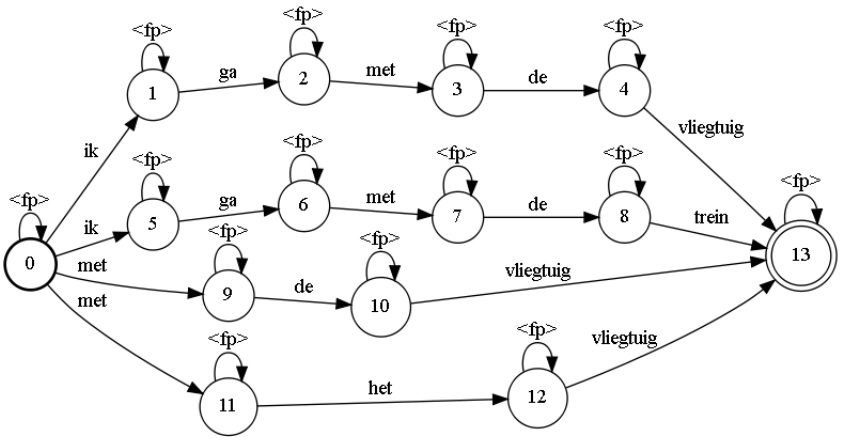
The baseline LM that is generated from this list is depicted in Fig. 2. Each of the parallel paths with words on the arcs represents a unique answer to a question. Silence is possible before and after each word (not shown).

Figure 2: 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. An example of this language model is depicted in Fig. 3

Figure 3: Language model with filled pause loops



To examine whether filled pause loops are an adequate way of modelling filled pauses, we also experimented with an oracle LM. This is an LM con-

taining the reference orthographic transcriptions, which include the manually annotated filled pauses without filled pause loops.

4.1.4 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 [47]. In total 11,660 triphones were created, using 32,738 Gaussians.

As discussed in Section 2.1, it has been observed in several studies that by adapting or retraining native acoustic models (AM) with non-native speech, decoding performance can be increased. 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 (comparable to an enrollment phase).

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.

4.1.5 Lexical Modelling

The baseline pronunciation lexicon contains canonical phonemic representations extracted from the CGN lexicon. The distribution of sizes of the 39 lexica is depicted in Fig. 4.

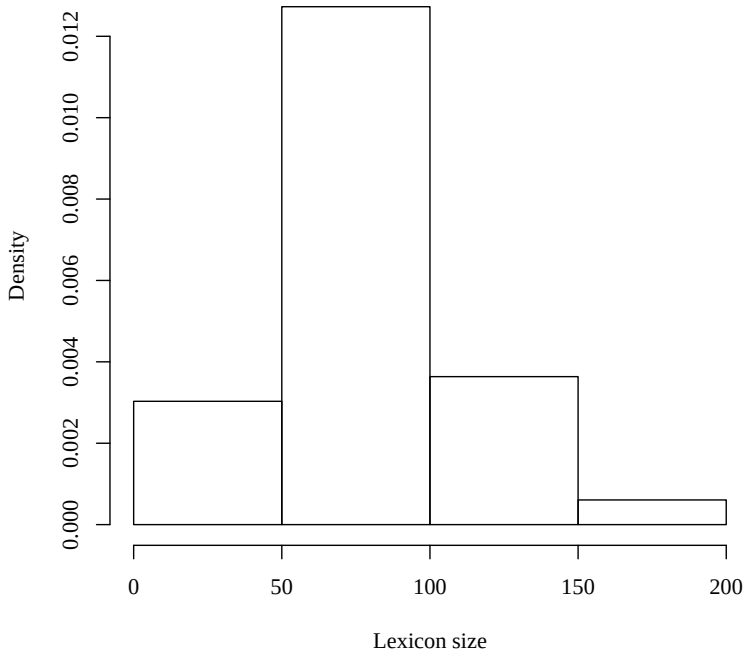
As explained in Section 2.1 non-native pronunciation generally deviates from native pronunciation, both at the phonetic and the phonemic level. To model pronunciation variation at the phonemic level, we added pronunciation variants to the lexicon.

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 one. Afterwards, probabilities of pronunciation variants per word were normalized so that these probabilities sum to one.

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

Figure 4: Distribution of lexicon sizes.



4.1.6 Evaluation

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

As explained in the introduction, we do not 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 a dynamic programming algorithm to align transcriptions on the basis of distance measures between phonemes represented as combinations of phonetic features [48]. These phonemic transcriptions were made using the canonical pronunciation variants from the words in the orthographic transcriptions.

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.

Response	1	2	3	4
1	0.0	-	-	-
2	20.5	0.0	-	-
3	15.0	23.5	0.0	-
4	23.5	30.0	10.0	0.0

Table 2: Phonetic distances between the example responses: (1) 'ik gaat met de vliegtuig', (2) 'ik ga met de trein', (3) 'met de vliegtuig', (4) 'met het vliegtuig'.

4.2 Results

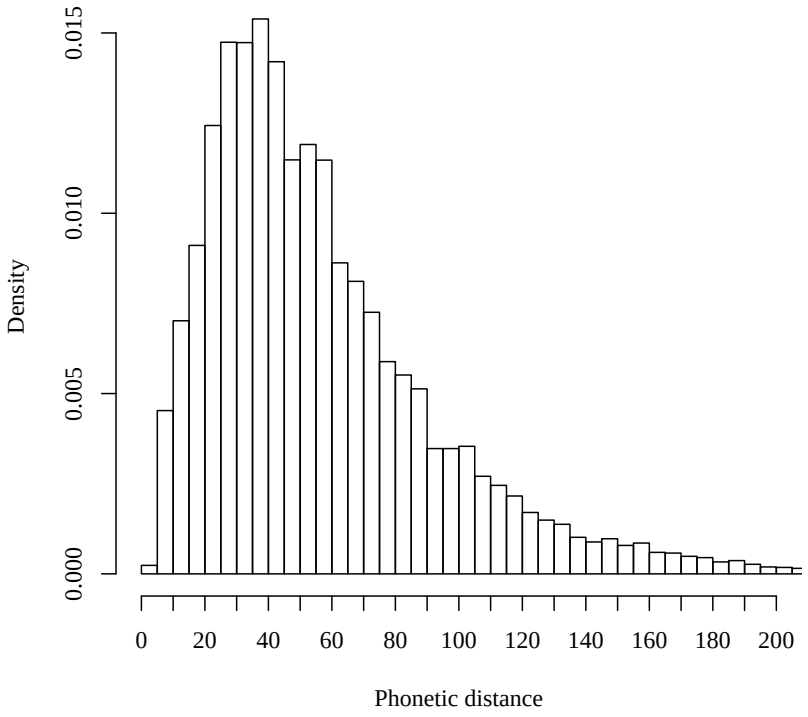
In Table 1 the UERs for the different language models and acoustic models can be observed. In all cases, the LM with filled pause loops performed significantly better than the LM without loops. Furthermore, the oracle LM with manually annotated filled pauses (with positions) did not perform significantly better than the LM with loops.

Decoding setups with AMs retrained on non-native speech performed significantly better than those with AMs trained on native speech. The performance difference between monophone and triphone AMs 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. The distances between the example responses in Section 4.1.3 are shown in Table 2. The density of the phonetic distances between all response pairs to all questions is depicted in Fig. 5. Since there are only few responses with a phonetic distance smaller than 5, differences between 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.

As can be seen in Table 3, performance decreased using lexicons with pronunciation variants generated using data-driven methods. The more variants are added, the worse the performance. Furthermore, there is no significant

Figure 5: The distribution of phonetic distances between all response pairs to all questions.



difference between using equal priors or estimated priors.

4.3 Discussion

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 carefully selected pronunciation variants to the lexicon helps improve performance to a certain extent [29] [30]. However, in the case of non-native speech this strategy is not always successful [31]. 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 [5] [3]. When the non-native speech is heterogeneous in the sense that

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 3: 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.

it is produced by speakers with different mother tongues, as in our case, it may be extremely difficult to capture the rather diffuse pattern of variation by including variants in the lexicon (see also [4]).

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 [12]. From these results we can conclude that external filled pause detection, for which better results were found for a large vocabulary task [49], is not necessary in this case.

Another reassuring result is that performance improved using 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 (and/or online speaker adaptation), to obtain better recognition results.

While in this experiment the correct transcription of the response was always in the language model, our system must also be able to reject utterances when they are not present in the language model, while still accepting correctly recognized utterances. This is the topic of the experiment presented in the next section.

5 Experiment 2: Utterance verification

The goal of this experiment is to develop a procedure for utterance verication. Our approach consists of combining an acoustic likelihood ratio with duration-related predictors into one confidence measure.

5.1 Method

5.1.1 Material

We used the same material as in the first experiment, but 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, each of the 1325 utterances in our dataset is decoded two times: one time when its representation is present in the language model and one time when it is not present.

5.1.2 Confidence predictors

As mentioned in Section 4.2, posterior probability estimation using rich word lattices is often used in large vocabulary applications, where it usually provides accurate confidence measures, although it is computationally expensive. Since in our case the search space only contains a limited set of sequences of words, the decoding lattice is not rich enough to estimate $p(\mathbf{x})$ (Eq. 4.). Estimating $p(\mathbf{x})$ on the basis of a free phone recognizer (FPR) is a more simple and faster approach, generally giving reasonably good results. For these reasons, we have used the ratio

$$\frac{p(\mathbf{x}|\hat{w})p(\hat{w})}{p(\mathbf{x}|u_{FPR})p(u_{FPR})} \tag{7}$$

as our baseline confidence measure. However, because we have equal prior probabilities for all language model paths and we do not use a language model during free phone recognition the priors $p(\hat{w})$ and $p(u_{FPR})$ can be discarded and Eq. 7 boils down to:

$$LR = \frac{p(\mathbf{x}|\hat{w})}{p(\mathbf{x}|u_{FPR})} \tag{8}$$

This ratio bears a close relation to Eq. 6 used in the statistical hypothesis testing approach to UV. The main difference is that in the denominator in Eq. 8 all paths are used, while in Eq. 6 only the alternative paths are used to compare with the recognition result to be verified. Modelling the alternative paths in an anti-model is especially difficult in our task because it is very difficult to determine what exactly it should represent if the utterance is produced by language learners with generally low levels of proficiency and very diverse L1 backgrounds (see also [44]). Furthermore, training such an anti-model requires a large amount of non-native speech data that is not available for Dutch.

We hypothesize that combining our baseline CM (LR) with other predictors that contain additional information about the quality of the recognition result will give better results than using LR alone. However, using the average hypothesis density in the word lattice as a predictor is probably not informative because in our task the word lattice is very small and contains very few competing hypotheses. Furthermore, a predictor like acoustic stability is difficult to define because different weightings of the language model have no effect on the combination score (because *a priori* each sequence of words in the language model is equally likely).

We expect that phone durations might contain additional information, because the phone segmentation of an incorrectly decoded sequence of words will generally be characterized by deviations in phone durations and this is

not directly coded in the acoustic likelihoods in *LR*. Therefore, we want to add information about these phone duration deviations.

When the input speech representation is not present in the list and the utterance is recognized as another sequence of words that is present in the LM, 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 [42].

Phone duration distributions were derived from manually verified phonemic transcriptions of 42 hours of read native speech from the CGN corpus [47]. 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.

5.1.3 Predictor 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 have used a logistic regression model. Logistic regression modelling is a straightforward and fast method known to produce accurate predictions when a binary variable is a linear function of several explanatory variables [50]. It fits the logit of the probability (logarithm of the odds) of a binary event as a linear function of the set of explanatory variables:

$$\text{logit}(p(y|\mathbf{p})) = \frac{p(y|\mathbf{p})}{1 - p(y|\mathbf{p})} = \beta_0 + \sum_{i=1}^{N} \beta_i x_i \tag{9}$$

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 chosen through Maximum Likelihood Estimation (MLE) in WEKA [51]. 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 were left out during training. This is repeated until all speakers are tested.

5.1.4 Evaluation

We 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 on the ROC curve where the error rates of both 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.

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%

Table 4: 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*.

5.2 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% (see Section 4.2). 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 on average 55.0% of all the cases was incorrectly recognized.

The task for the UV was to discriminate the correctly and incorrectly recognized cases. In Table 4 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 6.

Within the individual predictors *LR* performs best (14.4%) and all the duration-related predictors perform much worse. The best result for a single duration predictor is 27,3% for *nr_shorter_1*. 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 $\pm$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 5a and 5b 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 5c.

5.3 Discussion

The duration-related predictors have a weak performance individually, but they still contain additional information relative to the 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

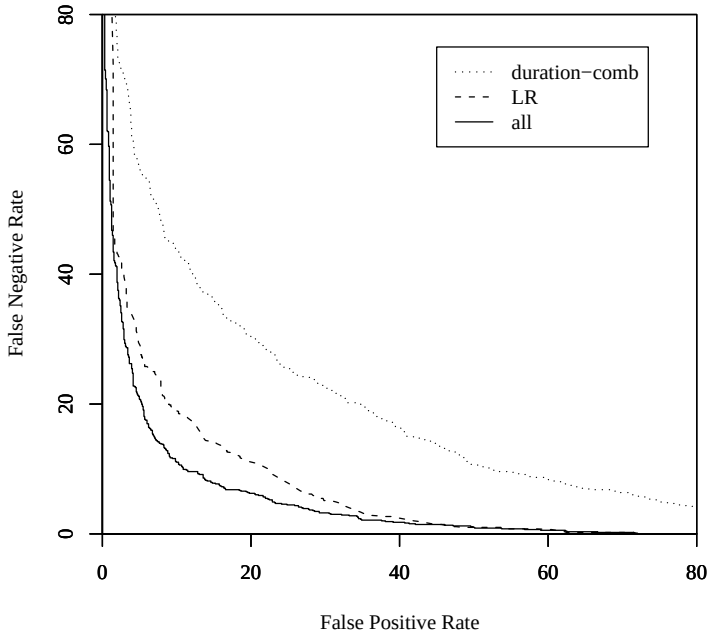


Figure 6: ROC curves for the feature *LR* and the combinations *duration_comb* and *all*.

(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%

Table 5: 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.

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. The ratio between these error rates depends on the threshold setting. We will discuss threshold calibration in the next section.

6 General Discussion

We carried out two experiments in order to evaluate methods for utterance selection and utterance verification which are going to be used in a CALL application for low-proficient L2 learners of Dutch. For utterance selection with the transcription of the response in the language model, our best error rates were between 10.0%-6.9% after optimizing acoustic and language models. In 90% of the cases the decoding result was equal to the corresponding transcription of the response (phonetic distance of 0) and in 93.1% of the cases the decoder was able to select a cluster of transcriptions with a phonetic distance of 15 or smaller to the 1-Best in which the corresponding transcription was present.

Using an utterance verifier that combined acoustic likelihoods and duration information of the decoding result, 89.8% of the correctly decoded responses is accepted and 70% of the incorrectly decoded utterances could be rejected when the transcription of the response was present in the language model. In addition, 91.7% of the utterances with no representation in the language model could correctly be rejected.

These results apply when we only perform error detection to the 1-Best decoding result, but as explained in Section 3 error detection will probably be performed on the cluster of responses that have a small phonetic distance to the 1-Best decoding result. For example, if it is not clear whether a segment or a (short) word was pronounced or not, this can be ascertained in the second step through a more detailed analysis [19]. At the moment we think that in the second step 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.

Until now we have evaluated the performance of UV 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, and where the feedback is based on. 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.

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:

$$C_{total} = p_{FR}C_{FR}p_{correct} + p_{FA}C_{FA}(1 - p_{correct}) \tag{10}$$

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 [52]. 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 could make an excellent research vehicle for future experiments with different error costs.

7 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/).

References

[1] D. V. Compernolle, "Recognizing speech of goats, wolves, sheep and..non-natives," *Speech Communication*, vol. 35, no. 1-2, pp. 81–79, 2001.

[2] L. Tomokiyo, "Recognizing non-native speech: Characterizing and adapting to non-native usage in speech recognition," Ph.D. dissertation, Carnergie Mellon University, 2001.

[3] G. Bouselmi, D. Fohr, I. Illina, and J. Haton, "Multilingual non-native speech recognition using phonetic confusion-based acoustic model modification and graphemic constraints," in *Proceedings of ICSLP*, 2006.

[4] M. Benzeghiba, R. D. Mori, O. Deroo, S. Dupont, T. Erbes, D. Jouvet, L. Fissore, P. Laface, A. Mertins, C. Ris, R. Rose, V. Tyagi, and C. Wellekens, "Automatic speech recognition and speech variability: a review," *Speech Communication*, vol. 49, no. 10-11, pp. 763–786, 2007.

[5] J. Flege, M. Munro, and I. MacKay, "Effects of age of second-language learning on the production of english consonants," *Speech Communication*, vol. 16.

[6] O.-S. Bohn and J. Flege, "The production of new and similar vowels by adult german learners of english studies in second language acquisition," vol. 14, no. 2, pp. 131–158, 1992.

[7] A. Bradlow, D. Pisoni, R. Akahane-Yamada, and Y. Tohkura, "Training japanese listeners to identify english /r/ and /l/: Some effects of perceptual learning on speech production," *Journal of the Acoustical Society of America*, vol. 101, no. 4, pp. 229–310, 1997.

[8] A. Neri, C. Cucchiarini, and H. Strik, "Selecting segmental errors in l2 dutch for optimal pronunciation training," *International Review of Applied Linguistics in Language Teaching*, vol. 44, pp. 357–404, 2006.

[9] R. DeKeyser, "What makes learning second language grammar difficult? a review of issues," *Language Learning*, vol. 55, pp. 1–25, 2005.

[10] C. Cucchiarini, H. Strik, and L. Boves, "Different aspects of expert pronunciation quality ratings and their relation to scores produced by speech recognition algorithms," *Speech Communication*, vol. 30, no. 2-3, pp. 109–119, 2000.

[11] ——, "Quantitative assessment of second language learners' fluency by means of automatic speech recognition technology," *Journal of the Acoustical Society of America*, vol. 107, no. 2, pp. 989–999, 2000.

[12] ——, "Quantitative assessment of second language learners' fluency: Comparisons between read and spontaneous speech," *Journal of the Acoustical Society of America*, vol. 111, no. 6, pp. 2862–2873, 2002.

[13] D. Moehle, *A comparison of the second language speech production of different native speakers*. Narr, Tuebingen, 1984.

[14] R. Towell, R. Hawkins, and N. Bazergui, "The development of fluency in advanced learners of french," *Applied Linguistics*, vol. 1, pp. 84–119, 1996.

[15] F. Ehsani, J. Bernstein, and A. Najmi, "An interactive dialog system for learning Japanese," *Speech Communication*, vol. 30, pp. 167–177, 2000.

[16] A. Raux and M. Eskenazi, "Using task-oriented spoken dialogue systems for language learning: potential, practical applications and challenges," in *Proceedings of STILL*, 2004.

[17] H. Bortfeld, S. Leon, J. Bloom, M. Schober, and S. Brennan, "Disfluency rates in conversation: effects of age, relationship, topic, role and gender," *Language and Speech*, vol. 44, no. 2, pp. 123–147, 2001.

[18] W. Menzel, D. Herron, P. Bonaventura, and R. Morton, "Automatic detection and correction of non-native English pronunciations," in *Proceedings of InSTIL*, 2000, pp. 49–56.

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

[20] W. Byrne, E. Knodt, S. Khudanpur, and J. Bernstein, "Is automatic speech recognition ready for non-native speech? a data collection effort and initial experiments in modeling conversational Hispanic English," in *Proceedings of STiLL*, 1998.

[21] M. Gerosa and D. Giuliani, "Preliminary investigations in automatic recognition of English sentences uttered by Italian children," in *Proceedings of InSTIL/ICALL*, 2004, pp. 9–13.

[22] L. Neumeyer, H. Franco, M. Weintraub, and P. Price, "Automatic text-independent pronunciation scoring of foreign language student speech," in *International Conference on Spoken Language Processing*, 1996, pp. 1457–1460.

[23] H. Franco, L. Neumeyer, V. Digalakis, and O. Ronen, "Combination of machine scores for automatic grading of pronunciation quality," *Speech Communication*, vol. 30, pp. 121–130, 2000.

[24] G. Deville, O. Deroo, S. Gielen, H. Leich, and J. Vanparys, "Automatic detection and correction of pronunciation errors for foreign language learners the Demosthenes application," in *Proceedings of Eurospeech*, vol. 2.

[25] G. Kawai and K. Hirose, "A method for measuring the intelligibility and nonnativeness of phone quality in foreign language pronunciation training," in *Proceedings of ICSLP*, 1998, pp. 1823–1826.

[26] S. Witt, "Use of speech recognition in computer assisted language learning," Ph.D. dissertation, University of Cambridge, 1999.

[27] S. Witt and S. Young, "Off-line acoustic modelling of non-native accents," in *Proceedings of Eurospeech*, 1999, pp. 1367–1370.

[28] L. Tomokiyo, "Handling non-native speech in LVCSR: A preliminary study," in *Proceedings of the InSTIL*, 2000, pp. 62–63.

[29] H. Strik and C. Cucchiarini, "Modeling pronunciation variation for ASR: a survey of the literature," *Speech Communication*, vol. 29, no. 2-4, pp. 225–246, 1999.

[30] H. Strik, "Pronunciation adaptation at the lexical level," in *Proceedings of the ISCA Tutorial & Research Workshop (ITRW) Adaptation Methods For Speech Recognition*, J.-C. Juncqua and C. Wellekens, Eds.

[31] S. Goronzy, S. Rapp, and R. Kompe, "Generating non-native pronunciation variants for lexicon adaptation," *Speech Communication*, vol. 42, no. 1, pp. 109–123, 2004.

[32] K. Livescu and J. Glass, "Lexical modeling of non-native speech for automatic speech recognition," in *Proceedings of ICASSP*, 2000.

[33] J. Markowitz, *Using Speech Recognition*. NJ: Prentice Hall, 1996.

[34] F. Ehsani and E. Knodt, "Speech technology in computer-aided learning: Strengths and limitations of a new call paradigm," *Language Learning and Technology*, no. 2, pp. 45–60, 1998.

[35] E. Atwell, D. Herron, P. Howarth, R. Morton, and H. Wick, *Pronunciation training: Requirements and solutions*. ISLE Deliverable 1.4, 1999.

[36] J. Waltje, "CALICO software review - Tell Me More-German," *CALICO Journal*, vol. 19, no. 2, pp. 405–418, 2002.

[37] H. Jiang, "Confidence measures for speech recognition: a survey," *Speech Communication*, vol. 45, pp. 455–470, 2005.

[38] F. Wessel, R. Schluter, K. Mackerey, and H. Hey, "Confidence measures for large vocabulary continuous speech recognition," *IEEE Transactions on Speech Audio Processing*, vol. 9, no. 3, pp. 288–298, 2001.

[39] S. Young, "Detecting misrecognitions and out-of-vocabulary words," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, 1994.

[40] G. Bouwman and L. Boves, "Utterance verification based on the likelihood distance to alternative paths," in *Proceedings of the 5th International Conference on Text, Speech and Dialogue*, 2002, pp. 213–220.

[41] T. Schaaf and T. Kemp, "Confidence measures for spontaneous speech recognition," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, 1997, pp. 875–878.

[42] S. Goronzy, K. Marasek, R. Kompe, and A. Haag, "Prosodically motivated features for confidence measures," in *ASR2000*, vol. 1.

[43] O. Desmukh and A. Verma, "Acoustic-phonetic approach for automatic evaluation of spoken grammar," in *Proceedings of Interspeech*, 2008, pp. 2614–2617.

[44] F. de Wet, C. Cucchiarini, H. Strik, and L. Boves, "Using likelihood ratios to perform utterance verification in automatic pronunciation assessment," in *Proceedings of Eurospeech*, 1999, pp. 173–176.

[45] C. Cucchiarini, J. Driesen, H. V. Hamme, and E. Sanders, "Recording speech of children, non-natives and elderly people for HLT applications: the JASMIN-CGN corpus," in *Proceedings of LREC*, 2008.

[46] K. Demuynck, J. Roelens, D. V. Compernolle, and P. Wambacq, "SPRAAK: an open source SPeech Recognition and Automatic Annotation Kit," in *Proceedings of ICSLP*, 2008, p. 495.

[47] N. Oostdijk, "The design of the spoken dutch corpus," in *New Frontiers of Corpus Research*, P. Peters, P. Collins, and A. Smith, Eds. Rodopi, 2002, pp. 105–112.

[48] C. Cucchiarini, "Assessing transcription agreement: ethodological aspects," *Clinical Linguistics and Phonetics*, vol. 102, pp. 131–155, 1996.

[49] F. Stouten, J. Duchateau, J. Martens, and P. Wambacq, "Coping with disfluencies in spontaneous speech recognition: Acoustic detection and linguistic context manipulation," *Speech Communication*, vol. 48, pp. 1590–1606.

[50] T. Mitchell, *Machine Learning*. McGraw Hill, 1997.

[51] I. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[52] D. A. van Leeuwen and N. Brümmer, "An introduction to application-independent evaluation of speaker recognition systems," in *Speaker Classification*, ser. Lecture Notes in Computer Science / Artificial Intelligence, C. Müller, Ed. Heidelberg - New York - Berlin: Springer, 2007, vol. 4343.