Automatic Pronunciation Error Detection in Repetitor

Eric Sanders [1], *Henk van den Heuvel* [2]

Centre for Language and Speech Technology, Radboud University Nijmegen, the Netherlands
[1] e.sanders@let.ru.nl, [2] h.vandenheuvel@let.ru.nl

Abstract

This paper describes a pronunciation error detection method for Repetitor, a pronunciation training computer program for second language learners of Dutch. A database of L2-speech was constructed and a selection of relevant pronunciation errors for Repetitor was made. Our error detection method is based on a weighted variant selection using forced alignment. Tested on the database, the results show that our method achieves satisfactory detection performance for most pronunciation errors yielding a precision of correct rejects of over 85% for most errors, and scoring accuracies between 85% and 100%.

Index Terms: CAPT, automatic pronunciation error detection

1. Introduction

Automatic Speech Recognition (ASR) can be very usefully applied for Computer Assisted Language Learning (CALL) and, more specifically Computer Assisted Pronunciation Training (CAPT) [1]. A system that assists in students' pronunciation training has some clear advantages over conventional methods: it is faster and cheaper, the student is independent and can work in his/her own pace. However, the technology is still under development [2]. The Centre for Language and Speech Technology (CLST) of the Radboud University Nijmegen has been working on ASR-CAPT for years now in the projects Dutch-CAPT [1] and DISCO [3].

At the IT&C group of Delft University of Technology (TUD), (conventional) second language (L2) training of Dutch is given following the "Delft Method" [4]. The focus of the Delft Method is on hands-on training. Underlying grammatical principles of the language are explained during practising, but are not the explicit object of language acquisition.

The TUD and the CLST combined their expertise in the project Repetitor, in which ASR-CAPT is integrated into the Delft Method. The result is a computer application in which a student repeats sentences that are read aloud by an exemplar speaker and prompted on the screen. The utterance is analysed by an ASR engine and feedback is given automatically with respect to pronunciation errors made by the student. Because of the presently limited capabilities of ASR-based systems to detect pronunciation errors, feedback is restricted to a predefined number of pronunciation errors.

This paper will focus on the method that is used to detect the pronunciation errors. A forced recognition is performed by an ASR engine, where the ASR can choose between a pronunciation variant reflecting the normative, exemplar pronunciation and predefined variants reflecting pronunciation errors.

In this paper we first describe the Repetitor system (section 2) and the speech that was collected with the pronunciation errors that are annotated in the speech data (section 3). Then we explain our pronunciation error detection method (section 4) and we describe how we tested the performance of our method (section 5). After reporting on the results (section 6) we round up with a discussion (section 7) and conclusions (section 8).

2. Repetitor

Repetitor is an extension of an existing program Groenstap developed for L2 acquisition of learners of Dutch by the TUD (IT&C group). It is part of the curriculum called the Delft Method. Basic notions of the method are simplicity and clarity. It is based upon the following principles:

- The core part consists of texts. Words and grammar are acquired by reading and listening to the texts.
- The texts provide plenty of information about the Netherlands and cover subjects that are of importance to foreigners. In a short space of time course participants learn the most frequently used words so that they are quickly able to talk about all kinds of matters.
- Much attention is devoted to the training of listening skills.
- From the start the important grammar is included.
- Grammar is 'explained' using examples and without the use of complicated terminology.

The Delft Method is designed for adults aged 16 and older. It contains a series of lessons about various topics. Each lesson consists of a fixed number of about 25 short sentences, typically in dialogue format. Students can listen to the sentences as spoken by an exemplar speaker, check word meanings and fill in missing words (offered as cloze).

Students have the option to go through a lesson sentence by sentence. After listening to a sentence they can read out and record their own pronunciation and listen to it. In Repetitor an ASR engine is incorporated. The recorded sentences are sent to the ASR and checked for a number of preselected pronunciation errors. Detected errors are shown on screen and marked at the corresponding locations in the orthography of the sentence. The student can listen to the exemplar speaker once more and compare it to the recording of his own pronunciation. The student can repeat the sentence if desired. All recordings are saved with a logging of the detected pronunciation errors.

3. Pronunciation Errors

A speech database was compiled containing the recordings of 42 students with a wide variety in language backgrounds practicing with the Delft Method. The database comprises recordings of 10 different lessons with 19-32 utterances each. In total 74 lessons with 1968 utterances were recorded.

For each utterance we created a phonemic transcription in SAMPA [5]. The transcription corresponds to the speech of the exemplar speaker (from now: exemplar pronunciation) and is used as the reference transcription to which the pronunciation of the students is compared.

The utterances of the students that were recorded were also transcribed. Four teachers of the Delft Method listened to the speech and judged the pronunciation of each word. If the pronunciation was not acceptable in the view of the teachers, they adjusted the transcription to reflect the perceived (incorrect) pronunciation. The four transcribers discussed about their decisions during the process, but there is no overlap in transcriptions allowing to measure intra- and inter-transcriber agreement. The transcriptions were analysed by the transcribers/teachers and a long-list of pronunciation errors was compiled. This list contains 158 types of different pronunciation errors accompanied with the context in which the error occurs.

From this list a ranking of pronunciation errors was made. Together with the teachers we selected the errors that they considered most important, and that could feasibly be detected on the basis of their phonetic and acoustic properties. Excluded from detection were e.g. insertion of /h/, deletion of /r/, and substitution of /t/ by /d/. For the acoustical models used in our ASR method, these distinctions are simply too subtle. Similar pronunciation errors were grouped together to overcome data sparseness (e.g. /Y/ pronounced as /u/; /y/ pronounced as /u/; and /9y/ pronounced as /u/ were taken together). Some errors have a broad context (e.g. 'before a vowel'), others have a narrow context (e.g. only in a few words). The selection procedure resulted in a short list of 25 pronunciation errors, that were implemented in Repetitor. Transcriptions of words possibly containing one of these pronunciation errors were checked once more. In this paper we present results of experiments on the set of 15 errors that appear at least 15 times in our database. An overview of the selected pronunciation errors can be found in Table 1.

4. Pronunciation Error Detection Method

To automatically detect the pronunciation errors, a forced recognition is carried out with a language model (LM) that allows only the correct words in the correct order, but permits different pronunciation variants per word. One variant reflects the exemplar pronunciation and other variants contain pronunciation errors. Probabilities are attached to the pronunciation variants in the LM, so that the probability of a variant with a pronunciation variant being selected by the ASR becomes smaller. This way the ASR needs strong acoustic evidence to select a variant with a pronunciation error.

The variants with pronunciation errors are generated by a set of rewrite rules that are derived from the pronunciation errors in Table 1. If the phonemic transcription of a word matches the pattern of a rewrite rule, the rule is applied and the new variant is added to the LM and to the pronunciation lexicon. Several rules can be applied to one word, generating many variants. Each rewrite rule has a weight between 0 and 1, and the weight of a generated variant is multiplied by this rule weight. The weight of the variant reflecting the exemplar pronunciation is always 1, so the weight of the variant(s) reflecting a pronunciation error is always equal or less than that of the exemplar pronunciation. The optimal rule weights have to be determined experimentally. The LM probabilities of the variants are the same as their weights, but scaled in such a way that the probabilities of all variants (including the one reflecting the exemplar pronunciation) for a word sum up to 1.

For example: If we take the word 'bus', which has exemplar transcription /bYs/ and we use pronunciation error rule M the variant /bus/ is created. If we use a rule weight of e.g. 0.5 we would have the following variants in our LM (and lexicon):
/bYs/ with a probability of 0.67 (1/(1+0.5))
/bus/ with a probability of 0.33 (0.5/(1+0.5))

The speech recognition system we use is SPRAAK [6], an open source speech recognition package developed within the STEVIN-programme [7]. The acoustic models (AM) were trained on 42 hours of read speech (spoken books) of the Spoken Dutch Corpus (CGN) [8]. We trained 47 3-state Gaussian Mixture Models (GMM). GMMs were trained using a 32 ms Hamming window, with a 10 ms step size. Acoustic feature vectors consisted of 13 mel-based cepstral coefficients, including c[0], plus their first and second order derivatives. In our experiments we used monophone AM, which in preliminary tests appeared to perform equally well as triphone AM for this task.

5. Experiments

For the evaluation of the experiments in this paper, we tested our pronunciation error detection method of the different pronunciation errors in isolation (one at the time), because there is not enough data to test combined pronunciation errors. Data sparseness is also the reason why we could not divide the data in a separate training and test set. For each pronunciation error we carried out the following experiment on the Repetitor data described in section 3: For each utterance, a lexicon and LM was created with the exemplar transcription of all words. For each word in an utterance, a pronunciation variant was generated from the exemplar transcription, if the rewrite rule for the pronunciation error was applicable. The variant was then added to the lexicon and LM. Next a forced recognition was carried out. After recognition the result for each target word falls in one of four categories:

- Correct Accept (CA): the word was pronounced correctly and the recogniser chooses the exemplar pronunciation
- Correct Reject (CR): the word was mispronounced and the recogniser chooses the variant reflecting the mispronunciation
- False Reject (FR): the word was pronounced correctly but the recogniser chooses the variant reflecting the mispronunciation
- False Accept (FA): the word was mispronounced but the recogniser chooses the exemplar pronunciation

Each pronunciation error detection was tested separately with 5 different rule weights (0.1, 0.3, 0.5, 0.7, 1) for the rewrite rule. The higher the weight for a rule is, the higher the probability of a pronunciation variant. A higher weight will result in more false rejects, but less false accepts.

Based on four categories described above we derived the following measures of system performance (per rule):

- Recall of CA = 100 x CA/(CA+FR)
- Recall of CR = 100 x CR/(CR+FA)
- Precision of CA = 100 x CA/(CA+FA)
- Precision of CR = 100 x CR/(CR+FR)
- Scoring Accuracy = 100 x (CA+CR)/(CA+CR+FA+FR)

Only words that could contain the pronunciation error are included in the computation.

Table 1: *Targeted pronunciation errors implemented in our detection method with the number of pronunciation errors in the database.*

Error	Description	Example	#
A	final /t/ deletion after sonorant or vowel, next word starts with sonorant or vowel (or at end of sentence)	Nederland (*the Netherlands*): /ned@rlAnt/ ⇒ /ned@rlAn/	73
B	/@/ as /e/ in the words 'een', 'je' and 'we'	je (*you*): /j@/ ⇒ /je/	99
C	/@/ as /i/ in the words 'we', 'ze', 'de', 'me' and in words that start with /b@/ and in words that end with /@x/	twintig (*twenty*): /twInt@x/ ⇒ /twIntix/	70
D	/@/ as /Ei/ in the words 'we', 'ze', 'je', 'me', 'mijn', 'een', and in words that end with /l@k/	mijn (*my*): /m@n/ ⇒ /mEin/	95
E	/9y/ as /Au/ before /t/ or /s/	thuis (*home*): /t9ys/ ⇒ /tAus/	50
F	vowel as /Aj/ in the words 'negen', 'een', 'drie', 'iets', 'precies', 'les'	precies (*precise*): /pr@sis/ ⇒ /pr@sAjs/	16
G	/@t/ as /hEt/ and /@m/ as /hEm/	het (*it*): /@t/ ⇒ /hEt/	153
H	final /t/ insertion after sonorant or vowel if the next word starts with sonorant or vowel (or at end of sentence)	ben je (*are you*): /bEn j@/ ⇒ /bEnt j@/	17
I	/x/, /G/, /k/ of /g/ insertion after /N/, before a vowel, or word final, if next word starts with vowel (or at end of sentence)	Engeland (*England*): /EN@lAnt/ ⇒ /ENg@lAnt/	16
J	/k/ or /g/ as /x/ or /G/ before a vowel	winkel (*shop*): /wInk@l/ ⇒ /wInx@l/	25
K	/k/ or /g/ as /x/ or /G/ after a vowel	boek (*book*): /buk/ ⇒ /bux/	52
L	/x/ or /G/ as /k/ or /g/ within word between sonorants and/or vowels, or at begin/end of word if next/previous word ends/begin with sonorant or vowel, or at begin/end of sentence	volgende (*next*): /vOlG@nd@/ ⇒ /vOlg@nd@/	22
M	/9y/, /Y/ or /y/ as /u/	suiker (*sugar*): /s9yk@r/ ⇒ /suk@r/	43
N	word beginning /x/ or /G/ as /k/	groen (*green*): /xrun/ ⇒ /krun/	22
O	/x/ or /G/ as /k/ after /s/	school (*school*): /sxol/ ⇒ /skol/	17

6. Results

As an example, Table 2 shows the results of pronuncation error B for all five rule weights. Shown are the number of CA, CR, FA, FR, recall of CA and CR, (R(CA) and R(CR)), precision of CA and CR (P(CA) and P(CR)) and Scoring Accuracy (SA).

Table 2: *Results of pronunciation error B for all rule weights.*

Rule weight			

0.1	CA=409
	FR=2
	R(CA)=99.5%
		FA=84
		CR=15
		R(CR)=15.2%
			P(CA)=83.0%
			P(CR)=88.2%
			SA=83.1%

0.3	CA=397
	FR=14
	R(CA)=96.6%
		FA=55
		CR=44
		R(CR)=44.4%
			P(CA)=87.8%
			P(CR)=75.9%
			SA=86.5%

0.5	CA=375
	FR=36
	R(CA)=91.2%
		FA=31
		CR=68
		R(CR)=68.7%
			P(CA)=92.4%
			P(CR)=65.4%
			SA=86.9%

0.7	CA=348
	FR=63
	R(CA)=84.7%
		FA=22
		CR=77
		R(CR)=77.8%
			P(CA)=94.1%
			P(CR)=55.0%
			SA=83.3%

1	CA=300
	FR=111
	R(CA)=73.0%
		FA=14
		CR=85
		R(CR)=85.9%
			P(CA)=95.5%
			P(CR)=43.4%
			SA=75.5%

In total there are 510 words in which the pronunciation error could occur. Of these 411 (CA+FR) were pronounced correctly and 99 (FA+CR) erroneously. The results show clearly that a higher rule weight yields a substantially higher number of correct rejects but at the cost of many more false rejects.

For all 15 pronuncation errors only the results of the optimal rule weight are given in Table 3. In line with [1] we consider as the optimal weight per rule that weight that yielded a recall of CA $>$ 90% and a maximum SA for that rule. A recall of CA $>$ 90% means that in maximum of 10% a correct pronunciation is falsely rejected. As can be concluded from Table 2 the optimal rule weight for rule B is 0.5.

Table 3: *Recall, Precision and Scoring Accuracy for each pronunciation error with the best performing rule weight.*

Error	Rule wt.	R(CA) (%)	R(CR) (%)	P(CA) (%)	P(CR) (%)	SA (%)
A	0.7	97.3	65.8	96.7	70.6	94.6
B	0.5	91.2	68.7	92.4	65.4	86.9
C	0.5	99.4	75.3	97.8	91.2	97.4
D	0.3	94.6	56.8	92.6	65.1	89.0
E	0.3	90.0	98.0	98.4	87.5	93.3
F	0.7	99.7	75.0	98.7	92.3	98.4
G	0.7	91.3	86.3	50.0	98.5	86.9
H	0.1	100	11.8	99.6	100	99.6
I	1	98.3	87.5	96.6	93.3	95.9
J	0.3	98.6	76.0	99.1	67.9	97.8
K	0.5	99.1	78.9	98.6	85.4	97.9
L	0.5	99.6	77.3	99.3	85.0	98.8
M	0.3	99.4	65.1	97.0	90.3	96.6
N	0.3	99.8	90.9	99.5	95.2	99.3
O	1	98.0	94.1	98.0	94.1	97.0

For all pronunciation errors the recall of CA is over 90%. The recall of CR is between 75 and 100% for most errors, but somewhat lower for errors A, B, D and M and very low for error H. Precision of CA is over 90 % for all errors, except G, for which it is 50%. Precision of CR is over 85% for most errors, but somewhat lower (around 70%) for rules A, B, D and J. SA is over 85% for all errors, often near 100%.

7. Discussion

The scores for error H look somewhat strange: very high recall of CA, precision of CR and SA, but a very low recall of CR. This is due to the high number of words in which the error could occur, but that were pronounced correctly (3436) compared to the number of words that were mispronounced (17). A higher rule weight will improve error detection, but at the cost of many more false rejects, which will decrease especially the precision of CR considerably. A recall of CA of 99% would mean 35 false rejects, which is already twice the number of pronunciation errors in the whole set.

The opposite is the case for error G. In this case there are 23 words that were pronounced correctly and 153 mispronunciations, which leads to a relatively low precision of CA.

These examples illustrate two ways to look at the results. One way is to look at how well the error detection works in terms of recall. This tells us how good the detection method is in selecting the correct variant. The other way to look at the results is in terms of precision. This tells us how well the whole pronunciation error feedback works, but it is very dependent on the data i.e. on the number of mispronunciations compared to the number of exemplar pronunciations. By looking at the scoring accuracy, this is partly avoided.

Commonly used pronunciation error detection methods based on post-processing, like GOP [9], primarily test the quality of a phone and detecting insertion or deletion errors is not straightforward, because the phones of the correct pronunciation are judged with confidence scores and inserted or deleted phones are not found. With the method we used we detect the phenomenon that took place (insertion, deletion, substitution) for those pronunciation errors for which a variant is added to the words in the lexicon and LM. In the method described in [10] words with pronunciation variants are added to the lexicon as well, but no weights are assigned to the different variants. Our method is relatively easy to implement, pronunciation errors can easily be added, by adding new rewrite rules and it works fast. The disadvantage is, that it is capable of detecting pronunciation errors only in words for which a variant is added to the lexicon and the LM. It could very well be used as a first pass before other error detection methods. The variant chosen in the first pass can be used as hypothesis in the second pass.

In our method it is easy to adapt the set of pronunciation rules, to the language background of the student who is using the system. Specific errors are more likely depending on the mother tongue of the speaker. Making the system L1-dependent by leaving out unlikely errors might reduce confusability and increase system performance.

In our method, we used acoustic models of native Dutch speakers only. The results may be further improved by adapting the AMs with data of non-native speakers of Dutch [11]. [12] presents a method which includes acoustic models from L1 of non-native speakers. The idea is that the choice of an L1-model by the system points to a mispronunciation. Also this approach would fit in a L1-dependent version of the system.

In the experiments described in this paper, the pronunciation errors were tested in isolation, because there are not enough occurrences of words with multiple pronunciation errors. In the implementation of our Repetitor system variants with combinations of all applicable pronunciation errors are added to lexicon and LM. It is hard to predict in what sense combining the pronunciation errors will influence their detection.

8. Conclusions

In this paper we presented a new method to detect pronunciation errors. It is based on forced recognition and weighted pronunciation variant selection: one of the variants is the correct realisation as spoken by the exemplar speaker while the other variants represent predefined mispronunciations generated by rule and selected on the basis of mispronunciation data from L2 students. Pronunciation variant selection is driven by probabilities that are attached to the variants and integrated into the LM. The method has been applied by using native acoustic models and performs well on non-native data. On a database of 2000 utterances, fifteen pronunciation errors are detected with a recall of correct accepts of over 90%, a recall of correct rejects between 75 and 100%, a precision of correct accepts of over 90% for all errors but one, a precision of correct rejects of over 85% for most errors, and scoring accuracies between 85% and 100%.

9. Acknowledgements

This work is granted by the Delft University of Technology. The Repetitor speech database was recorded by teaching members of the IT&C group of the TUD: Alied Blom, Piet Meijer, Sonja van Boxtel and Conny Wesdijk. They also indicated the pronunciation errors in the broad phonetic transcriptions of the recordings.

10. References

[1] Cucchiarini, C., Neri, A., and Strik, H., “Oral proficiency training in Dutch L2: The contribution of ASR-based corrective feedback”, Speech Communication, Volume 51, Issue 10, pp. 853-863, 2009.

[2] Strik, H., Truong, K., De Wet, F. and Cucchiarini, C. “Comparing different approachesfor automatic pronunciation error detection”, Speech Communication, Volume 51, Issue 10, pp. 845-852, 2009.

[3] Cucchiarini, C., van Doremalen , J. and Strik, H., “DISCO: Development and Integration of Speech technology into Courseware for language learning”, In Proceedings Interspeech, pp. 2791-2794, Brisbane, 2008.

[4] www.delftsemethode.nl

[5] www.phon.ucl.ac.uk/home/sampa/dutch.htm

[6] Demuynck, K., Roelens, J., Van Compernolle, D. and Wambacq, P., “SPRAAK: An Open Source SPeech Recognition and Automatic Annotation Kit”, In Proceedings Interspeech, page 495, Brisbane, 2008.

[7] www.stevin-tst.org

[8] Oostdijk, N., Goedertier, W., Van Eynde, F., Boves, L., Martens, J.-P., Moortgat, M. and Baayen, H. “Experiences from the Spoken Dutch Corpus project”, In Proceedings LREC, pp. 340-347, Las Palmas, 2002.

[9] Witt, S.M., “Use of speech recognition in Computer- assisted Language Learning”, Phd thesis, Department of Engineering, University of Cambridge, 1999.

[10] Kim, J., Wang, C., Peabody, M., Seneff, S., “An Interactive English Pronunciation Dictionary for Korean Learners”, In proceedings of Interspeech, pp. 1677-1680, Jeju, 2004.

[11] Mayfield Tomokiyo L., Waibel, A. “Adaptation Methods for Nonnative Speech”, In Proceedings of Multilinguality in Spoken Language Processing, Aalborg, 2001.

[12] Kawai, G., Hirose, K. “A Method for Measuring the Intelligibility and Nonnativeness of Phone Quality in Foreign Language Pronunciation Training”, In Proceedings of ICSLP, pp. 1823-1826, Sydney, 1998.