The Goodness of Pronunciation Algorithm: a Detailed Performance Study

Sandra Kanters [1], *Catia Cucchiarini* [2], *Helmer Strik* [2]

[1] Customer Contact Solutions, Logica, The Netherlands
[2] Department of Linguistics, Radboud University Nijmegen, The Netherlands
sandra.kanters@logica.com, [c.cucchiarini|h.strik]@let.ru.nl

Abstract

An inventory was compiled of pronunciation errors frequently made by foreigners speaking Dutch. On the basis of this inventory artificial errors were created in a native development corpus, which in turn were used to optimize thresholds for the Goodness of Pronunciation (GOP) algorithm. In the current study the GOP algorithm is evaluated in three different ways: (1) using a native test corpus with artificial errors which reflect errors frequently made by non-natives, (2) within an actual application used by non-natives for practicing pronunciation, and (3) post-hoc, using the recorded interactions of the pronunciation training application, to determine what the performance of the algorithm would have been if optimal speaker and phone specific thresholds had been used.

The results show that the performance of the GOP algorithm was satisfactory and that the procedure by which thresholds were determined by simulating realistic pronunciation errors was appropriate, because performance on the artificially introduced errors closely approximated performance on real data. This finding is particularly welcome if we consider that, in general, paucity of data is a common problem in this kind of research. Furthermore, it appeared that post-hoc threshold optimization only led to a slight increase in performance.

Index Terms: Goodness of Pronunciation (GOP), pronunciation error detection, Computer Assisted Pronunciation Training (CAPT)

1. Introduction

Research on second language (L2) acquisition has indicated that exposure to a second language might not be sufficient for L2 learning (e.g., [1]), especially for adult L2 learners. Relevant in this respect are Swain's output hypothesis [1], which emphasizes the role of output in L2 learning, and Schmidt's [2] 'noticing hypothesis', which underlines that awareness of discrepancies between the learner's output and the L2 is necessary for the acquisition of a specific linguistic item. Since exposure to the L2 and L2 output will not automatically guarantee this kind of awareness, corrective feedback is required to make learners aware of their errors and stimulate them to attempt self-improvement [3].

In pronunciation learning corrective feedback is particularly required because very often learners are not aware of the pronunciation errors they make. On the other hand, providing individual corrective feedback on pronunciation is particularly time-consuming for teachers, with the result that the amount of practice that is needed is almost never achieved in the classroom.

Computer Assisted Language Learning (CALL) systems that make use of Automatic Speech Recognition (ASR) seem to offer an alternative for practicing pronunciation, because they can offer specific feedback on individual errors and extra time for practicing at the learners' own tempo. An important requirement is then that the feedback provided be helpful. In part this is determined by the accuracy of the feedback. If learners receive inaccurate feedback (pronunciation errors are indicated where actually no errors occur, or pronunciation errors are missed) they are less likely to actually improve their pronunciation.

Corrective feedback on pronunciation can be given on different aspects. In this paper we focus on corrective feedback on the phoneme level. Providing this kind of detailed feedback is considerably more challenging than providing corrective feedback on a more global level such as word or sentence level. As a matter of fact, for providing global feedback pronunciation measures can be used that are calculated over longer stretches of speech, and therefore more data points, while detailed feedback at the segmental level requires computing a score for each individual realization of a given phone.

Various approaches to segmental error detection can be found in the literature. The best known example is the Goodness Of Pronunciation (GOP) algorithm proposed by Witt [5], [6]. The GOP algorithm calculates the likelihood ratio that the realized phone corresponds to the phoneme that should have been spoken according to the canonical pronunciation. Thresholds, calculated beforehand, are used to decide which likelihood ratio scores corresponded to mispronounced sounds. The GOP algorithm was applied in the Dutch-CAPT system [7] [10], a system designed to provide corrective feedback on a selected number of speech sounds, referred to as target phonemes, which had appeared to be problematic for learners of Dutch from various first language backgrounds: /ʏ/, /x/, /ɑ/, /y/, /œy/, /aː/, /ɛi/, /h/, /u/, /øː/, /ɪ/ [4], [7]. This inventory of errors was determined on the basis of three non-native corpora (not including the Dutch-CAPT corpus) [4].

In the current study the GOP algorithm is evaluated in three different ways to get insight into how GOP scores vary as a function of different parameters, in particular threshold values. The ultimate aim is to determine whether and how pronunciation error detection can be improved. In short, the three procedures are the following (more details are provided in sections 2.4.1., 2.4.2, and 2.4.3, resp., regarding the methodology, while the results are presented in sections 3.1, 3.2, and 3.3, resp.):

(1) Since not enough non-native material was available, as is often the case, GOP thresholds for the target phonemes were optimized by creating artificial pronunciation errors in native data. In our case, these artificial errors reflect the errors frequently made by foreigners [4], [7]. Thresholds per phoneme were optimized on one set, and tested on another set of native data.

(2) These thresholds were employed in an actual application, the Dutch-CAPT system, which was used by non-

natives. All interactions were recorded and evaluated afterwards [7] [10].

(3) Finally, we also tested, post-hoc, what the performance of the algorithm would have been if we had used speaker specific thresholds for all phones that are optimal for the current data.

The most relevant innovative aspects of the current study are that the GOP algorithm is evaluated for foreigners speaking Dutch, the thresholds are optimized using a native development corpus with artificial errors that reflect errors frequently made by foreigners (based on an inventory made using other corpora), and finally that the GOP algorithm is evaluated in three different ways: (1) using an independent native test corpus with realistic artificial errors, (2) in an actual application used by non-natives, and (3) and also post-hoc using speaker-specific thresholds.

2. Method

2.1. Material

The inventory of pronunciation errors was based on three corpora of non-native speech (for more details see [4], [7]). Speech from three other, non/overlapping corpora was used to form the databases of this study. Two corpora were sub-corpora of the Spoken Dutch Corpus (Corpus Gesproken Nederlands; CGN), a corpus of about 9 million words that constitutes a plausible sample of standard Dutch as spoken in the Netherlands and Flanders and contains various annotation layers [8]. We chose two sub-corpora of Dutch spoken by native speakers from the Netherlands, one was used as development corpus (CGN-dev) and an independent one as test corpus (CGN-test).

The last corpus contains the speech material that was collected through Dutch-CAPT and consists of interactions between non-native language learners and the Dutch-CAPT training system. The learners had different native languages. This material was manually annotated for pronunciation errors.

The performance of the algorithm was investigated for the 11 target phonemes. The databases were formed with all realizations of these phonemes. This made up a total of 92,798 realizations for CGN-dev, 191,147 realizations for CGN-test (about 50% are errors) and 1,806 for Dutch-CAPT (about 42% are errors).

2.2. The Goodness of Pronunciation algorithm

The GOP algorithm [5], [6] calculates the likelihood ratio that a phone realization corresponds to the phoneme that should have been spoken (the so-called GOP score). The student's speech is subjected to both a forced and a free speech recognition phase. During forced recognition a known orthographic transcription of the speech signal is used to force the recognition of the speech and in the free recognition phase the phoneme sequence most likely to be spoken is calculated. A GOP score of a specific phone realization is then calculated by taking the absolute difference of the log probability of the forced and the log probability of the free recognition phase. Phones with GOP scores above a pre-defined threshold are probably mispronounced and are for this reason rejected by the algorithm. Likewise, phones with scores lower than the pre-defined threshold will probably be well-pronounced and are accepted.

2.3. Performance measures

A classification algorithm like the GOP can produce four types of outcomes: 1) correctly accepted (CA) phone realizations, i.e. phones that were pronounced correctly and were also judged as correct; 2) correctly rejected (CR) phone realizations, i.e. phones that were pronounced incorrectly and were also judged as incorrect; 3) mispronunciations that were falsely judged as being correct (FA: False Accept) and 4) correct pronunciations that were falsely flagged as mispronunciations (FR: False Reject). To achieve optimal performance the algorithm should detect the mispronunciations and, at the same time, it should not flag as mispronunciations those realizations that were actually correct. For this reason both the amount of correctly rejected (CR) and correctly accepted (CA) realizations are important in the performance calculation.

The performance of an error detection algorithm can be calculated in different ways. One way is to measure the scoring accuracy (SA), which is calculated by formula (1) shown below:

$$SA = ((CA+CR) / (CA + CR + FA + FR)) * 100 \quad (1)$$

Other widely used measures for calculating the performance of a classification algorithm are precision, recall and the F-measure. These metrics can be calculated both for the correct accepts and the correct rejects (see (2) - (6)).

$$\text{Precision of CA} = (CA / (CA + FA)) * 100 \quad (2)$$

$$\text{Precision of CR} = (CR / (CR + FR)) * 100 \quad (3)$$

$$\text{Recall of CA} = (CA / (CA + FR)) * 100 \quad (4)$$

$$\text{Recall of CR} = (CR / (CR + FA)) * 100 \quad (5)$$

$$\text{F-measure} = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (6)$$

2.4. Analyses

2.4.1. Establishing thresholds

The aim of this exercise was to find GOP thresholds that maximize SA while keeping FR below 10%. The rationale behind this decision was that erroneously rejecting correct pronunciations would be more detrimental for learners than erroneously accepting mispronunciations.

Optimal GOP thresholds were established in the following way. First, since we did not have enough non-native speech material at our disposal, pronunciation errors were simulated by changing the phonemic representations in the lexicon of the native speech corpus. The artificial errors were introduced in the pronunciation dictionary for the 11 target phonemes, phone by phone. For each phone, for half of the entries containing that phone, the correct pronunciation (i.e. phone) was replaced by an incorrect pronunciation (i.e. another phone). The scheme according to which correct phones were replaced by erroneous ones was based on information that we had collected on how Dutch phones are frequently mispronounced by L2 learners [7] [10]. Optimal thresholds were then established for each phoneme-gender combination by carrying out an exhaustive search. Preliminary experiments had shown that a step size of about 0.25 was sufficient, since generally there is a range of threshold values for which the values of SA do not differ significantly. The GOP thresholds were established by using the development corpus CGN-dev and were evaluated on the independent test corpus CGN-test (see results in Section 3.1).

2.4.2. *Performance on Dutch-CAPT*

The thresholds obtained for the CGN-dev corpus (see Section 2.4.1.) were used in the Dutch-CAPT system. In order to get insight into the performance of the GOP algorithm on speech by non-native speakers, GOP scores were calculated for the speech collected from the users of the Dutch-CAPT system [9] [10]. Other than in the Dutch-CAPT system, where a maximum of three pronunciation errors per utterance was indicated, in this study GOP scores were calculated for all pronunciation errors. The performance was measured in SA and in precision, recall and F-measure of the correct accepts and the correct rejects.

2.4.3. *Threshold optimization*

Threshold optimization was carried out post-hoc on the Dutch-CAPT material with the aim of finding out whether the performance of the algorithm could be optimized by using thresholds on a more specific level. Instead of using thresholds for each phoneme-gender pair, thus pooling speakers of the same gender, the performance was measured with phoneme-speaker dependent thresholds, therefore for each separate speaker.

First, for each phoneme-speaker pair the threshold which yielded the highest SA for that specific pair was calculated. Threshold values in between a specific range were used to find those optimal thresholds. Performance was then calculated for each speaker separately. Subsequently, the values for the various speakers were combined to obtain measurements for the whole group.

3. Results

3.1. Establishing thresholds

Optimal thresholds were established for each phoneme-gender combination. The GOP thresholds were determined by means of CGN-dev (see Section 2.4.1), and evaluated on CGN-test. The average evaluation results are shown in the second column of Table 1. It can be observed that all performance values (SA, precision, recall, and F) are higher than 80%. The goal was to find GOP thresholds for which SA was high and FR remained below 10%. The FR value in Table 1 is indeed smaller than 10%. The percentage of artificial pronunciation errors in the material is about 50%. It can be seen that the performance of the algorithm for the correct and incorrect phonemes does not differ much, since CA and FA do not differ much from CR and FR, respectively.

3.2. Performance on Dutch-CAPT

In Table 1, third column, the performance results for the Dutch-CAPT database are presented. These results show that SA was 81.51%. For the performance measures precision, recall, and F-measure slightly higher percentages were obtained for correct accepts than correct rejects. Remarkably, these values for realistic errors of non-natives do not differ much from those for artificial errors in native data.

3.3. Threshold optimization

In Table 1, fourth column, the results of the threshold optimization analysis are presented. The performance values are all higher than those in column three. However, if one considers that this is the best that can be obtained (post-hoc) for this method, it can be concluded that the thresholds obtained with the method using realistic artificial errors in native data appear to work very well.

Table 1. *The number of phoneme realizations, their distribution into CA, CR, FA, and FR, and the performance results on CGN-test, Dutch-CAPT and Dutch-CAPT (optimized)*

	CGN-test	Dutch-CAPT	Dutch-CAPT (optimized)
Tot # realizations	191,147	1,806	1,806
CA	40.25 %	49.67 %	51.61 %
CR	41.42 %	31.84 %	35.99 %
FA	8.54 %	10.41 %	6.26 %
FR	9.79 %	8.08 %	6.15 %
SA	81.67 %	81.51 %	87.60 %
Precision of CA	82.49 %	82.67 %	89.19 %
Recall of CA	80.43 %	86.00 %	89.36 %
F-measure of CA	81.45 %	84.30 %	89.27 %
Precision of CR	80.88 %	79.75 %	85.41 %
Recall of CR	82.90 %	75.36 %	85.19 %
F-measure of CR	81.88 %	77.49 %	85.30 %

4. Discussion

In this paper the performance of the GOP algorithm was studied to get insight into how GOP scores vary as a function of different parameters, in particular threshold values. The ultimate aim was to determine whether and how pronunciation error detection could be improved.

The performance of the algorithm was studied for the 11 target phonemes using three databases. CGN-dev was used to determine threshold values for each phoneme-gender pair. With these thresholds the performance of the algorithm was calculated on a database of Dutch spoken by native speakers in which pronunciation errors had artificially been added (CGN-test) and on a database of Dutch spoken by non-natives (Dutch-CAPT), which had been manually annotated for pronunciation errors.

The performance of the GOP algorithm was measured in SA and in precision and recall of CA and CR. The results for CGN-test showed that SA was about 82%, and that precision and recall percentages were roughly the same. Also for Dutch-CAPT SA was about 82%, but precision and recall of CA were slightly higher (83% and 86%, respectively), and precision and recall of CR were slightly lower (80% and 75%, respectively).

Although both SA and precision and recall measure the performance of the algorithm, they analyze it from different perspectives. SA shows the percentage of correct classifications (CA and CR) versus incorrect classifications (FA and FR), but it does not focus on either correct accepts or correct rejects, which precision and recall do. Both in CGN-test and Dutch-CAPT about half of the phones are mispronounced, and FA and FR do not differ much. This explains why the performance measures do not differ considerably for the two corpora.

A post-hoc threshold optimization analysis on the Dutch-CAPT data showed that using more specific thresholds, i.e. thresholds for each phoneme-speaker pair, yielded slightly better performance. For each phoneme-speaker pair

thresholds which optimized SA for that specific pair were calculated. With these new thresholds the performance in SA, and in precision and recall of CA and CR was measured. This analysis resulted in an SA of approximately 88%. Precision and recall of CA were 89%, and precision and recall of CR were 85%.

Compared to the research by Witt [5] it has to be concluded that lower SA percentages are obtained here (90% against 82%, respectively), but probably this lower performance can be explained by the fact that we used a more realistic simulation of the real world, and consequently the task was more difficult.

Witt does not mention which and how many different phonemes she used in creating artificial errors. We calculated the performance on the 11 phonemes that tend to be difficult for language learners and that were addressed in Dutch-CAPT. Second, while simulating the pronunciation errors we first checked how Dutch phones are usually mispronounced and used this information in changing the phonemic representations. Witt, on the other hand, created artificial errors by replacing in the lexicon all realizations of a given phoneme, say /a/ by another one, say /i/. However, the chance that language learners will make that type of error is smaller than that they will confuse or mispronounce phonemes that are acoustically more similar such as /i/ and /I/, or /x/ and /k/. Likewise, the GOP algorithm will have a harder time in distinguishing /i/ from /I/ and /x/ from /k/ than in distinguishing /a/ from /i/. This might explain the higher SA values obtained by Witt for the native material.

The result that the accuracy measures for CGN-test are not very different from those of Dutch-CAPT indicates that the performance on real data approximates the performance on artificially introduced pronunciation errors. In other words, the procedure by which thresholds were determined worked properly. This is also partly related to the choice of the simulated errors we made, as these were based on knowledge about pronunciation errors that L2 learners actually make. This finding is particularly reassuring because in this kind of research data sparseness is just a fact of life. These outcomes show that when real data are not available they can at least be simulated with satisfactory results. Unfortunately, we cannot compare our results for non-natives to those of Witt, because Witt did not present SA results for non-natives.

The finding that post-hoc threshold optimization only led to a slight increase in performance can be explained by the fact that the GOP scores of well-pronounced and mispronounced sounds overlap to a considerable extent. In other words, whichever threshold is chosen, there will always be False Accepts and/or False Rejects. For this reason, the solution in improving the performance has to be sought in using speech characteristics for which such an overlap is minimized. A possible way of doing this is by enhancing the GOP algorithm with acoustic-phonetic information. With the latter approach results have been obtained that are even better than GOP results [9]. With this approach specific phoneme characteristics can be included, which could perhaps help the algorithm to better detect which sounds are correctly or incorrectly pronounced.

5. Conclusions

From the results presented in this paper we can draw the following conclusions. First, the performance of the GOP algorithm of 80-90% is satisfactory. Second, the procedure by which thresholds were determined was appropriate, because performance on artificially introduced pronunciation errors closely approximated performance on real data. This finding is particularly welcome if we consider that, in general, paucity of data is a common problem in this kind of research. Our results indicate that, in the absence of real data, acceptable results can be obtained by simulating pronunciation errors in a realistic way. Third, the performance of the algorithm could be improved (slightly) by taking more specific thresholds. Although adopting thresholds for each phoneme-speaker pair will not be easily feasible in practice, it is worth investigating whether groups of speakers can be formed to which the same thresholds can be applied (e.g. speakers with the same or comparable native languages). Fourth, as threshold optimization only led to a slight increase in performance, it is clear that other ways have to be found to improve the performance of the GOP algorithm, for instance by including acoustic-phonetic information (e.g., [9]) that better models specific phoneme characteristics.

6. Acknowledgements

We are indebted to Febe de Wet for her work on establishing the GOP thresholds and to Ambra Neri who collected the Dutch-CAPT speech material and its annotations.

7. References

[1] Swain, M., “Communicative competence: some roles of comprehensible input and comprehensible output in its development”, in *Input in Second Language Acquisition*, Gass, M.A. and Madden, C.G. [Eds.], Rowley MA: Newbury House, pp. 235-253, 1985.

[2] Schmidt, R.W., “The role of consciousness in second language learning”, *Applied Linguistics*, vol. 11, pp. 129-158, 1990.

[3] Havranek, G., “When is corrective feedback most likely to succeed?”, *International Journal of Educational Research*, vol. 37, pp. 255-270, 2002.

[4] Neri, A., Cucchiarini, C., Strik, H., “Selecting segmental errors in L2 Dutch for optimal pronunciation training.” IRAL - International Review of Applied Linguistics, 44, pp. 357–404, 2006.

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

[6] Witt, S.M. and Young, S., “Phone-level Pronunciation Scoring and Assessment for Interactive Language Learning”, *Speech Communication,* vol. 30, pp. 95-108, 2000.

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

[8] Oostdijk, N., “The design of the spoken Dutch corpus.” in *New Frontiers of Corpus Research*, Peters, P., Collins, P. and Smith, A. [Eds.], Rodopi, Amsterdam, pp. 105–112, 2002.

[9] Strik, H., Truong, K., de Wet, F. and Cucchiarini, C., “Comparing different approaches for automatic pronunciation error detection”, to appear in *Speech Communication*.

[10] Neri, A., Cucchiarini, C., Strik, H., “The effectiveness of computer-based corrective feedback for improving segmental quality in L2-Dutch”, ReCALL, Volume 20, Issue 02, May 2008, pp. 225-243.