How Many Speakers, How Many Texts – The Automatic Assessment of Non-Native Prosody*

Florian Hönig[1], *Anton Batliner*[1,2], *Elmar Nöth*[1]

[1]Pattern Recognition Lab, FAU Erlangen-Nuremberg, Germany
[2]Machine Intelligence & Signal Processing Group, TUM, Munich, Germany
florian.hoenig@fau.de

Abstract

We present an in-depth analysis of a method for automatically scoring the prosody of non-native speech. For studying its suitability for different application scenarios, we perform a systematic comparison of different evaluation schemes such as text (in-)dependence and/or speaker (in-)dependence. The focus lies on methodological issues, with the aim of promoting the careful evaluation of automatic assessment methods. Further contributions are the analysis of (1) a method that utilizes speaker IDs to improve performance, and (2) the analysis of performance as a function of the number of speakers and texts used for training the system.

Index Terms: non-native prosody, speech melody, rhythm, cross-validation, text-independent evaluation, text-dependent evaluation, speaker-independent evaluation, user modelling, user adaptation, oracle

1. Introduction

Non-native traits in speech present several limits for communication: Intelligibility can suffer and often listening effort increases. Further, the listener may jump to conclusions about the social skills, intellectual capability, and credibility of the talker [1, 2].

Although segmental traits are typically in the focus of attention, supra-segmental traits play an important role, too. The appropriate rhythm helps the human listener decode the stream of sounds into words; word accents ease word recognition and can carry lexical information; phrase accents and prosodic boundaries help uncovering the syntactic structure of spoken language. Beyond that, prosody carries semantic and pragmatic information, such as intentions, attitudes, or emotional and physical conditions [3, 4]. For non-native speakers, transfer of L1 supra-segmental patterns, but also segmental patterns such as missing centralization can lead to poor prosody, potentially corrupting all of its functions. Thus, prosody is an important part of second language learning [5].

There is certainly an interest in automatically assessing the quality of non-native speech with respect to prosody. Main applications are computer-assisted pronunciation training (CAPT) and computerized language proficiency tests. Automatic assessment of prosody could potentially also be useful for advancing the performance of ASR systems on non-native speech (e. g. automatically switching between acoustic models). One can discriminate between the detection of concrete errors such as word accent [6] and the assessment of the general appropriateness of the prosody [7, 8, 9]. In this paper, we will deal with the latter: the overall, especially rhythmic and melodic, quality of prosody.

Specifically, we compare the performance of our approach for automatic pronunciation assessment in different application scenarios: text-independent vs. the application to a limited set of known texts; absence vs. presence of knowledge about the speaker. While doing so, we make an effort to exemplify a methodologically sound evaluation. This seems to be important because it is not uncommon for studies to lack a rigorous evaluation or at least a precise description of the evaluation procedure. For example, while person-independent evaluation is commonplace (in the speech community), text-independent evaluation can be missing even for alleged text-independent systems. We demonstrate how cross-validation can be applied to make best use of limited data and at the same time comply with speaker- and/or text-independence constraints. Lastly, we study how the number of speakers and texts collected influences performance.

2. Data

We employ data from the AUWL corpus [10]. Here, learners of English as a second language practised pre-scripted dialogues. We created 18 dialogues on topics such as business negotiations, shopping, or holidays. For later automatic processing, we annotated a likely distribution of primary and secondary phrase accents and B2/B3 boundaries [11] of a prototypical, clear realisation.

For the virtual dialogue partner, recordings of native reference speakers were used. The learners had the opportunity to first familiarize themselves with each dialogue. When enacting the dialogue, the learner could either read his or her lines off the screen (karaoke), have them prompted by a reference speaker and repeat afterwards, or speak the lines together with a reference speaker (shadowing). For the less advanced learners, there was an option to break down longer lines into subphrases. Through these measures, we obtained material that is more natural and contains less reading-related hesitations than read non-native speech.

In order to simplify the experiments, we annotated whether the spoken words deviate from the target sequence, and exclude those cases from the data. In an application, a speech recognizer could be used for this tasks; also, at least in CAPT we can assume a cooperative user. The non-native material amounts to approx. 5.5 hours of speech, comprising 3732 tokens (items, recordings) and 412 distinct types (different texts)

* The research leading to these results has received funding from the German Federal Ministry of Education, Science, Research and Technology (BMBF) under grant 01IS07014B (C-AuDiT), and the German Ministry of Economics (BMWi) under grant KF2027104ED0 (AUWL). The responsibility lies with the authors.

from 31 speakers (age 36.5 ± 15.3 years; 13 female, 18 male; native languages: 2 Arabic, 1 Brazilian Portuguese, 3 Chinese, 1 French, 16 German, 1 Hungarian, 4 Italian, 3 Japanese). The native reference utterances were spoken by three female and three male speakers in both normal and slow tempo (1908 items, 159 tokens, 2.2 hours).

We had five phoneticians annotate each of the non-native recordings with respect to intelligibility and (general) non-native accent, and specifically with respect to its prosody, answering the following question:

> THE ENGLISH LANGUAGE HAS A CHARACTERISTIC PROSODY (SENTENCE MELODY AND RHYTHM, I. E. TIMING OF THE SYLLABLES). THIS SENTENCE'S PROSODY SOUNDS ...
>
> (1) *normal,*
> (2) *acceptable, but not perfectly normal,*
> (3) *slightly unusual,*
> (4) *unusual, or*
> (5) *very unusual.*

With the (simplifying) assumption of an interval scale, we took the arithmetic average of the five labellers to obtain reliable prosody scores [12, 13], with an average of 1.7 and a standard deviation of 0.53. Intra-speaker standard deviation is 0.35, inter-speaker standard deviation is 0.40.

3. Modelling

We compute a prosodic 'fingerprint' of each recording, a fixed-length feature vector that is later fed into a regression system. The features are described in detail in [9, 10]; here, we only give a short overview. All processing is fully automatic; however, we assume that the spoken word sequence is identical with the target sequence according to the current dialogue step. Thus, segmentation can be performed accurately with the help of a speech recognition system. The features make use of the pronunciation dictionary, which contains also syllable boundaries and word accents. The prototypical distribution of phrase accents and prosodic boundaries is utilized for inferring the probable accentedness of mono-syllabic words.

3.1. Specialized Rhythm Features

There is a body of research on modelling language-specific (native) rhythm. These hand-crafted, specialized parameters are promising candidates for our task. We use features modelling duration, possible isochrony properties [14], pair-wise duration variability indices [15, 16], and proportions of interval durations [17, 18], in total 19 features.

3.2. General-Purpose Prosodic Features

The expert-driven, specialized rhythm features described above are all based on duration, so they might miss other relevant information present in the speech data, such as pitch or loudness. Therefore, we tried to capture as much potentially relevant prosodic information of a recording as possible in an approach somewhere between knowledge-based and brute-force. We are aware that this exhaustiveness comes at the cost of some redundancy in the feature set, and also high dimensionality, so we leave it to data-driven methods to find out the relevant features and the optimal weighting of them.

We first apply our comprehensive general-purpose prosody module [19] which has proven suitable for various tasks such as phrase accent and phrase boundary recognition [19] or emotion recognition [20]. The features are based on duration, energy, pitch, and pauses. Short-time energy and fundamental frequency are computed on a frame-by-frame basis, suitably interpolated, normalized per recording, and perceptually transformed. The module provides 11 global features, which we use, but more importantly, the module can be applied to locally describe arbitrary units of speech such as words or syllables. Their contour over the unit of analysis is represented by a handful of functionals such as maximum or slope. To account for intrinsic variation, we include normalized versions of some of the features based on energy and duration, e. g. the normalized duration of a syllable based on the average duration of the respective phonemes and a local estimate of the speech rate. The statistics necessary for these normalization measures are estimated on the native reference utterances in case of text-dependent evaluation; when evaluating text-independent performance, we use the C-AuDiT database [6] which contains different text material (11 native speakers amounting to five hours). We apply the module to different local units and construct fixed-length, global features features from that:

- We apply the prosody module to all *stressed syllables* ± 2 neighbours (105 features). Global features are derived by calculating mean and standard deviation. The same is done for just the nuclei of stressed syllables, yielding $105 \cdot 2 \cdot 2 = 420$ features. These features can be interpreted to generically capture isochrony properties inspired by [14].
- We apply the prosody module to all words (without further context; 35 features), and again use mean and standard deviation to obtain global features. The same is done for syllables and nuclei ($3 \cdot 2 = 210$ features). These features can be interpreted as generalizations of the deltas and proportions proposed by [17, 18].
- Further global features are computed from all words, syllables, and nuclei by calculating the average pairwise difference between the features from neighbouring units ($3 \cdot 35 = 105$ features) These features can be interpreted to generalize the pairwise variability indices proposed by [15, 16].

3.3. Regression

In total, each recording is now represented by a 761-dimensional feature vector. We apply Support Vector Regression (SVR) [21] with a radial basis kernel $\mathrm{K}(\mathbf{x}, \mathbf{y}) = \exp(-\gamma||\mathbf{x} - \mathbf{y}||^2)$ to predict the prosody score from that feature vector. SVR has a regularization meta-parameter C which controls complexity: the higher C, the higher the complexity (and thus discrimination power, but also likelihood of overfitting). The other meta-parameter γ controls the properties of the non-linear feature space transform: higher γ = smaller radius of kernel = influence of support vectors is more local = less smooth transform $\approx$ higher complexity. Due to its regularization, SVR is sensitive to the scaling of the individual features, the more so with the non-linear kernel. Therefore, we first normalize each feature individually to a standard deviation of one. To ease the later optimization of the meta-parameters C and γ, we then apply a global scaling to normalize the length of the feature vectors to an average of one. The normalization factors are estimated on the training data.

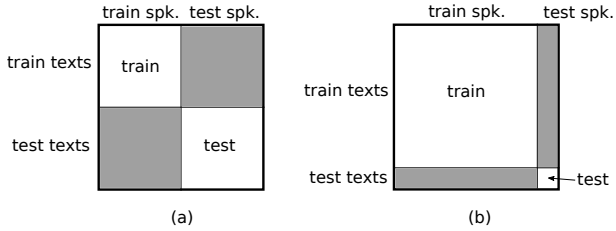

Figure 1: (a) Partitioning of a database into speaker- and text-independent training and test set of equal size by halving the set of speakers and halving the set of texts. The resulting sets comprise $1/4$ of the data each; half of the data cannot be used (grey). (b) One out of $9 \cdot 9 = 81$ single iterations in a nested 9-fold speaker independent and 9-fold sentence independent cross-validation. The training set comprises $64/81 \approx 79.0\%$ of the data, the test set $1/81 \approx 1.2\%$; the remaining 19.8% cannot be used (grey). Through the course of all iterations, all data is tested exactly once.

4. Experiments and Results

The performance of the regression system must be estimated on unseen test data. When we want to evaluate speaker-independent performance, this test data must not contain speakers used to train the model; similarly, for text-independent performance, the test data must not contain texts used in training. When evaluating speaker- *and* text-independently, both conditions have to be met at the same time, which limits the amount of data that can be used. For example, when training with half of the speakers, and half of the texts, and testing with the other half of the speakers, and the other half of the texts, the training set will contain[1] only $(1/2)^2 = 1/4$ of the data; just the same, the test set will contain only $1/4$ of the data. Half of the originally available data cannot be used at all, cf. Figure 1 (a).

4.1. Cross-validation

In order to make better use of the limited data, we resort to cross-validation, i. e. the database is split up into N folds and we loop over these as the test data while the respective other $N - 1$ folds serve as the training data. Care has to be taken when evaluating speaker-independent performance: the folds have to be disjunct with respect to speakers; similar constraints hold for text-independent evaluation. When evaluating both speaker- and text-independently, a *double nested* loop over speaker and text folds is necessary. With N speaker folds and M text folds, $N \cdot M$ total iterations result. The fraction of the data that can be used for training in each iteration is $(M - 1) \cdot (N - 1)/(N \cdot M)$, and $1/(N \cdot M)$ for testing. The simplest case of using $N = M = 2$ can be illustrated with Figure 1 (a): In each of the $2 \cdot 2 = 4$ folds, one of the sub-squares serves a training; the diagonally opposite sub-square serves as test. Now, all data is exploited for testing, but the problem of the small training set in each iteration remains. This can be alleviated by increasing the number of folds, N and M. We intend to compare different evaluation schemes, so we aim for equally large training sets in all schemes. This is achieved in the following way: We choose $N = 9$ speaker folds and $M = 9$ text folds for the double nested case, resulting in $9^2 = 81$ total folds with $(8/9)^2 \approx 79.0\%$ available for training, cf. Figure 1 (b). For the single nested case (evaluating just text- *or* speaker-independently, or neither), we choose $N = 5$ folds, resulting in 5 iterations with $4/5 = 80.0\%$ available for training. Thus, results for the different evaluation set-ups are comparable with respect to the number of items used for model fitting.

[1] unless the database has been designed in a specifically stratified way [10]

4.2. Optimization of Meta-Parameters

C and γ have to be chosen suitably to get decent performance on unseen data. They cannot be optimized on the same data used to fit the SVR model, as this would lead to overfitting, i. e. poor generalization. What is more, the data used for optimisation should reflect the mode of evaluation, i. e. speaker- and/or text-independence where applicable. Strictly speaking, we would have to optimize on a separate validation set; however, this would either reduce the data available for training, or incur further nested loops in the cross-validation. For example, in the speaker- and text-independent case, we would need a quadruple nested cross-validation with 18 speaker and 18 text folds and $18^4 = 104\,976$ iterations to reach our intended training size of $79.6\% \approx (17/18)^4$. For simplicity, we refrain from doing this, and instead optimize for the best overall result of the cross-validation (hill climbing in powers of ten starting from $C = \gamma = 1$). Thus, we effectively optimize the meta-parameters on test. This leads to slightly optimistic results, but the effect is small since we are only optimizing two parameters, and only very coarsely.

4.3. Evaluation

We report (and optimize the meta-parameters for) Spearman's correlation coefficient ρ between the target labels and SVR predictions on test. We use Spearman because it is more 'conservative' and robust than Pearson's correlation coefficient. Rather than computing a correlation coefficient for each cross-validation iteration, we compute a single coefficient for the combined test data (through the course of all iterations, all data is tested exactly once).

The pronunciation quality of the items within a single speaker can be expected not to vary too much. Indeed, in our data, intra-speaker variance of the scores is even smaller than inter-speaker variance, cf. Section 2. For a CAPT application, it is interesting to see how well the system recognizes these subtler differences within a single speaker. We therefore also report (and optimize the meta-parameters for) the average correlation within speakers, denoted $\bar{\rho}$.

The fact of relatively constant scores within a speaker can be exploited to increase ρ. Let y_i denote the predicted score of item i, and $\bar{y}_{s(i)}$ the averaged predictions of the speaker $s(i)$. With a suitable weight w, an improved prediction is given by $y'_i = (1 - w) \cdot y_i + w \cdot \bar{y}_{s(i)}$, i. e. we pull the less reliable per-item predictions towards each speaker's supposed mean which is more reliable. Note that the method is also applicable if speaker IDs are not provided for test, as they can be estimated with speaker diarisation techniques. This is not a method one would consider for a CAPT application, as it tends to cement the scores the learner gets in spite of his or her efforts. In line with this, our measure for intra-speaker performance $\bar{\rho}$ is invariant against it, as long as $w < 1$. (This is because $\bar{y}_{s(i)}$ is constant per speaker, and thus doesn't affect the intra-speaker correlation. If $w = 1$, y'_i is constant per speaker, resulting in an undefined intra-speaker correlation.) Nevertheless, the method can be used to improve results in official evaluations such as the INTERSPEECH paralinguistic challenges [23, 24], so it is instructive to see how far one can get with it. (Moreover, this

Table 1: *Results for different cross-evaluation set-ups: Testing on speakers/texts unseen in training ('independent') or speakers/texts included in train ('dependent'). 'ID' refers to the explicit provision of speaker/text IDs. Meta-parameters C and γ once optimized for overall correlation ρ (rows in normal typeface), and once optimized for average intra-speaker correlation $\bar{\rho}$ (rows in italics). w is the optimal weight for pulling predictions towards speaker means, resulting in ρ^*. ρ_s is the correlation for speaker means. Further explanations in Section 4.3.*

Speaker	Text	C	γ	ρ	$\bar{\rho}$	w	ρ^*	ρ_s
independent	independent	1	1	0.571	0.350	0.7	0.679	0.910
		1	*0.01*	*0.516*	*0.409*	*0.7*	*0.598*	*0.782*
independent	dependent	1	1	0.614	0.417	0.6	0.693	0.901
		1	*0.1*	*0.585*	*0.439*	*0.6*	*0.662*	*0.876*
independent	dependent + ID	10	0.1	0.603	0.461	0.7	0.684	0.865
		1	*0.1*	*0.597*	*0.475*	*0.6*	*0.668*	*0.850*
dependent	independent	1	1	0.648	0.368	0.6	0.726	0.954
		1	*0.01*	*0.608*	*0.419*	*0.7*	*0.693*	*0.874*
dependent	dependent	*1*	*1*	0.712	0.434	0.5	0.759	0.958
		1	*0.1*	*0.686*	*0.450*	*0.5*	*0.742*	*0.931*
dependent	dependent + ID	1	1	0.701	0.451	0.5	0.752	0.935
		1	*0.1*	*0.688*	*0.479*	*0.5*	*0.746*	*0.919*
dependent + ID	independent	1	0.1	0.725	0.421	0.4	0.749	0.990
		1	*0.1*	*0.725*	*0.421*	*0.4*	*0.749*	*0.990*
dependent + ID	dependent	1	1	0.762	0.441	0.3	0.774	0.990
		1	*0.1*	*0.756*	*0.469*	*0.3*	*0.770*	*0.990*
dependent + ID	dependent + ID	10	0.1	0.766	0.483	0.3	0.776	0.996
		1	*0.1*	*0.755*	*0.499*	*0.4*	*0.776*	*0.982*

method might be useful when not monitoring the development of speakers over time but assessing different speaker groups only once.) As an upper bound, we report the result with the best weight, denoted ρ^*, and use the actual speaker IDs.

For language proficiency tests, it is interesting to see how well the average score of a speaker is predicted. Therefore, we also report the Spearman correlation when averaging reference and predicted scores over all items of a speaker (3732 items / 31 speakers $\approx$ 120 on average), denoted ρ_s.

For both CAPT and language proficiency tests, one can imagine scenarios where the model is applied to known texts. We estimate performance under this setting by executing a normal cross-validation without the text independence constraint, i. e. just selecting the folds randomly from all items. Thus, nearly all sentences of test are also contained in train in each iteration. In this 'known text' scenario, one could even go further and train an individual model for each sentence. However, in our database we have not enough samples per text for that (3732 items / 412 texts $\approx$ 9.1 on average). What we can still do is to take the text-dependent evaluation, and additionally provide a 'text oracle' – giving the text ID explicitly to the model. We do this by appending a one-hot-encoding of the ID to the features, i. e. a 412-dimensional vector with one at the index of the text ID and zero elsewhere.

In CAPT it is conceivable to improve performance with some kind of user adaptation, either in an unsupervised way, or there may be configurations where the user ID is known to the system. Without delving into actual adaptation methods, we measure performance when the system is applied to known speakers. This should give an upper bound of the performance that one may reach with speaker adaptation techniques. Similarly to the text dependent evaluation, we estimate performance on known speakers by executing a normal cross-validation without the speaker independence constraint for train/test, i. e. just select the folds randomly from all items. Thus, nearly all speakers of test are also contained in train in each iteration. Again, one step further is the 'speaker oracle' – adding a one-hot-encoding of the speaker ID to the features.

4.4. Results

Table 1 gives the measured performance for the different evaluation schemes. In a user- and text-independent setting (cf. Speaker='independent' and Text='independent'), predictions are correlated with the target scores with $\rho = 0.571$ (when optimizing for ρ, row with normal typeface). Intra-speaker correlation is much lower: $\bar{\rho} = 0.409$ (when optimizing for $\bar{\rho}$, row with italic typeface). The lower performance can be explained by the fact that this task is principally harder – consider that intra-speaker standard deviation is only 0.35 while total standard deviation is 0.53, cf. Section 2. Going back to the overall performance (normal typeface), we see that the trick of pulling predictions toward the speaker means improves results strongly: $\rho^* = 0.679$. The average speaker performance is estimated with $\rho_s = 0.910$.

When the texts are known to the system (cf. Speaker='independent' and Text='dependent'), results improve a little: overall correlation ρ from 0.571 to 0.614, and intra-speaker $\bar{\rho}$ from 0.409 to 0.439. Note that the text dependent experiments differ also slighly in the features – normalization statistics are now estimated on the same texts. The improvement due to this, however, is small (from $\rho = 0.610$ to $\rho = 0.614$, not contained in Table 1). Adding explicit text IDs (Speaker='independent' and Text='dependent + ID') was not successful for improving overall correlation: ρ drops slightly from 0.614 to 0.603. For intra-speaker correlation, however, it gave a relatively large gain from $\bar{\rho} = 0.439$ to 0.475.

When simulating user adaptation by speaker-dependent evaluation, results also improve. We first look at the version with text independence (cf. Speaker='dependent' and Text='independent') and compare it with the initial configuration (both speaker- and text-independence). Overall correlation ρ improves considerably from 0.571 to 0.648, and also aver-

age speaker performance is estimated more precisely (ρ_s rises from 0.910 to 0.958), but intra-speaker correlation $\bar{\rho}$ improves only slightly from 0.409 to 0.419. Apparently, the system learns to recognize the average speaker performance quite well, but this does not help much for within-speaker performance. When adding explicit speaker IDs (cf. Speaker='dependent + ID' and Text='independent'), overall correlation ρ improves further from 0.648 to 0.725, but again, within-speaker performance improves only slightly ($\bar{\rho}$: from 0.419 to 0.421). It should be mentioned that the high performance for the speaker average, $\rho_s = 0.990$, should not be taken literally: the explicit speaker IDs allow the system system to 'memorize' the average performance of a speaker without even considering the prosodic features, so the results for ρ_s are moot in the three cases with given speaker IDs.

The improvements seen for adding implicit or explicit knowledge about test speakers or texts are largely additive. For example, when evaluating both speaker- and text-dependently (cf. Speaker='dependent' and Text='dependent'), overall correlation ρ improves from 0.571 to 0.712, an absolute difference of 0.141 (the individual improvements were $0.614 - 0.571 = 0.043$ for text dependence and $0.648 - 0.571 = 0.077$, together 0.12). Similarly, intra-speaker correlation $\bar{\rho}$ improves from 0.409 to 0.450, a difference of 0.041 (individual improvements were $0.439 - 0.409 = 0.030$ for text dependence and $0.419 - 0.409 = 0.010$ for speaker dependence, together 0.04). For the evaluation with maximal prior knowledge, i. e. speaker and text dependence plus speaker IDs and text IDs, we reach an overall correlation ρ of 0.766 and an intra-speaker correlation ρ_s of 0.499.

We now have a detailed look at the most relevant setting, the text- and speaker-independent evaluation. Here, we analyse how results change when varying the number of texts and the number of speakers used in training, while keeping the number of items constant. The results are shown in Figure 2. Thinning the training data completely randomly (curve 'Items') has the least negative effect. Even when using only the 32th part of items, i. e. 92 trainings items instead of 2949, performance 'only' drops from 0.571 to 0.391. Thinning out with respect to the number of texts (curve 'Texts') has a worse effect: here, using only 12 instead of 348 texts (but still 92 items) leads to a degradation to 0.313. Training with fewer speakers has the greatest impact: as the curve 'Speakers' shows, fewer speakers lead – at the same number of items – to a much quicker breakdown in performance than fewer texts or fewer randomly selected items. In the extreme case of using only the 32th part of items, resulting in 1.7 speakers on average (but still 92 items), performance is down to ρ =0.101.

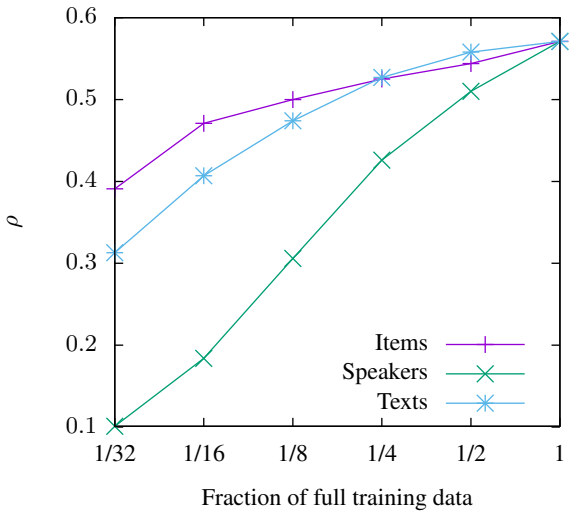

Figure 2: Speaker- and text-independent performance as a function of the amount of training data: In each of the 81 cross-validation folds, only the indicated fraction of the respective training items is used. Selection is done randomly, either across all *items*, or stratified by *speakers*, or by *texts*. For all stratifications, the full training data ('1') comprises $64/81 \cdot 3732 \approx 2949$ items, 1/2 $\hat{=}$ 1474 items, ..., 1/32 $\hat{=}$ 92 items (on average). For stratification by speaker, 1 $\hat{=}$ 28 speakers, 1/2 $\hat{=}$ 14 speakers, ..., 1/32 $\hat{=}$ 1.7 speakers. For stratification by text, 1$\hat{=}$ 348 texts, $1/2\hat{=}$ 174 texts, ..., $1/32\hat{=}$ 12 texts.

5. Conclusion

By systematically comparing different evaluation procedures – text dependence/independence, speaker dependence/independence – we were able to quantify which performance can be expected in different application scenarios: How much can be gained by limiting a CAPT training session to known texts; how much could possibly be gained by suitable speaker adaptation techniques? Further, the performance difference between dependent and independent evaluation schemes highlights the importance of careful evaluation in order not to overestimate performance. For example, when evaluating a system aimed for assessing unknown speakers pronouncing new texts, the difference between correct evaluation (speaker- and text-independent) and applying a conventional cross-validation as offered by standard machine learning packages such as WEKA [25] which mixes speakers and texts (and thus evaluates speaker- and text-dependently) is as large as $\rho = 0.571$ vs. 0.712. Further, we quantified the improvement that can be gained by pulling the predictions towards the speakers' mean prediction – a method not particularly meaningful in a CAPT context, but nevertheless possible and promising, e.g., within official evaluations or across 'static' speaker groups.

Finally, we analysed how performance for the most important use case (speaker- and text-independence) depends on the number of items, speakers and texts used for training: Collecting different texts rather than having speakers pronounce the same material is beneficial; more important is however the collection of as many speakers as possible (rather than having a lot of material from few speakers). For that particular constellation, one could conclude from Figure 2 that in terms of different texts, some kind of saturation is being reached when using all available material (348 text types), so it seems that taking more than 500 different texts will not be the key to a pronounced further improvement. Regarding the number of speakers, the available material (28 speakers) seems far from saturation, so we can expect considerable improvement from collecting 50 or more speakers. As this constellation aims for text-independent performance, having each speaker produce different material should be most effective.

For the question formulated in the title of how many speakers and texts to collect, we cannot give an universal answer: what level of correlation it takes to build an acceptable system has to be evaluated in user studies. Intuitively, the best correlations obtained in the speaker-independent evaluations ($\rho = 0.614$, $\bar{\rho} = 0.457$) leave still room for improvement.

6. References

[1] A. Gluszek and J. F. Dovidio, "The way they speak: A social psychological perspective on the stigma of non-native accents in communication," *Personality and Social Psychology Review*, vol. 14, no. 2, pp. 214–237, 2010.

[2] S. Lev-Ari and B. Keysara, "Why don't we believe non-native speakers? the influence of accent on credibility," *Journal of Experimental Social Psychology*, vol. 46, no. 6, pp. 1093–1096, 2010.

[3] H. Fujisaki, "Foreword," in *Proceedings of the International Symposium on Prosody*, Yokohama, Japan, 1994.

[4] B. Schuller and A. Batliner, *Computational Paralinguistics – Emotion, Affect, and Personality in Speech and Language Processing*. Chichester, UK: Wiley, 2014.

[5] R. M. Dauer, "The lingua franca core: A new model for pronunciation instruction?" *TESOL Quarterly*, vol. 39, pp. 543–550, 2011.

[6] F. Hönig, A. Batliner, K. Weilhammer, and E. Nöth, "Islands of failure: Employing word accent information for pronunciation quality assessment of english L2 learners," in *Proceedings of SLATE*, Wroxall Abbey, 2009.

[7] C. Teixeira, H. Franco, E. Shriberg, K. Precoda, and K. Somnez, "Prosodic features for automatic text-independent evaluation of degree of nativeness for language learners," in *Proceedings of ICSLP*, Beijing, 2000.

[8] J. Tepperman, T. Stanley, K. Hacioglu, and B. Pellom, "Testing suprasegmental english through parroting," in *Proceedings of Speech Prosody*, Chicago IL, USA, 2010.

[9] F. Hönig, A. Batliner, K. Weilhammer, and E. Nöth, "Automatic assessment of non-native prosody for english as L2," in *Proceedings of Speech Prosody*, Chicago IL, USA, 2010.

[10] F. Hönig, A. Batliner, and E. Nöth, "Automatic assessment of non-native prosody – annotation, modelling and evaluation," in *Proceedings of the International Symposium on Automatic Detection of Errors in Pronunciation Training (IS ADEPT)*, Stockholm, 2012, pp. 21–30.

[11] A. Batliner, R. Kompe, A. Kießling, M. Mast, H. Niemann, and E. Nöth, "M = Syntax + Prosody: A syntactic–prosodic labelling scheme for large spontaneous speech databases," *Speech Communication*, vol. 25, pp. 193–222, 1998.

[12] F. Hönig, A. Batliner, K. Weilhammer, and E. Nöth, "How many labellers? Modelling inter-labeller agreement and system performance for the automatic assessment of non-native prosody," in *Proceedings of SLaTE*, Tokyo, Japan, 2010.

[13] F. Hönig, A. Batliner, and E. Nöth, "How many labellers revisited – naives, experts and real experts," in *Proceedings of SLaTE*, Venice, Italy, 2011, pp. 137–140.

[14] D. Abercrombie, *Elements of General Phonetics*. Edinburgh: University Press, 1967.

[15] E. Grabe and E. L. Low, "Durational variability in speech and the rhythm class hypothesis," in *Laboratory Phonology VII*, C. Gussenhoven and N. Warner, Eds. Berlin: de Gruyter, 2002, pp. 515–546.

[16] P. M. Bertinetto and C. Bertini, "On modeling the rhythm of natural languages," in *Proceedings of Speech Prosody*, Campinas, Brazil, 2008.

[17] F. Ramus, "Acoustic correlates of linguistic rhythm: Perspectives," in *Proceedings of Speech Prosody*, Aix-en-Provence, 2002, pp. 115–120.

[18] V. Dellwo, "Influences of speech rate on the acoustic correlates of speech rhythm. An experimental phonetic study based on acoustic and perceptual evidence," Ph.D. dissertation, Rheinische Friedrich-Wilhelms-Universität Bonn, 2010.

[19] A. Batliner, J. Buckow, H. Niemann, E. Nöth, and V. Warnke, "The Prosody Module," in *Verbmobil: Foundations of Speech-to-Speech Translations*, W. Wahlster, Ed. Springer, 2000, pp. 106–121.

[20] A. Batliner, S. Steidl, C. Hacker, E. Nöth, and H. Niemann, "Tales of tuning – prototyping for automatic classification of emotional user states," in *Proceedings of INTERSPEECH*, Lisbon, Portugal, 2005, pp. 489–492.

[21] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," *Advances in neural information processing systems*, vol. 9, pp. 155–161, 1997.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[23] B. Schuller, S. Steidl, A. Batliner, J. Epps, F. Eyben, F. Ringeval, E. Marchi, and Y. Zhang, "The INTERSPEECH 2014 computational paralinguistics challenge: Cognitive & physical load," in *Proceedings of INTERSPEECH*, Singapore, 2014, pp. 427–431.

[24] B. Schuller, S. Steidl, A. Batliner, S. Hantke, F. Hönig, J. R. Orozco-Arroyave, E. Nöth, Y. Zhang, and F. Weninger, "The INTERSPEECH 2015 computational paralinguistics challenge: Nativeness, Parkinson's & eating condition," in *Proceedings of INTERSPEECH, Dresden, Germany*, 2015, to appear.

[25] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.