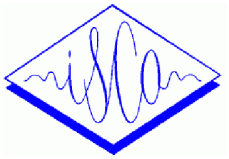

Third Workshop on Spoken Language
Technologies for Under-resourced Languages
Cape Town, South Africa
May 7-9, 2012

VALIDATING SMARTPHONE-COLLECTED SPEECH CORPORA

Marelie H. Davel, Charl J. van Heerden and Etienne Barnard

North-West University, Vanderbijlpark, South Africa

ABSTRACT

We investigate the effectiveness with which the accuracy of a prompted speech corpus can be validated when minimal additional speech resources are available, and specifically when a language model in the target language is not available. We compare a word-based variant of Goodness of Pronunciation (GOP) with a phone-based dynamic programming (PDP) scoring technique. The first technique uses the acoustic likelihood ratio and the second the optimal alignment between an observed phone string (generated by a speech recogniser) and a reference phone string (obtained from a dictionary) to generate validation scores. We define a new technique to obtain a PDP scoring matrix in a data-driven fashion, examine different ways of using GOP for word scoring, and find that variants of both techniques provide results that are effective for corpus validation.

Index Terms— speech corpora, corpus validation, goodness of pronunciation, phone-based dynamic programming scores

1. INTRODUCTION

The development of high-quality automatic speech-recognition (ASR) systems requires large and diverse corpora of orthographically transcribed speech [1]. For the vast majority of the languages on earth, such corpora are not available [2] and it does not seem feasible that normal commercial forces will produce resources for all these languages in the foreseeable future. Hence, it is significant that recent years have seen the development of several novel approaches to obtain such data at substantially lower costs than those of conventional corpora. For example:

- Audio recordings along with approximate transcripts, which are made available on the Internet for human consumption, can be mined to extract useable corpora [3, 4, 5];
- Crowdsourcing can be used to solicit recordings or transcriptions from a wide range of contributors [6]; or
- Smartphone applications can be used to collect speech from targeted speakers quickly and efficiently [7, 8]. Smartphone applications (also utilised in this study) typically assists with the enrolment of contributors, displays the prompts which are to be read and records the resulting speech.

In all of the above approaches, the accuracy of the available transcriptions (i.e. how well the transcriptions match the spoken acoustics) is an important unknown. The identification or validation of acoustic segments that match transcription segments is an important task for several reasons, including the following:

- Depending on the fraction of the corpus with inaccurate transcriptions, such selection may be crucial, mildly beneficial or irrelevant for the training of acoustic models.
- In all cases, having a gold standard of accurately transcribed speech is a prerequisite for assessing the accuracy of ASR systems developed using the data.
- The well-transcribed portions may be useful for other purposes, such as the development of pronunciation dictionaries or for the analysis of the acoustic phonetics of the target languages.

Clearly, corpus selection / validation of this nature is a multi-faceted task, and a number of factors must be considered in order to develop approaches that perform well under a given set of circumstances. A basic issue concerns the extent of the items that are to be validated, which can range from entire utterances down to words or even individual phones. Another important set of issues relates to the quality of existing resources in the target language: are resources such as acoustic models, high-accuracy pronunciation models or language models with suitable coverage available? These issues interact with the size of the corpus to be validated: a large corpus can, to some extent, compensate for limitations in the existing resources. Finally, the extent of the expected mismatch between acoustics and transcriptions is an important variable: for example, prompts read in a laboratory by reasonably literate respondents will be significantly different from those produced by infrequent readers in an outdoor setting. Approximate transcriptions of spontaneous speech (aimed, for example, at visitors to a Web page who have the option of either listening to a podcast or reading the associated transcript) will have yet another set of characteristics.

In this work we consider the validation of corpora that were collected with a smartphone-based application, *Woefzela* [8]. As is typical of under-resourced languages, both the language models and the available pronunciation lexicons are quite limited; on the other hand, our data-collection tools and protocol make it easy to collect fairly sizeable corpora (consisting of 50 to 100 hours of speech). Also, the resulting transcriptions are reasonably accurate, since the field workers doing the data collection are instructed to verify that the speakers are sufficiently competent at reading the prompts displayed by Woefzela. Given our wish to retain as much of the corpus as possible, and to perform lexicon development within this context, we have chosen to select individual *words* as the items to be verified.

Below, we briefly review some previous work that is relevant to this task. Thereafter, Section 3 describes the approach we have taken, as well as some of the variations possible. In section 4 we describe an empirical investigation undertaken for two under-resourced South African languages (isiNdebele and Afrikaans), the results of which are presented in Section 5. In Section 7 we review our findings and provide a perspective on what remains to be done.

2. BACKGROUND

The validation task that we introduced above can be schematised as follows: given a spoken utterance x, how confident are we that the supposed transcription $w_1 w_2 ... w_N$ corresponds to x? To move to a finer grain, we can perform forced alignment, and then ask the same question about each word w_i and its aligned stretch of speech x_i, or even each of the individual phones in a phonetic transcription of w_i. The utterance-level question is generally associated with confidence scoring in ASR – for example, to decide whether confirmation of a recognised user response is required. The word-level analysis typically occurs in spoken-term detection (STD), and phone-level judgments are commonly required for computer-aided pronunciation tutoring (CAPT).

In light of the statistical approaches that currently dominate ASR, it is natural to phrase each of these questions in probabilistic terms. For example, one can compute the posterior probability of the relevant entity having occurred, given the acoustic evidence. If lattice-based recognition is used for either ASR or STD, a relatively straightforward algorithm for the computation of such posteriors was developed by Wessel *et al.* [9]; if a sufficiently rich language model is available, such lattice-based confidence scoring has been shown to perform very well.

In the absence of such a representative language model, alternative confidence estimates have to be considered. Several approaches are summarised in [10]; typically, some combination of acoustic likelihoods, density of the hypothesised words in the N-best list, and prosodic features is used. Innovations in this category continue to flourish – for example, Vu *et al.* recently proposed [11] the combination of N-best lists from acoustic models for several languages in order to perform confidence scoring when a sufficiently accurate acoustic model for the target language is not available.

Different requirements, namely the need for faster search speed and smaller index size while doing spoken term detection, were the motivation for the phone-based detection techniques developed in [12]. Here a Hidden Markov Model is used to create a probabilistic pronunciation model of the expected sequence, and this is used to score the observed sequence relative to an n-gram based garbage model.

For CAPT, the need to make phone-level decisions and the relative certainty about the intended utterance are both contra-indications for a lattice-based solution. In fact, the most popular confidence measures in this application – known as goodness-of-pronunciation (GOP) measures [13] – simply compare the phone scores found during forced alignment of the intended utterance with the highest phone score during a free decode (with a phone loop) of the same portion of the audio; this can also be seen as an approximation of the posterior probability of the target phone.

In our case, the weakness of available language models excludes the use of lattice-based confidence scores; on the other hand, we have enough training data to create reasonably accurate language-specific acoustic models, suggesting that multilingual N-best lists will be an unnecessarily complicated solution. We have recently achieved good performance with an approach that computes the confidence score as the lowest dynamic-programming cost when aligning the freely decoded phone string to the forced alignment of the transcription provided [5, 14], which we describe and extend below. This approach, along with a word-based adaptation of GOP, seems like the most suitable options for our verification task, and are explored during our empirical investigation.

3. PHONE-BASED DYNAMIC PROGRAMMING (PDP) SCORES

As with GOP and some lattice-based confidence scores, we perform (a) forced alignment with the presumed transcription and (b) phone recognition with an ergodic phone loop of the same audio segment but then use the phone strings directly to obtain a distance measure. We use dynamic programming – with a variable scoring matrix – to map the one phone string to the other. The variable scoring matrix allows us to penalise more probable recognition errors less severely than differences that are more likely to be indicative of an unmatched text/audio segment. We then normalise the resulting dynamic programming score and use this as confidence score. While this measure is less grounded in the standard Bayesian theory of ASR than those that estimate a posterior probability of the presumed transcription, it is likely to be less fragile in environments where the acoustic models do not match the acoustics of the target utterances very well [14]. The phone-based

dynamic programming (PDP) score has two other potential advantages that are relevant in our case:

- *Dictionary refinement*: the phone-string distances make it simple to group together related pronunciations of words found in the free-decode process, and thus to suggest alternative dictionary entries when such pronunciations are sufficiently frequent.
- *Robustness to phone substitutions*: dialect differences often lead to systematic substitutions; by appropriate manipulation of the scoring matrix, we can detect such substitutions, yet not treat them as erroneous utterances.

Below, we describe the general process of calculating PDP scores, as well as a novel process we employ for training a scoring matrix.

3.1. Calculating PDP scores

Given an audio and text segment, the overall process in our current implementation consists of the following steps:

1. Free recognition is performed on the audio segment using a phone-loop grammar in order to produce an *observed string*.
2. A dictionary lookup (or an ASR alignment if the target phone string is a segment within a larger utterance) produces a *reference string*.
3. A standard DP algorithm with a pre-calculated scoring matrix is used to align the observed and reference string with each other.
4. The resulting score obtained from the best dynamic programming path is divided by the number of phones in the alignment (which may be longer than either of the strings individually).
5. This score is normalised by subtracting the optimal score that can be obtained for the given reference string. (More on this below.)

Additional refinements can be implemented through phonotactic rules applicable to the specific language. For example, during alignment, not scoring any repeated phones that are mapped to insertions in the aligned string. (For example, ignoring the 'n' of the phone string /i n n/ if mapped to the aligned string /i n -/ when scoring the word 'in'.) In addition, variant

In this work, we are interested in obtaining the PDP score at the word level, where the words themselves are embedded within an utterance. Such word-level scores can be generated in different ways. In prior work [15], we used ASR-based alignments of the audio data to identify word boundaries, and used these start and end times to select the relevant phonemes from the decoded string. The decoded string was then scored against all possible variants obtained from a pronunciation dictionary, and the best score selected. A customisable time-based margin was used to decide whether a phone on the boundary of an event should be included in the event or not.

Given that such a time-based approach is sensitive to miss-alignments introduced by mismatching text and audio (such as a word omitted, added or mispronounced by the speaker, or background speech included in the recording), we utilise an alternative approach in this study. We perform ASR-based alignment of the entire utterance to select a single phone string to be used as reference, and then perform DP alignment of the reference to the observed string at the utterance level. For each word, we then use the word start and end phones from the reference string to identify corresponding words in the observed string. Inter-word phones are flagged as inserted words.

The main process to calculate PDP scores, as proposed above, is quite straightforward, but some implementation choices affect the performance of this measure. These include:

- The *granularity of the phone set*: directly related to the number of phones in the set, different choices may result in more reliable, less exact models or more detailed models that may potentially be poorly estimated.
- The *scoring matrix* used during dynamic programming. Such a matrix specifies the cost associated with a specific substitution between a phone in the reference string and the observed string, and can either be constructed manually or derived from the data.
- The precise *dynamic programming algorithm* used, and specifically, the scoring of inserted and deleted elements. For example, the Needleman-Wunsch algorithm specifies a fixed gap penalty – the cost of allowing a gap in either the observed or reference string – which may also be manipulated separately as the cost of opening a gap, and the cost of extending a gap. Our implementation associates a different value with the insertion or deletion of any given phone (based on the identity of the phone).

3.2. Training a variable scoring matrix

As it is expected that the quality of the variable scoring matrix would be central to the performance of this technique, we define a data-driven process for obtaining the scoring matrix. For this discussion, we use $score(r, o)$ to refer to the value of the scoring matrix associated with reference phone r and observed phone o. We use $score(r, -)$ to refer to the score associated with the deletion of r, and $score(-, o)$ to refer to the score associated with the insertion of o. We initialise our scoring matrix using a flat matrix, for example, a score of +1

when $r = o$ and a score of -1 of $r \neq o$. (The training process is not sensitive with regard to the initial values.)

We then align all our data at the utterance level, and calculate a counts matrix containing the number of times that each true phone r was recognised as phone o. We smooth this matrix by adding a single count to each entry. From this smoothed matrix, a set of log posterior probabilities can be calculated directly, that is, a matrix containing $logP(r|o)$: the log probability that that the true phone that occurred was r, given that o was observed. (Note that a probability-based confusion matrix usually contains $P(o|r)$, that is, the probability that a phone o would be observed given that the true phone that occurred was r.) We then add the overall deletion probability to each of the deletion scores ($score(r, -)$). We use log likelihoods to allow us to sum values across intra-word phones, and add the overall deletion log probability to compensate for the fact that the likelihood of a deletion is unknown even after seeing the observed phone string.

The final matrix is used to calculate a set of optimal log likelihoods per reference phoneme. (The maximum value of $score(r, .)$ over all elements in row r.) This is used to calculate a 'optimal score' for each reference string, which is used in (5) described above. This score acts as a 'background' model to ensure that scores across words containing either more or less confusable phonemes remain comparable.

3.3. Discussion

Apart from a different form of normalisation and calculation of the scoring matrix, the most significant difference between this method and the one proposed by [16] is the way in which the reference and observed phone strings are selected and aligned. As described in (3) above, we search for the optimal alignment, while [16] uses a simpler chunk-based system (which is appropriate in their case as their aim is to rank hypotheses, not calculate a confidence measure across terms). Interestingly, the Viterbi search for a single best path through an HMM, as used in [12], produces very similar results to performing dynamic programming alignment using an iteratively improved scoring matrix, with the same $P(o|r)$ used to calculate word-based scores in both approaches.

In the next section, we apply this technique to the validation task described in Section 1, and compare its performance to extensions of the GOP algorithm that were developed for whole-word verification.

4. EXPERIMENTAL METHOD

Our experiments were performed in the context of a real validation task, namely the development of wideband ASR corpora in the eleven official languages of South Africa, as commissioned by the South African National Centre for Human Language Technologies (NCHLT) [8]. In each language, data were collected from approximately two hundred speakers; each speaker read approximately 500 short prompts which were displayed and recorded with the Android smartphone application Woefzela. To simulate typical use cases, all languages (besides English) contained a mixture of prompts from the target language, English, and diverse proper names; more details are available in [8].

We report on experiments that were conducted on two of the NCHLT languages, namely isiNdebele and Afrikaans.

4.1. Extracting phone strings

The data in each language were split into four partitions with no speaker overlap. Each partition thus contained approximately 12 – 15 hours of speech, from approximately 50 speakers. These partitions were then used in cross validation: we repeatedly trained acoustic models on three of the partitions; corpus validation and performance evaluation were then performed on the remaining portion.

A standard 3-state left to right Hidden Markov Model (HMM) architecture was used to model context-dependent triphones in each language. As acoustic features, 39-dimensional Mel Frequency Cepstral Coefficients were used: 13 static coefficients with cepstral mean normalisation applied, 13 delta and 13 double delta coefficients. Triphones were tied at the state level using decision tree clustering, and each tied-state triphone was estimated with 8 Gaussian mixtures per state. Semi-tied transforms were employed throughout.

The pronunciation dictionaries used for acoustic model development (and for generating reference phone strings) were quite different for the two languages: both were developed using a combination of manual and semi-automatic means, but the Afrikaans pronunciation dictionary [17] was fairly comprehensive at approximately 24k words, while the isiNdebele dictionary [18] contained only about 5k words. Both contained mainly generic words (almost no proper names or words from foreign origin) and a large number of words had to be generated using letter-to-sound rules [19] extracted from the same dictionaries.

4.2. Evaluation protocol

For the evaluation process, we manually validated 50 utterances from each of 8 speakers in each language. Each utterance and all words in each utterance were judged as 'acceptable' renditions of the target transcription or not, where acceptability depended both on the acoustic quality and the pronunciation of each word or utterance. Acceptable matches were also either marked as either an *exact* match, or a *close* match, where the latter may for example include a reader rendering 'speakers speak' where 'speaker speak' was prompted. We also annotated insertions (e.g. partial repetitions which often occur if speakers struggle to read a word or phrase, or background noise / speech) and *strange* words in the utterances – these included all words that were not considered

generic words in the target language, such as foreign words, abbreviations, acronyms and proper names.

Depending on the purpose of the corpus, different levels of validation may be required. We therefore evaluate the accuracy with which our scoring system can identify problem words for different classification tasks, as listed in Table 1. The *strict* scoring strategy only accepts perfectly matched words and requires that all else be rejected; the *lenient* scoring strategy is impartial with regard to all words that are not clearly good or bad; and the *harvest* strategy is the typical one required when building a corpus for ASR purposes: both exact and close matches are acceptable and 'strange' words (when plausibly produced) are also included; only poor quality audio and badly pronounced or deleted words are rejected.

Table 1. *Different scoring strategies used during evaluation.*

strategy	accept	reject	ignore
strict	exact match strange word	bad audio wrong word deleted word close match	
lenient	exact match	bad audio wrong word deleted word	close match strange word
harvest	exact match close match strange word	bad audio wrong word deleted word	

Corpus validation is performed by accepting all words that exceed a selected threshold and determining to what extent this automated accept/reject decision matches the manually generated accept/reject decision. As we can trade off between sensitivity and specificity by adjusting the accept/reject threshold, we evaluate the effectiveness of our scoring technique for each threshold setting, and use a Detection-Error Trade-off (DET) curve to plot the fraction of correctly detected acceptable words against the percentage of correctly rejected unacceptable words. (The closer to the top right-hand corner the curve, the more effective the technique.)

4.3. Obtaining a GOP benchmark

GOP is typically applied at the phone-level. In order to score individual words – rather than phones – we create a combined score in two different ways: by either obtaining a GOP score per phone and averaging over all phones in the word (*phone-normalised GOP*) or obtaining a GOP score per frame and averaging over all frames in the word (*frame-normalised GOP*).

As we use triphones for DP scoring while most GOP studies tend to use monophones rather than triphones (seemingly to circumvent the irregularities introduced by the need for subsequent triphones to match contexts), we produce results both for monophones and triphones. (For the monophone models, we increase the Gaussian mixtures to 256 per state.)

5. RESULTS

5.1. Validating Afrikaans

Our approach, as described in Section 3, assigns a score between +1 and -1 to each word. A +1 score indicates that the observed phone string is highly likely to be a rendition of the target word; a -1 score that this is highly unlikely. (Note that a perfect match between reference and observed string is not required to obtain a +1 score).

We first demonstrate the implications of using the different evaluation strategies listed in Table 1 by comparing the DET curves for a single scoring technique (here PDP with a flat scoring matrix). The resulting figure (Fig. 1) shows that all three classification schemes can be utilised, with 'strict' classification the most difficult to score correctly, and 'lenient' the easiest, as expected. The 'harvest' classification strategy – which is probably the closest to what is required for the specific application studied here – lies somewhere in between.

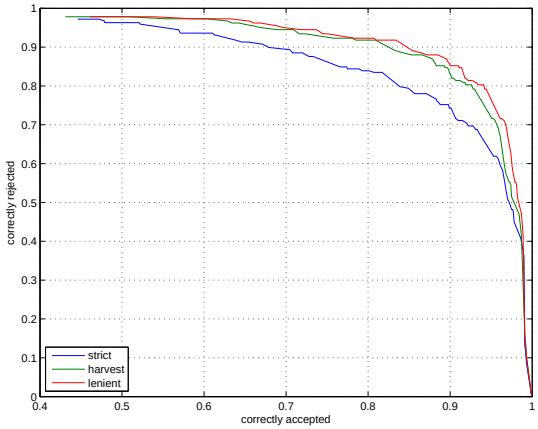

Fig. 1. *DET curves for the three classification schemes listed in Table 1, obtained using Afrikaans data and PDP scoring with a flat matrix.*

Next, we compare the four GOP variants in order to select a suitable benchmark. The four DET curves in Fig. 2 correspond to phone-normalised and frame-normalised GOP scoring (as described in 4.3 above), using either monophones or triphones to generate acoustic likelihoods. We see that – for all threshold settings – phone-normalised GOP outperforms the frame-normalised method, and that triphones outperform monophones.

Finally, in Fig. 3 we compare our best GOP candidate (frame-based scoring using triphones) with PDP using either a flat or a trained matrix. The trained matrix was generated as described in section 3.2. We see that effective rejection of unacceptable words is achieved by all three methods: for example, if we want to ensure that 90% of any unacceptable utterances are rejected, we are still able to retain around 80% of

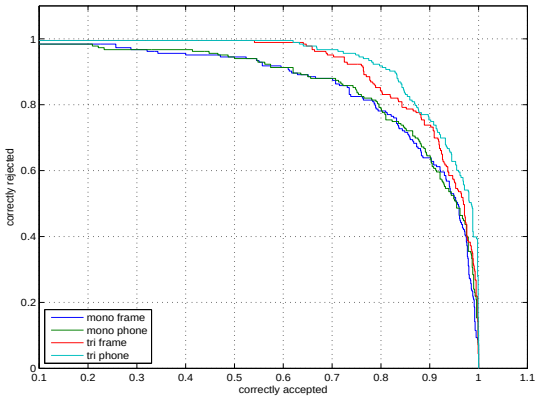

Fig. 2. *'harvest' DET curves for four GOP variants: monophone-based (mono) or triphone-based (tri), and phone-normalised (phone) or frame-normalised (frame), obtained using Afrikaans data.*

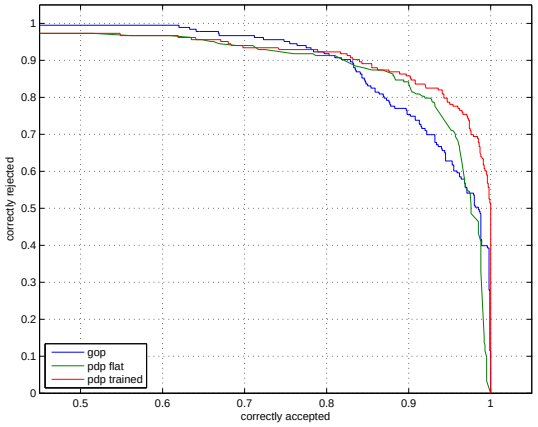

Fig. 3. *'harvest' DET curves on Afrikaans data for PDP with a flat and trained matrix, compared to the best performing GOP variant (phone-normalised, triphone-based).*

the good audio with all the methods. Interestingly, while PDP has the higher equal error rate and perform better at high correct acceptance values, GOP outperforms PDP at high correct rejection values. This behaviour is confirmed by viewing the performance of the same three methods when using 'strict' scoring (Fig. 4); now GOP clearly outperforms PDP at high correct rejection values.

It should be noted in Fig. 3 that the PDP methods are not able to attain a perfect rejection rate at any threshold. This is in contrast with GOP, which is able to reject all wrong words while retaining about 60% of the correct audio. The reason for this can be traced back to five utterances marked as 'bad audio' by the human verifiers, but portions of which are decoded perfectly by the speech recogniser (resulting in a perfect phone string, and a perfect score for the relevant words). Whether or not such words should be flagged as problematic

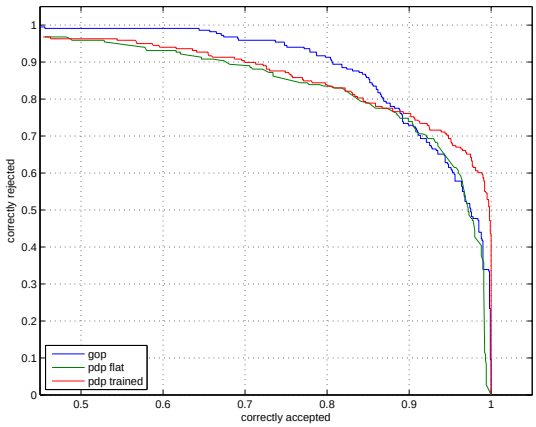

Fig. 4. *'strict' DET curves on Afrikaans data for PDP with a flat and trained matrix, compared to the best performing GOP variant (phone-normalised, triphone-based).*

or not, is not immediately clear. (While our evaluation protocol requires that they are flagged as errors, there is little chance of such utterances hurting acoustic model training.)

A better intuition of the function of the trained matrix can be obtained from Fig. 5. Each block corresponds to $score(r, o)$, with red values close to +1, and blue values close to -1. The left-hand column is associated with deletions, and the top row with insertions.

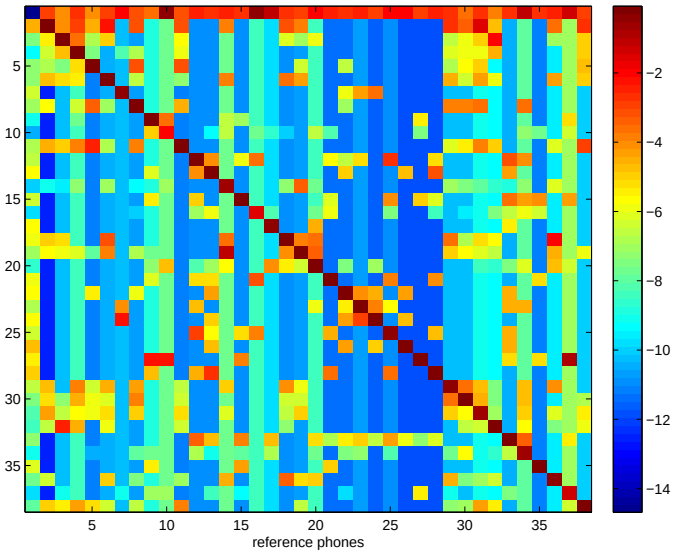

Fig. 5. *PDP scoring matrix trained on the Afrikaans data.*

5.2. Validating isiNdebele

When extracting similar DET curves for isiNdebele (Fig. 6), we again see that the automated validation is successful, with DET curves following fairly similar trends, except that trained

PDP and GOP are now much closer competitors. Again, to ensure that at least 90% of unacceptable utterances are rejected, around 80% of the good utterances are retained with the best method (trained PDP at that operating point).

An error analysis of the isiNdebele data shows that the speakers evaluated generated very few errors overall: only about 5% of words contained some form of error, while the Afrikaans corpus contained approximately 17% errors. (The isiNdebele task itself is therefor more suitable to methods performing well at higher rejection rates.)

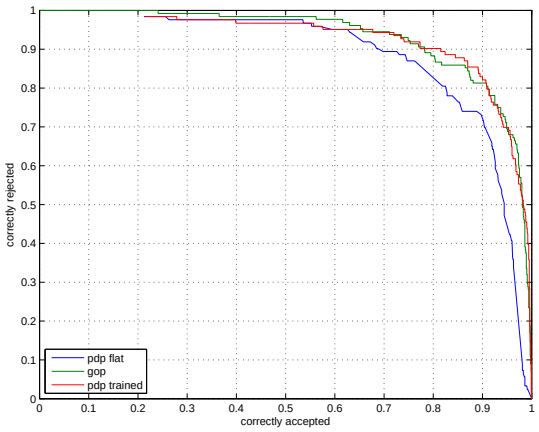

Fig. 6. *'harvest' DET curves on isiNdebele data for PDP with a flat and trained matrix, compared to phone-normalised, triphone-based GOP (the best performer of the GOP variants on isiNdebele data).*

6. CONCLUSION

With the availability of tools such as Woefzela, and the increasing affordability of smartphones, speech corpus collection in under-resourced languages can be performed quickly and efficiently. As comprehensive language models are typically not as readily obtainable, techniques are required for processing speech corpora without relying on extensive language models. In this work, we compared the effectiveness of two such phone-based approaches to corpus validation: a word-based variant of Goodness of Pronunciation (GOP) and a new phone-based dynamic programming (PDP) scoring technique. We found that variants of both techniques provide results that are effective for corpus validation. Results are comparable, with PDP scoring using a trained scoring matrix producing better results at high acceptance rates (and when requiring more lenient scoring); and phone-normalised, triphone-based GOP producing better results at high rejection rates (and when requiring stricter scoring).

The techniques described in this paper were used to extract trusted subsets from Woefzela-collected data in all eleven of South Africa's official languages. In future work, we aim to use these techniques to assist us with dictionary refinement in the same eleven languages: both by scoring existing pronunciations and by suggesting and scoring variant pronunciations observed in the data. We are specifically interested in improving the pronunciation of code-switched words (mostly English words embedded in the matrix language) as our current pronunciations of these words are fairly crude. We expect the process of dictionary improvement and transcription selection to interact, with improved dictionaries leading to larger portions of the corpus that can be salvaged (while making sure that erroneously added dictionary entries do not lead to inflated corpus acceptance rates).

An additional avenue that we have not yet explored relates to the use of phonotactic transformations prior to DP scoring. Since the phone strings themselves are being manipulated, typical phonotactic effects (such as the end-of-word deletions common to the Bantu languages) can be modelled explicitly to produce additional candidates prior to DP scoring or to score phones differently based on word position.

7. ACKNOWLEDGEMENTS

This work was supported by the Department of Arts and Culture of the government of the Republic of South Africa, and utilised data developed in collaboration with the NCHLT project team at the CSIR Meraka Institute. All support is gratefully acknowledged.

8. REFERENCES

[1] M. Gales and S. Young, "The application of Hidden Markov Models in speech recognition," *Foundations and Trends in Signal Processing*, vol. 1, no. 3, pp. 195–304, 2008.

[2] E. Barnard, J. Schalkwyk, C. van Heerden, and P.J. Moreno, "Voice search for development," in *Proc. Interspeech*, 2010, pp. 282–285.

[3] Lori Lamel, Jean-Luc Gauvain, and Gilles Adda, "Lightly supervised and unsupervised acoustic model training," *Computer Speech and Language*, vol. 16, pp. 115–129, 2002.

[4] T. J. Hazen, "Automatic alignment and error correction of human generated transcripts for long speech recordings," in *Proc. Interspeech*, Sept. 2006, pp. 1606–1609.

[5] M. H. Davel, C. van Heerden, N. Kleynhans, and E. Barnard, "Efficient harvesting of Internet audio for resource-scarce ASR," in *Proc. Interspeech*, Aug. 2011, pp. 3153–3156.

[6] G. Parent and M. Eskenazi, "Speaking to the crowd: looking at past achievements in using crowdsourcing for speech and predicting future challenges," in *Proc. Interspeech*, Aug. 2011, pp. 3037–3040.

[7] T. Hughes, K. Nakajima, L. Ha, P. Moreno, and M. LeBeau, "Building transcribed speech corpora quickly and cheaply for many languages," in *Proc. Interspeech*, Makuhari, Japan, Sept. 2010.

[8] N. J. De Vries, J. Badenhorst, M. H. Davel, E. Barnard, and A. De Waal, "Woefzela - an open-source platform for ASR data collection in the developing world," in *Proc. Interspeech*, Aug. 2011, pp. 3177–3180.

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

[10] H. Jiang, "Confidence measures for speech recognition: a survey," *Speech communication*, vol. 45, no. 4, pp. 455–470, 2005.

[11] N. T. Vu, F. Kraus, and T. Schultz, "Multilingual A-stabil: A new confidence score for multilingual unsupervised training," in *IEEE Spoken Language Technology Workshop (SLT)*, 2010, pp. 183–188.

[12] Joel Pinto, Igor Szoke, S.R.M Prasanna, and Hynek Hermansky, "Fast approximate spoken term detection from sequences of phonemes," in *Proc. ACM SIGIR Workshop on Searching Spontaneous Conversational Speech*, 2008.

[13] S. M. Witt and S. J. Young, "Phone-level pronunciation scoring and assessment for interactive language learning," *Speech communication*, vol. 30, no. 2-3, pp. 95–108, 2000.

[14] E. Barnard, M. Davel, C. van Heerden, N. Kleynhans, and K. Bali, "Phone recognition for spoken web search," in *CEUR Workshop Proceedings*, MediaEval 2011 Multimedia Benchmark Workshop, Pisa, Italy, Sept. 2011, vol. 807.

[15] Florian Metze, Nitendra Rajput, Xavier Anguera, Marelie Davel, Guillaume Gravier, Charl van Heerden, Gautam V. Mantena, Armando Muscariello, Kishore Prahallad, Igor Szke, and Javier Tejedor, "The Spoken Web Search task at MediaEval 2011," .

[16] S. Srinivasan and D Petkovic, "Phonetic confusion matrix based spoken document retrieval," in *Proc. ACM SIGIR Conference on Research and Development on Information Retrieval*, 2000, pp. 81–87.

[17] M. H. Davel and F. de Wet, "Verifying pronunciation dictionaries using conflict analysis," in *Proc. Interspeech*, Makuhari, Japan, Sept. 2010, pp. 1898–1901.

[18] M. Davel and O. Martirosian, "Pronunciation dictionary development in resource-scarce environments," in *Proc. Interspeech*, Brighton, UK, Sept. 2009, pp. 2851–2854.

[19] M. Davel and E. Barnard, "Pronunciation prediction with Default&Refine," *Computer Speech and Language*, vol. 22, pp. 374–393, 2008.