Automatic speech recognition using probabilistic transcriptions in Swahili, Amharic, and Dinka

Amit Das[⋆], *Preethi Jyothi*[⋆], *Mark Hasegawa-Johnson*

Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Illinois, IL 61801, USA
{amitdas, pjyothi, jhasegaw}@illinois.edu

Abstract

In this study, we develop automatic speech recognition systems for three sub-Saharan African languages using probabilistic transcriptions collected from crowd workers who neither speak nor have any familiarity with the African languages. There is a language mismatch in this scenario. More specifically, utterances spoken in African languages were transcribed by crowd workers who were mostly native speakers of English. Due to this, such transcriptions are highly prone to inaccuracies in labels. The three African languages in consideration are Swahili, Amharic, and Dinka. First, we use a recently introduced technique called mismatched crowdsourcing which processes the raw crowd transcriptions through merging, contextual weighting, and ranking. Next, we adapt multilingual HMM and DNN systems using the probabilistic transcriptions of the African languages. Finally, we report the results using both deterministic and probabilistic phone error rates. Automatic speech recognition systems developed using this recipe are particularly useful for low resource languages where there is limited access to linguisitc resources and/or transcribers in the native language.

Index Terms: mismatched crowdsourcing, cross-lingual speech recognition, deep neural networks, African languages

1. Introduction

This work is focussed on knowledge transfer from multilingual data collected from a set of source (train) languages to a target (test) language that is mutually exclusive to the source set. More specifically, we assume that we have easy access to native transcripts in the source languages but that we do not have native transcripts in the target language. However, mismatched transcripts for the target language (i.e. transcriptions in a different orthography) can be easily obtained from crowd workers on platforms such as Amazon's Mechanical Turk[1] and Upwork.[2] An automatic speech recognition (ASR) system trained using these non-native transcripts in the target language can be particularly useful for low-resource African languages as it circumvents the need to find native transcribers.

We elaborate on some terminology used in this paper. Deterministic transcripts (DT) refer to ones collected from native speakers of a language. We assume no ambiguity in these ground truth labels, and hence they are deterministic in nature. As an example, the DT for the word "cat", after converting the labels to IPA phone symbols, can be represented as shown in Fig. 1 with each arc representing a symbol and a probability value. Here, each symbol occurs with probability 1.0. On the other hand, the term probabilistic transcript (PT) means that the transcript is probabilistic or ambiguous in nature. Such transcripts frequently occur, for example, when collected from crowd workers who do not speak the language they are transcribing [1]. Usually a training audio clip (in some target language L) is presented to a set of crowd workers who neither speak L nor have any familiarity with it. Due to their lack of knowledge about L, the labels provided by such workers are inconsistent, i.e., a given segment of speech can be transcribed using a variety of labels. This inconsistency can be modeled as a probability mass function (pmf) over the set of labels transcribed by crowd workers. Such a pmf can be graphically represented by a confusion network as shown in Fig. 2. Unlike the DT in Fig. 1 which has a single sequence of symbols, the PT has 3×4×3×4 = 144 possible sequences, one of which could be the right sequence. In this case, it is "k æ ∅ t".

Collecting and processing PTs for audio data in the target language L from crowd workers who do not understand L is called *mismatched crowdsourcing* [1]. The objective of this study is to present a complete ASR training procedure to recognize African languages for which we have PTs but no DTs. The following low resource conditions outline the nature of the data used in this study:

- PTs in Target Language: PTs in the target language L are collected from crowd workers who do not speak L.
- PTs are limited: The amount of PTs available from the crowd workers is limited to only 40 minutes of audio.
- Zero DT in Target Language: There are no DTs in L.
- DTs only in Source Languages: There are DTs from other source languages ($\neq L$).
- DTs are limited: The DTs are worth about 40 minutes of audio per language. Hence, the total amount of multilingual DTs available for training is 2 hours. (40 minutes/language $\times$ # languages)

2. Sub-Saharan African Languages

2.1. Swahili

Swahili is a widely spoken language in Southeast Africa with over 15 million speakers. Swahili's written system uses a variant of the Latin alphabet; it consists of digraphs (other than the standard ones like ch, sh, etc.) corresponding to prenasalized consonants that appear in many African languages. Swahili has only five vowel sounds with no diphthongs.

⋆first authors
[1]http://www.mturk.com
[2]http://www.upwork.com

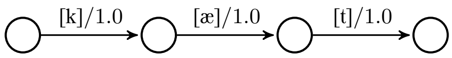

Figure 1: A deterministic transcription (DT) for the word *cat*.

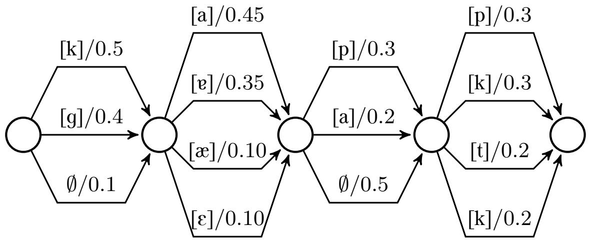

Figure 2: A probabilistic transcription (PT) for the word *cat*.

2.2. Amharic

Amharic is the primary language spoken in Ethiopia with over 22 million speakers. The Amharic script has more than 280 distinct characters (or *fidels*) representing various consonant+vowel sounds. Ejective consonants and labialized sounds are special characteristics of Amharic's phonology. There are seven vowels, thirty one consonant sounds in Amharic and no diphthongs. More details of Amharic phonology are in [2].

2.3. Dinka

Dinka is a Western Nilotic language which is a member of the family of Nilo-Saharan languages. It is spoken by over 2 million people living in South Sudan. The four major dialects are Padang, Rek, Agar, and Bor of which the Rek dialect is considered the standard dialect of Dinka. This study is based on the Rek dialect. The orthography of Dinka closely follows its pronunciation. There are 33 alphabet symbols in the Dinka orthography which are borrowed from a mixture of Latin and IPA alphabets [3]. Furthermore, 4 out of the 33 symbols are digraphs. The Dinka phonology consists of 7 vowels and 20 consonants [4], described in more detail below.

The set of vowels comprises {/a/, /e/, /ɛ/, /i/, /o/, /ɔ/, /u/}. Coincidentally, the orthographic symbols of these vowels are the same as their corresponding phonemic IPA symbols. The vowels often have a creaky quality. With the exception of /u/, these vowels could also have a breathy quality. For example, the breathy version of /a/ is /a̤/, orthographically represented as ä. The breathy vowels are characterized by lower F1 values. Compared to breathy vowels, creaky vowels have relatively more energy at higher frequencies. Vowel lengths can be short or long. Orthographically, long vowels are usually indicated by repeating the letter twice. For example, the word *nëë* is pronounced as /n/ /e̤ː/.

The 20 Dinka consonants are given in Table 1. Voiced and voiceless plosives occur at five places of articulation gradually moving from external to internal portions of the mouth - labial, dental, alvelolar, palatal, and velar. Nasals follow a similar pattern. Interestingly, there is only one fricative. The 4 digraphs *dh*, *nh*, *th*, *ny* translate to /d̪/, /n̪/, /t̪/, /ɲ/ phonemes, respectively.

Table 1: Consonants in the Dinka language

Manner	Place				
	Lab	Den	Alv	Pal	Vel
Plosive	p b	t̪ d̪	t d	c ɟ	k g
Nasal	m	n̪	n	ɲ	ŋ
Trill			r		
Fricative					ɣ
Approx.	w			j	
Lat. Approx.			l		

3. Training an ASR system using Probabilistic Transcripts

3.1. Data

Multilingual audio files were obtained from the Special Broadcasting Service (SBS) network [5] which publishes multilingual radio podcasts in Australia. These data include over 1000 hours of speech in 68 languages. The following languages were used in our experiments - Swahili (SW), Amharic (AM), Dinka (DI), Hungarian (HG), Cantonese (CA), Mandarin (MD), Arabic (AR), Urdu (UR). However, only the sub-Saharan languages - Swahili, Amharic, Dinka - were considered as the target languages. The remaining languages represent the set of source languages. The podcasts were not entirely homogeneous in the target language and contained utterances interspersed with segments of music and English. An HMM-based language identification system was used to isolate regions that correspond mostly to the target language. These long segments were then split into smaller 5-second chunks. The short segments make it easier for crowd workers to annotate since they are unfamiliar with the utterance language. More than 2500 Turkers participated in these tasks, with roughly 30% of them claiming to know only English. The remaining Turkers claimed to know other languages such as Spanish, French, German, Japanese, and Chinese. Since English was the most common language among crowd workers, they were asked to annotate the sounds in the 5-second utterances using English letters that most closely matched the audio. The sequence of letters were not meant to correspond to meaningful English words or sentences as this was found to be detrimental to the final performance [6]. PTs and DTs, worth about 1 hour of audio, were collected from crowd workers and native transcribers respectively. Thus, the training set consists of a) about 40 minutes of PTs in the target language and, b) about 40 minutes of DTs in other source languages excluding the target language. The development and test sets were worth roughly 10 minutes each.

To accumulate the PTs, each utterance was transcribed by 10 distinct Turkers. First the letters in the transcripts were mapped to IPA symbols using a misperception G2P model learned from the source languages. More specifically, the misperceptions of the crowd workers were approximated by letter-to-phone mappings learned from mismatched transcripts and their corresponding DTs in the *source languages*. No target language data is used while estimating the misperception G2P model since we assume there are no DTs in the target language. Multiple mismatched transcripts, collected for the same utterance, were then merged into a compact structure by aligning the sequences (after defining equivalence classes for similar sounds). The process of creating PTs is detailed further in [1].

To accumulate DTs, the same set of utterances were labeled by native transcribers in the utterance language. DTs were mainly accumulated for data in the source languages; these were used in the estimation of the mismatched G2P model. For ASR evaluation purposes, DTs were also acquired for a small amount of development/evaluation data in the target languages. For the words in the DTs, canonical pronunciations of the words were derived from a lexicon. If a lexicon was not available, a language specific G2P model was used [7].

Next, language dependent phones were merged into a compact multilingual phone set to enable data sharing across lan-

Table 2: SBS Multilingual Corpus

Language	Utterances		Phones
	Train	Test	
Swahili (swh)	463	123	53
Amharic (amh)	516	127	37
Dinka (din)	248	53	27
Hungarian (hun)	459	117	70
Cantonese (yue)	544	148	37
Mandarin (cmn)	467	113	57
Arabic (arb)	468	112	51
Urdu (urd)	385	94	45
All	-	-	82

Table 3: PERs of monolingual HMM and DNN models. Dev set in parentheses.

Lang	PER (%)	
	HMM	DNN
swh	35.63 (47.00)	34.18 (39.49)
amh	51.90 (48.68)	46.63 (43.92)
din	51.56 (47.03)	48.58 (48.40)

guages. Language specific diacritics such as tones and stress markers tend to make the phone symbols unique to a particular language. Therefore, diacritics were removed.

There are two distinct features unique to Swahili consonants (among our chosen set of languages): implosive sounds and prenasalized sounds. In addition, Swahili does not distinguish implosive versus explosive stops. To build the multilingual phone set, the implosive sounds were merged with their corresponding non-implosive counterparts (e.g. ɓ → b, ɗ → d). The prenasalized consonants were written as phone pairs combining a nasal sound with the consonant sound (i.e. ᵐb → m b). Amharic's phonology has a particularly distinct feature: ejective consonants. Hence, it does distinguish ejective versus aspirated stops. Nevertheless, we merge them (e.g. t' → t^h, p' → p^h) to allow for cross-lingual transfer. Labialized sounds in Amharic were written as the base sound preceded by the voiced labio-velar approximant sound, w (e.g. a^w → w a). As for Dinka, since breathy vowels are very specific to Dinka, all breathy vowels were mapped down to the regular vowels. For example, a̤ → a. The long vowels ɛː and oː were mapped by repeating the symbols twice: ɛː → ɛɛ, oː → oo. In addition, the dental nasal was mapped to the alveolar nasal: n̪ → n.

Finally, phone based language models (LMs) for Swahili were built from text available on the web. For Amharic and Dinka, phone LMs were built from the DTs although these could also be built using web resources. In all experiments, phone error rates (PER) are evaluated. The corpus is summarized in Table 2 with the language acronyms borrowed from ISO 639-3 codes.

3.2. Monolingual HMM and DNN

We first build monolingual Gaussian mixture (GMM) based hidden Markov models (HMM) and deep neural network (DNN) models trained using DTs in the target language. This is an oracle baseline since it assumes the ideal scenario of DTs in the target language being available during training time. This baseline is an estimate of the best possible (lower bound) PER.

Context-dependent GMM-HMM acoustic models were trained using 39-dimensional Mel frequency cepstral coefficients (MFCC) features which include the delta and acceleration coefficients. Temporal context was included by splicing 7 successive 13-dimensional MFCC vectors (current +/- 3 frames) into a high dimensional supervector and then projecting the supervector to 40 dimensions using linear discriminant analysis (LDA). Using these features, a maximum likelihood linear transform (MLLT) [8] was computed to transform the means of the existing model. The forced alignments obtained from the LDA+MLLT model were further used for speaker adaptive training (SAT) by computing feature-space maximum likelihood linear regression (fMLLR) transforms [9] per subset of speakers. The LDA+MLLT+SAT model is the final HMM model that will be simply referred to as HMM in all experiments. The forced aligned senones obtained from the HMM were treated as the ground truth labels for DNN training.

Table 4: PERs of multilingual HMM and DNN models. Dev set in parentheses.

Lang	PER (%)		
	HMM	DNN	# Senones
swh	65.73 (67.58)	61.17 (63.12)	1003
amh	68.40 (68.20)	66.53 (65.39)	987
din	66.89 (67.24)	64.78 (65.15)	1002

For DNN training, we start with greedy layer-wise Restricted Boltzmann Machines (RBMs) unsupervised pretraining since this leads to better initialization [10]. Then the DNNs were fine-tuned using supervised cross-entropy training. The DNNs were trained using 6 hidden layers with 1024 nodes per layer. All experiments were conducted using the Kaldi toolkit [11]. The monolingual PERs over a total of about 7K-8K phones are given in Table 3.

3.3. Multilingual HMM and DNN

DTs from the source languages were used to train multilingual HMMs and DNNs. Since we assume zero DTs in the target language during training, the DTs used for training multilingual HMM and DNN exclude any data in the target language. The steps for building HMM and DNN systems were the same as in Section 3.2 except that the training data consists of multilingual DTs. The PERs are given in Table 4. Unsurprisingly, due to the lack of DTs in the target language, the PERs are much higher than the oracle monolingual case in Table 3. Hence, the PERs in Table 4 establish the upper bound of PERs. In all subsequent experiments, our goal is to start from the upper bound of PERs in Table 4 and attempt to approach the lower bound PERs in Table 3.

3.4. PT Adapted MAP-HMM

In this step, the multilingual systems in Section 3.3 are adapted using only the PTs in the target language since DTs are not available for adaptation. The multilingual HMM can be adapted using maximum aposteriori (MAP) adaptation described in more detail in [12]. We briefly review the steps here. The goal is to obtain meaningful adaptation data using the PTs. For our implementation, we follow the Weighted Finite Transducer (WFST) [13] framework both during training and testing. The ASR search graph is represented as a WFST mapping the acoustic signal to a sentence and is defined by the composition $H \circ C \circ L \circ G$ where H maps a sequence of HMM states to a triphone sequence, C maps triphone to monophone sequences, L maps monophone sequences to words (pronunciation model) and G reorders the resulting word sequence (language model). Since our tasks involve phone recognition, L is an identity mapping of phones and G is a phone N-gram model. In the case of

Table 5: PERs of multilingual DNN (MULTI), MAP adapted HMM (MAP-HMM), and adapted DNNs (DNN-1, DNN-2). First element in parentheses is the PER of the dev set. Second element is the absolute improvement in PER of the test set over the MULTI system.

Lang	PER (%)			
	MULTI-DNN	MAP-HMM	DNN-1	DNN-2
swh	61.17 (63.12, 0.0)	44.77 (50.97, 16.4)	45.14 (47.83, 16.03)	**43.03 (45.87, 18.14)**
amh	66.53 (65.39, 0.0)	61.95 (62.15, 4.58)	61.64 (61.43, 4.89)	**59.48 (59.61, 7.05)**
din	64.78 (65.15, 0.0)	59.58 (59.71, 5.20)	59.33 (60.97, 5.45)	**58.22 (60.86, 6.56)**

DTs, the training graph for a transcript DT is constructed using $H \circ C \circ L \circ DT$ where DT is a linear chain acceptor representing a *single* sequence of phones. In the case of PTs, the training graph is $H \circ C \circ L \circ G \circ PT$ where PT is a *confusion network* of phones (similar to Fig. 2) obtained from crowd workers. Considering the PTs as adaptation transcripts, the sufficient statistics required for MAP adaptation are obtained from the lattice $H \circ C \circ L \circ G \circ PT$. There is no change in the testing stage, i.e., we look for the 1-best path in the decoding lattice $H \circ C \circ L \circ G$. The PER results for the MAP adapted HMM are under the column heading MAP-HMM in Table 5. The PER results for the multilingual DNN, under the column heading MULTI-DNN in Table 5, is replicated from Table 4 for comparison purposes.

3.5. PT Adapted DNN

We briefly review different strategies for DNN adaptation using PTs. These are illustrated in Fig. 3 and described in greater detail in [14]. In Fig. 3(a), the softmax layer of the multilingual DNN in Section 3.3 is replaced by another randomly initialized softmax layer while the shared hidden layers (SHLs) of the multilingual DNN are retained. The resulting DNN is fine tuned using the PT alignments generated by the MAP adapted HMM from Section 3.4. This is the conventional way to adapt a DNN using DTs [15]. However, this approach does not work very well for PTs largely due to the presence of incorrect labels in PTs. The results for DNN-1 are under the column heading DNN-1 in Table 5. The performance of DNN-1 is worse than MAP-HMM for Swahili and only marginally better for the other languages. To alleviate the effect of incorrect labels, the DNN-2 system of Fig. 3(b) is used. In this approach, two separate softmax layers are used. The first softmax layer is trained with target language PTs only whereas the second softmax layer is trained with multilingual DTs. In Fig. 3(c), there is a third softmax layer trained using self-training transcripts (ST). Here, the DNN-2 system decodes some additional unlabeled audio in the target language and then uses a subset of the decoded labels, with high posterior probabilities (confidences), to retrain itself in the target language. The self-training algorithm is a semi-supervised algorithm to train DNNs [16]. We report the results only for the DNN-2 system in Table 5. The absolute decrease in PER compared to DNN-1 is consistent and is in the range 1.11%-2.16%. Comparing the most adapted system (DNN-2) with the unadapted system (MULTI), the total absolute decrease in PER is in the range 6.56%-18.14%.

3.6. Probabilistic Error Rate

The aforementioned sections computed the phone error rates by measuring the edit distance between the 1-best path in the ASR decoding lattice and the reference DT. Hence, they may be considered as deterministic PERs. Our assumption was that there were no DTs in the target language in the training stage. Thus, it is fair to assume that there may not be DTs in the target language in the testing stage as well. An obvious question is how do we

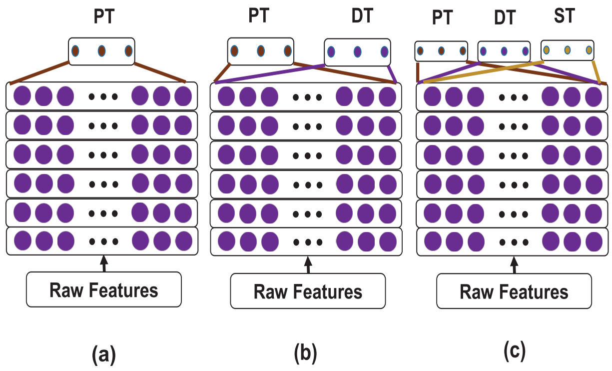

Figure 3: DNN adaptation using probabilistic transcripts (PT).

Lang	PPER (%)	
	MULTI - MAP-HMM	MULTI - DNN-2
swh	??-??=??	?? - ?? = ??
amh	64.46 - 57.16 = 7.3	64.46 - 59.80 = 4.67
din	64.09 - 58.59 = 5.5	64.09 - 60.64 = 3.45

Table 6: Probabilistic Phone Error Rates

evaluate ASR systems for the target language in the absence of DTs? In the absence of DTs, we consider PTs to serve as a proxy for the reference ground truth labels. We denote the edit distance between the 1-best path in the ASR decoding lattice and the PTs as probabilistic phone error rate (PPER). This is calculated as follows. First, the PTs are pruned to retain the most reliable transcripts. Next, the probabilities on the arcs of the pruned PTs are stripped making the PTs unweighted. Finally, the edit distance between the 1-best path in the ASR decoding lattice and the unweighted pruned PT is computed. The PPERs are reported in Table 6. Comparing the MAP-HMM and the MULTI systems, the absolute decrease in PPER in Table 6 correlates well with the absolute decrease in PER in Table 5 (refer to the second elements in parentheses under the column MAP-HMM). In addition, the PPER of DNN-2 also outperforms the MULTI system. (However, as opposed to the behavior in PER, the PPER of DNN-2 is higher than MAP-HMM.) Thus, PPERs allow us to correlate the improvements of the adapted systems over the unadapted ones; these improvements are verified to be accurate by PER computations in Table 5.

4. Conclusions

In this study, we presented a complete set of ASR training methods to train HMM and DNN systems using no deterministic but only probabilistic transcripts in Swahili, Amharic, and Dinka. We reported absolute phone error rate improvements of the PT adapted systems in the range 6.56%-18.14%. In addition, we found improvements in probabilistic error rates can correlate well with the improvements in deterministic phone rates. This is useful in the absence of deterministic transcripts in the test set.

5. References

[1] P. Jyothi and M. Hasegawa-Johnson, "Transcribing continuous speech using mismatched crowdsourcing," in *Interspeech*, 2015.

[2] M. Y. Tachbelie, S. T. Abate, and L. Besacier, "Using different acoustic, lexical and language modeling units for ASR of an under-resourced language – Amharic," vol. 56, pp. 181 – 194, 2014.

[3] Dinka omniglot. [Online]. Available: http://www.omniglot.com/writing/dinka.php

[4] B. Remijsen and C. A. Manyang, "Luanyjang dinka," *Journal of the International Phonetic Association*, vol. 39, no. 1, pp. 113–124, 2009.

[5] Special Broadcasting Service (SBS). [Online]. Available: http://www.sbs.com.au/yourlanguage

[6] P. Jyothi and M. Hasegawa-Johnson, "Acquiring speech transcriptions using mismatched crowdsourcing," in *AAAI*, 2015.

[7] M. Hasegawa-Johnson, "WS15 dictionary data," 2015, downloaded 9/24/2015 from http://isle.illinois.edu/sst/data/dict.

[8] R. Gopinath, "Maximum likelihood modeling with Gaussian distributions for classification," in *ICASSP*, 1998, pp. 661–664.

[9] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language.*, vol. 12, pp. 75–98, 1997.

[10] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Adv. in Neural Information Processing Systems*, 2006.

[11] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Veselý, "The Kaldi speech recognition toolkit," in *IEEE ASRU Workshop.*, 2011.

[12] C. Liu, P. Jyothi, H. Tang, V. Manohar, R. Sloan, T. Kekona, M. Hasegawa-Johnson, and S. Khudanpur, "Adapting ASR for under-resourced languages using mismatched transcriptions," in *ICASSP*, 2016.

[13] F. P. M. Mohri and M. Riley, "Weighted finite-state transducers in speech recognition," *Computer Speech and Language.*, vol. 20, no. 1, pp. 69–88, 2002.

[14] A. Das and M. Hasegawa-Johnson, "An investigation on training deep neural networks using probabilistic transcriptions," in *Submitted to Interspeech*, 2016.

[15] A. Ghoshal, P. Swietojanski, and S. Renals, "Multilingual training of deep neural networks," in *ICASSP*, 2013.

[16] K. Vesely, M. Hannemann, and L. Burget, "Semi-supervised training of deep neural networks," in *IEEE ASRU Workshop.*, 2013, pp. 267–272.