Phone Merging for Code-switched Speech Recognition

Sunit Sivasankaran *
Université de Lorraine, CNRS,
Inria, LORIA,
F-54000 Nancy, France
sunit.sivasankaran@inria.fr

Brij Mohan Lal Srivastava, Sunayana Sitaram,
Kalika Bali, Monojit Choudhury
Microsoft Research - Bangalore, India
{v-bmlals,susitara,kalikab,
monojitc}@microsoft.com

Abstract

Speakers in multilingual communities often switch between or mix multiple languages in the same conversation. Automatic Speech Recognition (ASR) of code-switched speech faces many challenges including the influence of phones of different languages on each other. This paper shows evidence that phone sharing between languages improves the Acoustic Model performance for Hindi-English code-switched speech. We compare *baseline* system built with separate phones for Hindi and English with systems where the phones were manually merged based on linguistic knowledge. Encouraged by the improved ASR performance after manually merging the phones, we further investigate multiple data-driven methods to identify phones to be merged across the languages. We show detailed analysis of automatic phone merging in this language pair and the impact it has on individual phone accuracies and WER. Though the best performance gain of 1.2% WER was observed with manually merged phones, we show experimentally that the manual phone merge is not optimal.

1 Introduction

Multilingual speakers tend to alternate between several languages within a conversation. This phenomenon is referred to as code-switching (CS). Automatic Speech Recognition for CS speech is challenging. Code-switched speech recognition present challenges in acoustic, language and pronunciation modeling of speech. Acoustic Models (AMs) need to model phones in a mixed language setting, where co-articulation effects from one language may influence the other. Moreover, language models needs to be capable of predicting code-switch points between the two languages. The vocabulary size may be double of what is present in monolingual systems. Accents and native language influence may pose challenges to pronunciation models. Another major challenge in building code-switched ASR is the lack of data for different language-pairs. To curb the issue of contextual data availability per phone, we study the effect of manual merging and two automatic merging over the performance of Hindi-English code-switched speech recognition system.

In systems with a small amount of data for training the AMs, phones that are similar to each other in the two languages being mixed can be merged, leading to more data for each phone. This may be especially useful in the case of related languages, or when a strong native language accent is expected to influence pronunciations in the other language. We experiment with phone merging in AMs of Hindi-English code-switched conversational speech, and show that we can get improvements on Word Error Rate (WER) by merging certain phones.

One technique to merge phones in the two languages being mixed is to use a common phoneset such as the International Phonetic Alphabet (IPA), or knowledge from a bilingual speaker to decide which phones can be merged (*manual-merge*). However, this may not always find the optimal merges, particularly if the phoneset we are starting with is not the appropriate representation for the dialect present in the speech. Another technique is to automatically find candidate merges by taking into account phone errors made by the ASR system in presence of a **monolingual context** (*data-driven*). Thirdly, we can create lex-

[*] This work was done while interning at Microsoft Research-India

Proceedings of The Third Workshop on Computational Approaches to Code-Switching, pages 11–19
Melbourne, Australia, July 19, 2018. ©2018 Association for Computational Linguistics

icons with all possible pronunciation variants covering all candidate phones and allow the decoder to choose the correct pronunciation variant during decoding (*probabilistic*). We implement and discuss these techniques for phone merging in Hindi-English code-switched speech recognition.

The paper is organized as follows. Section 2 relates this work to prior work in code-switched speech recognition. Section 3 describes the Hindi-English speech data that we used. We describe our proposed techniques and experiments with phone merging in Section 4 and conclude the paper in Section 5.

2 Relation to Prior Work

Code-switched speech recognition has been studied in the context of acoustic, language and pronunciation modeling. The Language Identification (LID) based approach is to identify the language boundaries and subsequently use an appropriate monolingual ASR system to recognize monolingual fragments (Chan et al., 2004) or run multiple recognizers in parallel with an LID system (Ahmed and Tan, 2012; Weiner, 2012). Another approach is to train an AM on bilingual data (Lyu et al., 2006; Vu et al., 2012) or to use one of the monolingual AMs (Bhuvanagirir and Kopparapu, 2012) or to pool the existing monolingual AMs by sharing phones belonging to both languages. Yeh et al. (Yeh and Lee, 2015) tackle the problem of code-switching in which a speaker speaks mainly in one language, leading to an imbalance in the amount of data available in the two languages, with cross-lingual data sharing approaches. (Pandey et al., 2017) also propose studies to adapt matrix language (monolingual Hindi) resource to build better code-mixed acoustic model in case of read speech.

Yilmaz et al. (Yılmaz et al., 2016) describe two DNN architectures for recognizing Frisian-Dutch code-switched speech. They use language dependent phones in which each phone is tagged with the language and modeled separately. They also use language independent phones by modeling them jointly and merging them on the basis of the associated IPA symbol. In their experiments, they find that the language dependent approach performs better. (Lyudovyk and Pylypenko, 2014) describe an approach for code-switched speech recognition of closely related languages, namely, Ukrainian and Russian by creating a bilingual pronunciation lexicon.

(Chan et al., 2009) describe a two pass approach for Cantonese-English mixed speech recognition, in which they develop a cross-lingual phonetic AM, with the phone set designed based on linguistic knowledge. (Yu et al., 2004) present three approaches for bilingual phone modeling for Mandarin-English speech, namely combining phone inventories, use IPA mappings to construct a bilingual phone set and clustering phones with hierarchical clustering by using the Bhattacharyya distance and the acoustic likelihood. The third approach outperforms the IPA-based mapping and is comparable to the combination of the phone inventories.

A closely related area of research is the Multi-Lingual speech recognition (Toshniwal et al., 2018; Schultz and Waibel, 1997, 2001; Lin et al., 2009; Vu et al., 2014). Though the problems in multi-lingual ASR and ASR for code-switched data seem similar such as; large phonetic space due to the incorporation of the phones of both languages, code-switched ASR has its own set of challenges. For example, even with a large corpus, getting enough data at code-switched points for both the Acoustic Model and the Language Models is very challenging.

3 Data

The dataset used in this work contains conversational speech recordings spoken in code-switched Hindi and English. Hindi-English bilinguals were given a topic and asked to have a conversation about the topic with another bilingual. They were not explicitly asked to code-switch during recording, but around 40% of the data contains at least one English word in an utterance. The conversations were transcribed by bilingual transcribers. Hindi words were transcribed in the native Hindi script Devanagari, and English words in Roman script. There was no distinction made between borrowed words and code-switching, which led to some inconsistencies in transcription. Each conversation was broken down into utterances ranging from very short one word utterances to long sentences.

A summary of the Hindi-English code switched dataset used in our experiments is shown in Table 1. The code-switching statistics mentioned are particular to this dataset and is subject to change depending on the speaker. However, the phone

Data	**Utts**	**# of Spkrs**	**Hrs**	**Total Words**	**En (%)**	**Unique Words**	**En (%)**
Train	41276	429	46	560893	16.6	18900	40.23
Test	5193	53	5.6	69177	16.5	6000	41.01
Dev	4689	53	5.7	68995	16.05	6051	40.04

Table 1: Hindi-English code switched data

merging experiments described in this paper is still relevant due to the performance gains of the acoustic model.

4 Phone merging

In this section, we first describe the baseline ASR system built by combining the Hindi and English phonesets, followed by experiments conducted on phone merging and the resulting impact on Word Error Rate (WER). All experiments were carried out using Kaldi (Povey et al., 2011) and the phone merging techniques are implemented in Python.

4.1 Baseline ASR System

The CMU Pronunciation dictionary (Weide, 1998) containing 39 phones was used as the lexicon for all the English words in the data. This is not the most appropriate choice given that all the speakers in the data speak Indian English, however, due to the lack of a publicly available large Indian English lexicon, we used CMUdict. To generate the lexicon for the Hindi words, we used the Festvox Indic front end (Parlikar et al.), which does basic grapheme to phone conversion for Hindi, including schwa deletion and contextual nasalization. The Hindi phoneset consisted of 50 phones after removing some very low frequency phones.

We used Feature-space maximum likelihood linear regression (fMLLR) (Gales, 1998) to train a context-dependent GMM-HMM acoustic model and a trigram Language Model (LM) built on the transcripts during decoding. With this system, we obtained a baseline Word Error Rate (WER) of 40.36%.

We evaluated the accuracies at phone level by comparing the transition arcs that correspond to phones in the decoded lattices with the alignments obtained from the GMM-HMM model as ground truth. Figure 1 shows a scatter plot of the phone accuracy with respect to the log of the data available per phone in the test dataset. Evidently, we observe higher accuracies for phones with larger count with a few exceptions such as, /nN_HP/, /zh/ and /dR_HP/.

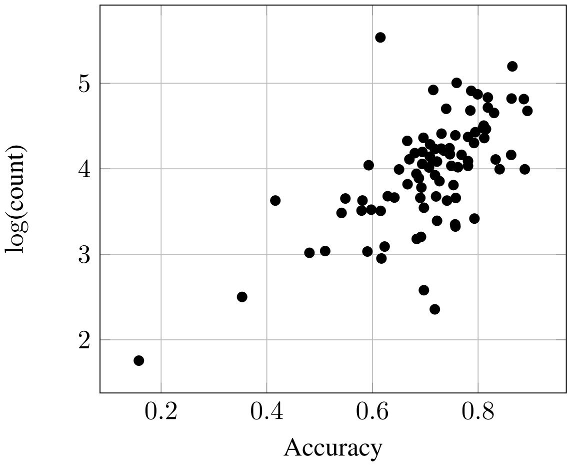


Figure 1: Scatter plot of log of data count per phone with respect to the phone accuracy on the test set.

4.2 Manually merging similar sounding phones

To increase the data availability per phone, we merged similar sounds in both languages even if they are not exactly the same linguistically (in terms of their articulation). The mapping between Hindi phones and CMUdict phones in the Festvox Indic frontend, built for enabling a Hindi TTS system to pronounce English words (Sitaram and Black, 2016), was used for this purpose. All the merged phones (a total of 31) were prefixed with "-HP-M".

To analyze the impact of merging, we started by merging a pair of phones - the English phone *eh* (example "*academic* ae k ah d *eh* m ih k") with a similar sounding Hindi phone *E-HP* (example in Roman script: "*kehana* k-HP *E-HP* hv-HP nB-HP Aa-HP"). This resulted in 38 English phones, 49 Hindi phones and 1 merged phone resulting in 88 unique phones. We obtained a WER of 40.21 using a GMM-HMM acoustic model, which is a negligible improvement over the system with no merging. The bar plot in Figure 2 shows the change in the accuracies for each phones with respect to the baseline. We notice an improvement of 4.18% and 4.5% in the accuracy of the '*eh*' and '*E-HP*' phones respectively. Similar performance was obtained while merging the English phone *n* with the Hindi phone *nB-HP*. In both cases, a decrease in accuracies for a few phones were observed.

Then, we merged all the 31 pairs of similar sounding English and Hindi phones. We refer to this system as the "all merged" system, for which we obtained a WER of 39.7%. There was a noticeable improvement in the accuracies of about 75% of the phones as shown in Figure 3 . We observed a decrease in accuracies for phones which do not have similar sounding equivalents to merge, such as the Hindi phones 'sr-HP' and 'nN-HP' and the English phone 'ng'. Large improvements in the phone accuracies, amounting to around 50%, were observed in merged phones such as 'ow-HP' (merged with the English phone 'ao') and 'tr-HP' (merged with the English phone 't'). Conspicuously, the highest improvements were for phones with low count prior to merging.

4.3 Measuring phone accuracy changes

To evaluate the performance of these systems in terms of phone accuracy, we computed the weighted average of change in accuracies. The weights correspond to the data available per phone. We measure Weighted Average of Phoneme Improvement (henceforth referred to as $WAPI$) as:

$$WAPI = \frac{\sum_i w_i \times \Delta acc_i}{\sum_i w_i},$$

where w_i and Δacc_i represent the data available per phone and change in phone accuracy with respect to the baseline respectively. A summary of the $WAPI$ score for different acoustic models is shown in Table 2. The "all merged" system gives the highest WAPI score of 2.45%, whereas merging 'eh' with 'E-HP' and 'n' with 'nB-HP' results in a WAPI score of 0.31 and 0.47 respectively. Interestingly merging both 'eh' with 'E-HP' and 'n' with 'nB-HP' results in WAPI score of 0.49 which is higher compared to individually merging these phones.

From the above experiments we infer that merging phones result in more data for data-starved phones which in turn improves the phone accuracies.

Phones Merged	WAPI
All	2.45
eh with E-HP	0.31
n with nB-HP	0.47
eh with E-HP & n with nB-HP	0.49
P_{30}	-0.59
P_{70}	1.63

Table 2: Weighted average of phone improvement (WAPI) scores for different phone merging.

4.4 Improved Acoustic Models

We also performed the same experiments using two Deep Neural Network (DNN) based acoustic models. The first model was built using 5 hidden layers with p-norm (Zhang et al., 2014) as the non-linearity. The input dimension of each hidden layer was 5000 and output dimension was 500. We used 40 dimensional MFCC along with 4 splicing frames on each side, appended with 100 dimensional i-vectors (Dehak et al., 2011) as input features. We also built a time-delay neural network (TDNN) (Peddinti et al., 2015) with 6 hidden layers and ReLU as the non linearity.

Merging	GMM	p-norm DNN	TDNN
No merge	40.36	32.81	29.15
All merge	39.70	31.89	28.78
DDPM	52.95	45.99	42.16
DDPM ($\alpha_c(p)$)	41.07	34.69	31.52
DDPM (α_c)	40.75	34.48	31.28
P_{30}	41.21	33.40	29.84
P_{70}	40.92	34.30	28.94

Table 3: Word Error Rates of all systems. DDPM stands for Data Driven Phone Merge. It has 3 variants as mentioned in section 4.5

Table 3 summarizes the WER for the different systems. The all-merge model with p-norm DNN had a WER of 31.89% (with confidence interval of 0.34%). This is 2.8% lower than the baseline DNN model with no merging. The TDNN network outperformed both the GMM and p-norm based DNN AMs. The relative WER improvement of 1.29% using TDNN with merging is statistically significant compared to TDNN model without merging.

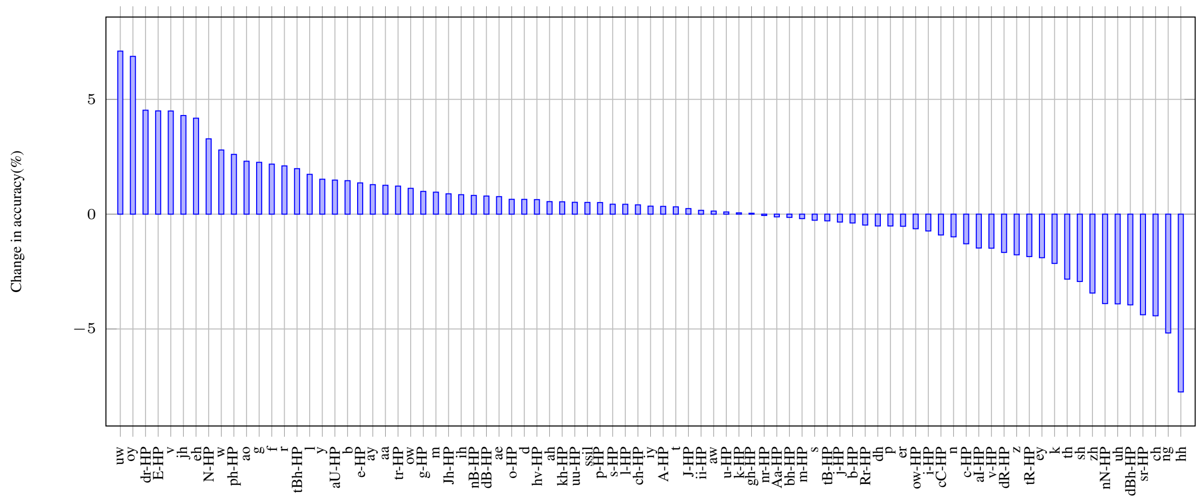


Figure 2: Accuracy changes (%) in the phone accuracies when eh (En) phone is merged with E-HP (Hi) phone

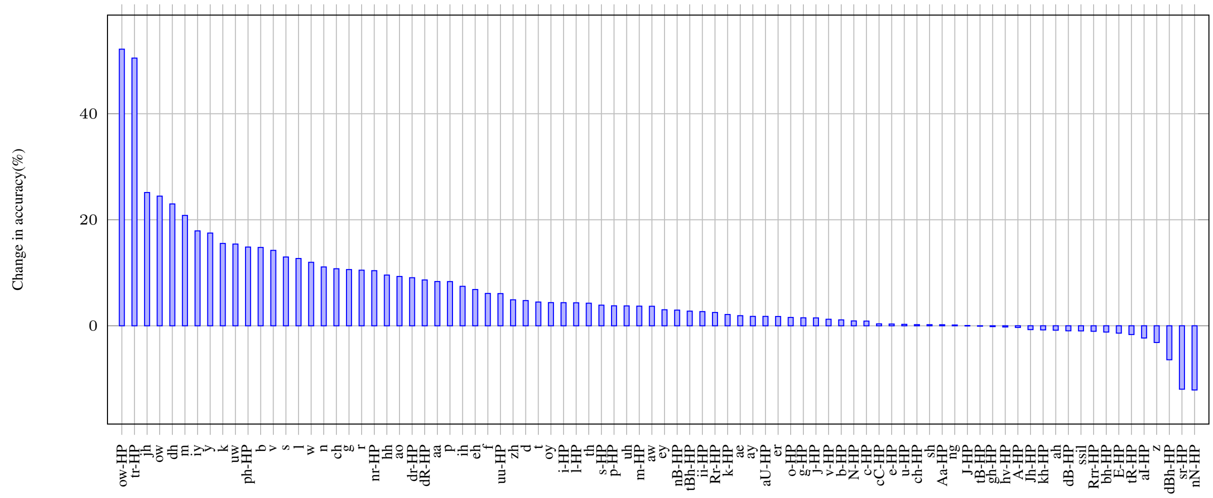


Figure 3: Change in phone accuracies after merging all similar sounding Hindi and English phones.

Motivated by this improved ASR performance, we investigate approaches to merge phones using data driven methods. We hypothesize that the data driven methods will provide us with clues on the acoustic similarity between phones to be merged.

4.5 Data-driven phone merging (DDPM)

The potential phone pairs to merge can be identified based on the errors made by the decoder, with respect to the alignments. A TDNN acoustic model trained using the unmerged (89 phones) phoneset was used to decode the utterances from the dev dataset, with a low LM weight of 1 so as to minimize the influence of LM. Phoneme sequences are then derived by parsing the best path through the decoded lattice. The same acoustic model is also used to align the dev data and obtain the corresponding true phone sequences and their durations. We choose pairs of aligned and decoded utterances with a strict threshold of 80% or more overlap in duration. Using the alignments, we identify English phones which were wrongly decoded as Hindi phones. We call them cross-language swaps. Frequent swapping between English-Hindi phone pair, indicates the need for the pair to be merged.

Using this approach we observe several merges that were present in the manual merge. For instance, (e_HP, ey), (A_HP, ah), (ii_HP, iy), (m_HP, m), (b_HP, b), (ph_HP, f). Errors such as, (Aa_HP, l), (u_HP, ah), (j_HP, ey), (g_HP, l), were also noticed. WER of 52.95, 45.99 and 42.16 were obtained using GMM, DNN and TDNN models respectively, after merging the phones identified by the data-driven method. The decrease in performance can be attributed to the wrong phone swaps.

4.5.1 Inducing context sensitivity through reliability coefficients

Spurious phone swaps degrade the performance of the ASR substantially. This can be reduced by taking into account the phone context. We associate a notion of *context reliability* with each context c, which is defined as the proportion of correct *within-language* phone predictions by the decoder out of all the instances of a context c. We compute this reliability as a coefficient (α_c) for left, right and both contexts in two different ways. α_c can be computed with respect to a specific center phone ($\alpha_c(p)$) as:

$$\begin{aligned} \alpha_c(p) &= \frac{\text{correct instances of } p \text{ with } c}{\text{total instances of } p \text{ with } c} \\ &= P(x = p|c) \end{aligned} \tag{1}$$

An alternate method referred to as global context reliability coefficient, is to compute the context reliability coefficient for every context irrespective of the center phone p. This is obtained by computing the ratio of the counts of the correct instances for any arbitrary phone in presence of context c to the total instances of context c:

$$\alpha_c = \frac{\text{correct instances of context c}}{\text{total instances of context c}} \tag{2}$$

The computed α_c are applied as weights while combining the probability of *cross-language* swaps conditioned on the context c.

Our goal is to compute the conditional probability of the decoded phone(x_d) given the alignment phone (x_a), which is $P(x_d|x_a)$. The phone ($\hat{x_d}$) with the highest probability will be chosen as the potential merge for x_a (eq. 3).

$$\hat{x_d} = \arg\max_{x_d} P(x_d|x_a) \tag{3}$$

The context information is incorporated by computing the conditional probability specific to a context as $P(x_d|x_a, c)$ and then marginalizing over all possible c to obtain the swap probability $P(x_d|x_a)$ (eq. 4).

$$\begin{aligned} P(x_d|x_a) &= \sum_c P(x_d|x_a, c)P(c|x_a) \\ &\approx \sum_c P(x_d|x_a, c)P(x_a|c)P(c) \\ &\approx \sum_c P(x_d|x_a, c)\alpha_c P(c) \end{aligned} \tag{4}$$

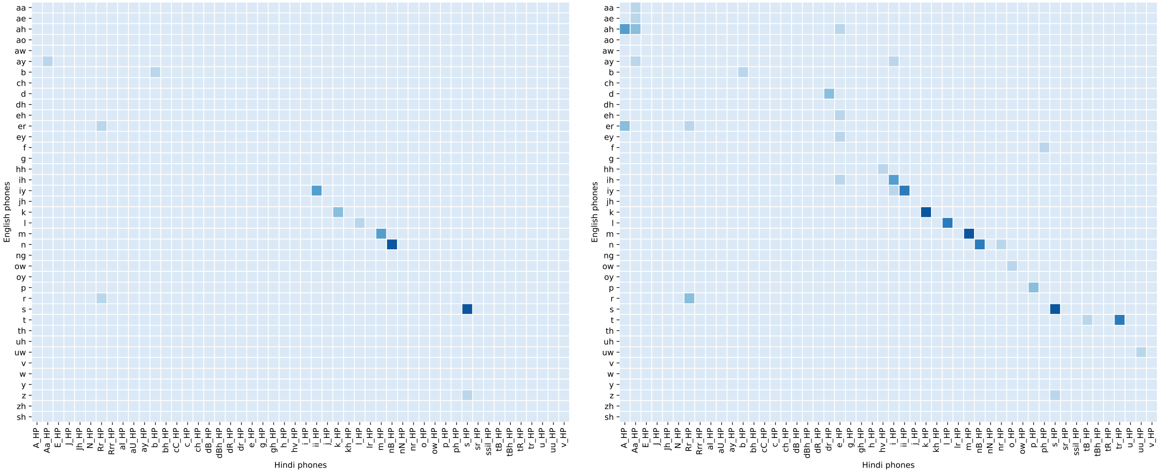
$P(c)$ is the prior probability for each context c which is computed using the dev dataset. We assign a neutral reliability (α_0) score of 0.01 and prior $P(c)$ of 0.01 to all the unseen contexts. The min and max values of α_c are 0.0 and 1.0 respectively.

Figure 4 shows cross-language phone confusion matrices for the two context-sensitive data driven phone merging approaches. We observe that phone-specific coefficients are able to capture only the most prominent merges while global coefficients produce merges that highly correlate with the manual merge. This might be due to the division of context information across phones which reduces the context sensitivity. This clearly suggests that some contexts help in producing better predictions than others, regardless of the reference phone. Although many of the swaps predicted using DDPM closely resemble the manual merges, the manual merge method outperformed global-DDPM by approx. 3% absolute as seen in Table 3. The distribution of α_c values show that the left and the right context exhibit high confidence scores, whereas low confidence scores were observed when both the context were considered. Hence, we will benefit by removing the spurious low-confidence contexts while merging.

Figure 5 presents the swap likelihood of the predicted phone-pairs in decreasing order. We observe that phone pairs that have the highest swap likelihoods include nasals, close vowels and stops. It is interesting that the data driven method identified new phone merges such as (ae, Aa_HP) and (aa, ow_HP) compared to manual merged phones. We believe that incorporating these new phone merges into the manually merged phone set will improve the ASR performance. Further experiments need to be conducted to verify this claim.

4.6 Probabilistic merging

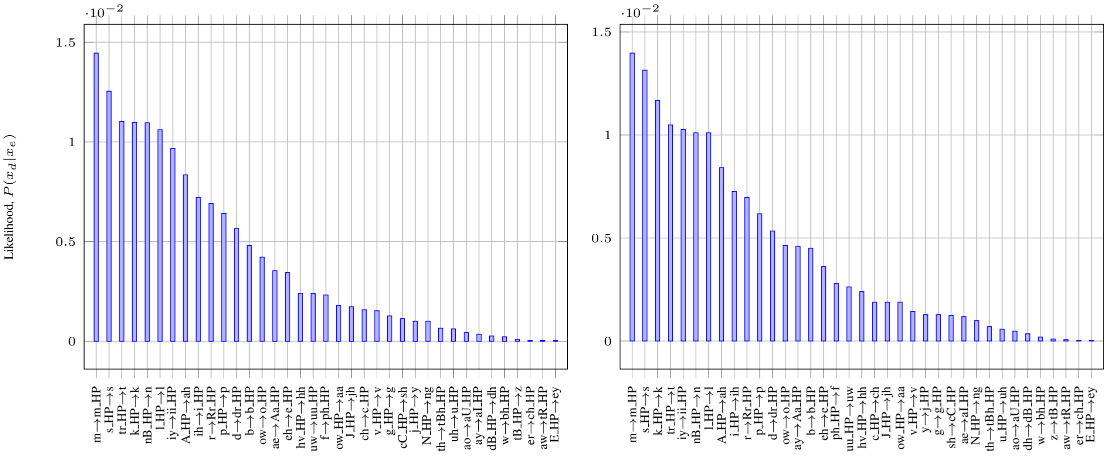
Next, we propose a method to allow the acoustic model to select appropriate phones during decoding. We trained an acoustic model using the merged phones while also retaining the Hindi and English phones. For example, the phone set for the new AM contained the English phone ‘eh’, the Hindi phone ‘E-HP’ as well as the merged phone ‘E-HP-M’. The intuition behind this approach is to let the decoder choose between multiple pronunciation variants in the lexicon so as to determine the pronunciation used by a speaker who code-



(a) Phone-specific

(b) Global

Figure 4: Cross-language phone confusion matrices using phone-specific ($\alpha_c(p)$) and global (α_c) context reliability coefficients



(a) Phone-specific

(b) Global

Figure 5: Swap-pair likelihood for phone-specific ($\alpha_c(p)$) and global (α_c) context reliability coefficients.

switched. This approach has been shown to work well in speech synthesis for pronunciation modeling of homographs (Sitaram et al., 2015). Table 4 compares the number of unique Hindi, English and merged phones for the different systems mentioned.

During training, we modified the lexicon so as to retain part of the data for the unmerged phones and assigned the rest to the merged phones. We assigned 30% and 70% of the data to merged phones which we refer to as P_{30} and P_{70}. During decoding, we created a different lexicon allowing all possible pronunciation variants. We obtained WERs of 41.21% and 40.92% using the HMM-GMM based AM for P_{30} and P_{70} models respectively which are lower than the baseline. WAPI

	En	**Hi**	**Merged**	**Total**
Baseline	39	50	0	89
All merge	8	19	31	58
DDPM	9	20	30	59
DDPM ($\alpha_c(p)$)	20	25	14	59
DDPM (α_c)	21	24	14	59
Probabilistic Merging	39	50	31	120

Table 4: Number of phones before and after merging

score of 1.63 for P_{70} model was higher compared to -0.59 of P_{30} model but lower compared to the all merge model. Table 4 shows the number of English, Hindi and merged phones for each technique.

The WER is a function of number of insertion, deletion and substitution errors as well as the correct token numbers. Figure 6 shows the relative percentage change in the insertion, deletion and substitution values of the “ all merge”, P_{30} and P_{70} model compared to the baseline system using TDNN as AM. The best system should have the lowest insertion, deletion and substitution errors and highest correct tokens. The “all merge” model, which has the best WER scores, has higher insertion and substitution errors but performs better on deletion errors and recognition of correct tokens. We can infer that certain phone merges are causing higher insertion and substitution errors and should be avoided, thus concluding that the manually merged phones are sub-optimal.

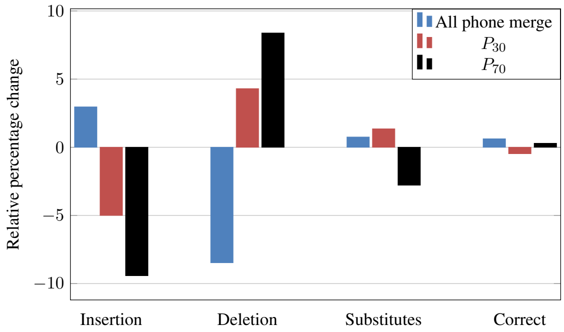


Figure 6: Percentage change in insertion, deletion, substitution and correct token recognition numbers compared to the baseline system with TDNN as AM. The best system should have the lowest of insertion, substitution and deletion numbers and the highest of correct token numbers. The number of word tokens in baseline system for insertion, deletion, substitution and correct tokens are 3342, 5735, 11089 and 49011 respectively.

5 Conclusion

In this work, we compare phone merging techniques in context of code-mixed Hindi-English speech with a baseline system built using a union of both phone sets. We first merge similar sounding phones across both languages manually in order to reduce the phone set size and increase the data availability per phone. We observe a 3% relative improvement in the WER values compared to the baseline using a p-norm based DNN model along with a significant improvement in phone accuracies. We then propose data-driven approaches to merge phones automatically. To correct the errors made by data-driven method, we weight the cross-language swaps using reliable within-language contexts. These methods gave newer phone merge recommendations which can be useful to improve the ASR performance. We further propose probabilistic methods where in the decoder is provided with both the merged as well as the unmerged phones which reduced the insertion errors compared to the manually merged system. these techniques came close to, but did not improve upon the manually merged ASR system. Error analysis of manual merging indicates that it is not optimal and there is a need for better data-driven techniques to automatically merge phones for code-switched ASR.

6 Acknowledgments

We thank the anonymous reviewers for carefully reading our manuscript and offering valuable suggestions.

References

Basem HA Ahmed and Tien-Ping Tan. 2012. Automatic speech recognition of code switching speech using 1-best rescoring. In *Asian Language Processing (IALP), 2012 International Conference on*, pages 137–140. IEEE.

Kiran Bhuvanagirir and Sunil Kumar Kopparapu. 2012. Mixed language speech recognition without explicit identification of language. *American Journal of Signal Processing*, 2(5):92–97.

Joyce YC Chan, Houwei Cao, PC Ching, and Tan Lee. 2009. Automatic recognition of cantonese-english code-mixing speech. *Computational Linguistics and Chinese Language Processing*, 14(3):281–304.

Joyce YC Chan, PC Ching, Tan Lee, and Helen M Meng. 2004. Detection of language boundary in code-switching utterances by bi-phone probabilities. In *Chinese Spoken Language Processing, 2004 International Symposium on*, pages 293–296. IEEE.

Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. 2011. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798.

Mark JF Gales. 1998. Maximum likelihood linear transformations for hmm-based speech recognition. *Computer speech & language*, 12(2):75–98.

Hui Lin, Li Deng, Dong Yu, Yi-fan Gong, Alex Acero, and Chin-Hui Lee. 2009. A study on multilingual acoustic modeling for large vocabulary asr. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 4333–4336. IEEE.

Dau-Cheng Lyu, Ren-Yuan Lyu, Yuang-chin Chiang, and Chun-Nan Hsu. 2006. Speech recognition on code-switching among the chinese dialects. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 1, pages I–I. IEEE.

Tetyana Lyudovyk and Valeriy Pylypenko. 2014. Code-switching speech recognition for closely related languages. *Proc. SLTU*, pages 188–193.

Ayushi Pandey, Brij Mohan Lai Srivastava, and Suryakanth V Gangashetty. 2017. Adapting monolingual resources for code-mixed hindi-english speech recognition. In *Asian Language Processing (IALP), 2017 International Conference on*, pages 218–221. IEEE.

Alok Parlikar, Sunayana Sitaram, Andrew Wilkinson, and Alan W Black. The festvox indic frontend for grapheme to phoneme conversion. In *WILDRE: Workshop on Indian Langauge Data, Resources and Evaluation, 2016*.

Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. 2015. A time delay neural network architecture for efficient modeling of long temporal contexts. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. 2011. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, EPFL-CONF-192584. IEEE Signal Processing Society.

Tanja Schultz and Alex Waibel. 1997. Fast bootstrapping of lvcsr systems with multilingual phoneme sets. In *Fifth European Conference on Speech Communication and Technology*.

Tanja Schultz and Alex Waibel. 2001. Language-independent and language-adaptive acoustic modeling for speech recognition. *Speech Communication*, 35(1-2):31–51.

Sunayana Sitaram and Alan W Black. 2016. Speech synthesis of code-mixed text. In *LREC*.

Sunayana Sitaram, Serena Jeblee, and Alan W Black. 2015. Using acoustics to improve pronunciation for synthesis of low resource languages. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Shubham Toshniwal, Tara N Sainath, Ron J Weiss, Bo Li, Pedro Moreno, Eugene Weinstein, and Kanishka Rao. 2018. Multilingual speech recognition with a single end-to-end model. In *ICASSP*. IEEE.

Ngoc Thang Vu, David Imseng, Daniel Povey, Petr Motlicek, Tanja Schultz, and Hervé Bourlard. 2014. Multilingual deep neural network based acoustic modeling for rapid language adaptation. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 7639–7643. IEEE.

Ngoc Thang Vu, Dau-Cheng Lyu, Jochen Weiner, Dominic Telaar, Tim Schlippe, Fabian Blaicher, Eng-Siong Chng, Tanja Schultz, and Haizhou Li. 2012. A first speech recognition system for mandarin-english code-switch conversational speech. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4889–4892. IEEE.

Robert L Weide. 1998. The cmu pronouncing dictionary. *URL: http://www. speech. cs. cmu. edu/cgibin/cmudict*.

Jochen Weiner. 2012. *Integration of language identification into a recognition system for spoken conversations containing code-switches*. Ph.D. thesis, Language Technologies Institute.

Ching-Feng Yeh and Lin-Shan Lee. 2015. An improved framework for recognizing highly imbalanced bilingual code-switched lectures with cross-language acoustic modeling and frame-level language identification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(7):1144–1159.

Emre Yılmaz, Henk van den Heuvel, and David van Leeuwen. 2016. Investigating bilingual deep neural networks for automatic recognition of code-switching frisian speech. *Procedia Computer Science*, 81:159–166.

Shengmin Yu, Shitwu Zhang, and Bo Xu. 2004. Chinese-english bilingual phone modeling for cross-language speech recognition. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, volume 1, pages I–917. IEEE.

Xiaohui Zhang, Jan Trmal, Daniel Povey, and Sanjeev Khudanpur. 2014. Improving deep neural network acoustic models using generalized maxout networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 215–219. IEEE.