EM-based Phoneme Confusion Matrix Generation for Low-resource Spoken Term Detection

Di Xu, Yun Wang, Florian Metze

Language Technologies Institute, School of Computer Science, Carnegie Mellon University
Pittsburgh, PA, USA

dix@cs.cmu.edu, yunwang@cs.cmu.edu, fmetze@cs.cmu.edu

Abstract

The idea of using a data-driven phoneme confusion matrix (PCM) to enhance speech recognition and retrieval performance is not new to the speech community. Although empirical results show various degrees of improvements brought by introducing a PCM, the underlying data-driven processes introduced in most papers are rather ad-hoc and lack rigorous statistical justifications. In this paper we will focus on the statistical aspects of PCM generation, propose and justify a novel expectation-maximization based algorithm for data-driven PCM generation. We will evaluate the performance of the generated PCMs under the context of low-resource spoken term detection, with primary focus on out-of-vocabulary keywords.

Index Terms— Expectation-maximization algorithm, machine learning, information retrieval, spoken term detection, out-of-vocabulary words

1. Introduction

A confusion matrix is a generic data structure which serves a variety of purposes across different fields. In machine learning, they are commonly known as contingency tables or error matrices that are used to evaluate the performance of a classifier, where the rows are associated with actual classes and the columns are associated with predicated classes. In the speech community, a phoneme confusion matrix (PCM) is used to visualize what types of errors automatic speech recognition (ASR) systems tend to make by confusing one phoneme with another.

There have been some pieces of work in the past decade that addressed how the use of a phoneme confusion matrix might improve the performance in various ASR, spoken document retrieval (SDR) and spoken term detection (STD) tasks.

One of the earliest significant work on using a PCM to enhance SDR performance can be found in [1], in which Srinivasan *et al.*introduced a probabilistic SDR framework based on combined word-based indices and phonetic indices. Phonetic recognition typically has limited accuracy due to the fact that certain groups of phones are likely to be confused even by humans, not to mention by ASR systems. In their work, a PCM was used to address this issue by bridging gaps between falsely recognized phoneme sequences and their ground truths. One major contribution of their work was a formalization of the generation of a probabilistic PCM, which has been inherited in many later works. Another major contribution was defining a way of estimating probabilistic similarities between two phoneme sequences, and also some add-ons to the method which may help achieve better and more realistic estimations. We will refer to the PCM generation method proposed in [1] as the conventional PCM generation method.

Following the work of Srinivasan *et al.*were a couple of PCM based applications, with less focus on the generation of PCMs. A similar approach has been presented in [2], but used only a phoneme based index. The phonetic confusability values stored in the PCM were used to perform document level expansion. This was done by considering proxy terms in a target document that were most confusable with the query term, and using that proxy term for document scoring.

In the works done in [3] and [4], PCMs were used to enhance ASR performance. In [3], a comprehensive comparison was made between an expert-generated and a data-driven PCM. In their work, both an expert-generated PCM and a data-driven PCM were used to generate a phonetic broad class to provide additional knowledge for the speech recognizer. Their data-driven PCM was generated the same way as in [1]. This piece of work also focused on multilingual environments and showed that data-driven phonetic broad classes significantly outperformed expert-generated ones under multilingual environments. On the other hand, the work presented in [4] sought to use PCMs to resolve confusions brought by dysarthric speech. In addition, an ad-hoc way of improving PCM quality was also discussed, which can be regarded as an advancement compared to the standard PCM generation technique used in other papers.

An application of PCMs in STD was studied in [5], where PCMs were generated to resolve the confusions brought by various dialects of Mandarin Chinese. The authors explored a couple of ways of collecting potential confusion pairs for PCM generation, which included the use of 1-best recognition results, and 1-best or even n-best hypotheses in the confusion networks (CNs). They were able to generate a better PCM for their STD tasks by borrowing more information from the CNs. This was expected because in each segment of the phoneme CNs one may find more confusion pairs, thus supporting the PCM generation with more training data, making the PCM less biased. In this work, search of phoneme sequences was done by using sliding windows, while in [6] more advanced weighted edit distance based methods have been investigated, which served as a baseline in that paper for more complex similarity measures.

An overview of previous work on generating and applying PCMs in various tasks indicates that PCMs can improve the performance as long as there are phoneme-level confusions. However, the fundamental PCM generation process has not been the focus in most applications after [1]. We consider the two-pass method proposed in [4] that improves the alignment generated by the dynamic programming algorithm a reasonable way to improve the PCM generation process. Apart from that, there have been virtually no attempts to refine the PCM generation process. Unfortunately the two-pass method failed to rouse enough attention and hasn't been widely discussed.

Our idea came without prior knowledge of the two-pass method proposed in [4], but shares the same motivations. We seek to pursue even further than in [4] and propose a novel PCM generation algorithm based on the expectation-maximization (EM) framework. In the rest of this paper, we will briefly recapitulate the EM framework, elaborate on our EM-based PCM generation algorithm and discuss some of its properties. We will evaluate our PCM generation method with a STD task on 4 low-resources languages.

2. EM-based PCM Generation

2.1. Probabilistic Phoneme Confusion Matrix

We will stick to a probabilistic PCM where each row stands for the truth, i.e. the phoneme in the reference transcription (R), and each column stands for the hypothesized phoneme (H) provided by the decoder, such that each entry stores the conditional probability that hypothesis ph_H is observed given that the reference the corresponding phoneme is ph_R, which is denoted and computed as:

$$P_{ph_R}(ph_H) = P(\underset{\text{hyp}}{ph_H}|\underset{\text{ref}}{ph_R}) = \frac{C_{ph_R}(ph_H)}{C_{ph_R}(.)}, \qquad (1)$$

where $C_{ph_R}(ph_H)$ is the number of times reference ph_R is substituted by hypothesis ph_H, and $C_{ph_R}(.)$ is the number of substitution errors associated with the reference ph_R. We will use the simplified notation $P_{ph_R}(ph_H)$ instead of $P(\underset{\text{hyp}}{ph_H}|\underset{\text{ref}}{ph_R})$. In particular we define $P_{ph_R}(ph_H)$ to be 0, if ph_R never appears in the references. Therefore, the resulting PCM may not be a full ranked matrix.

A subtle and often overlooked fact is that the role of the reference and the hypothesis cannot and should not be switched during the MLE. For example, given the phoneme reference "a b b b" and the hypothesis "a c d c", the alignment carried out by the Viterbi algorithm will regard "c" and "d" as substitution errors of "b", but not the other way around. Therefore $P_b(c) = 2/3$, $P_b(d) = 1/3$, and of course they sum to 1, but $P_c(b) = 0$ and $P_d(b) = 0$, based on this training alignment. Apparently, as long as we maintain the reference-hypothesis relation, the quantity $C_{ph_R}(ph_H)$ is directional. In some previous work, this quantity was mistakenly made symmetrical.

2.2. Expectation-maximization Algorithm

The expectation-maximization (EM) algorithm has been thoroughly studied and justified in the late 20th century [7] [8] [9], and has many well-known applications such as estimating the parameters of a Gaussian Mixture Model for audio signal classification [10], the Baum-Welch algorithm for learning a Hidden Markov Model [11], and finding the optimal linear interpolation weights for hierarchical language models [12]. It is also worth mentioning that most applications of EM have used maximum-likelihood estimations (MLE) and are therefore frequentist, although there is also a Bayesian version of the EM algorithm which performs maximum *a posteriori* (MAP) estimations.

In general, the EM algorithm can be applied whenever there is incomplete data that prevents learning (non-hidden) model parameters in a mathematically tractable manner. It is important to point out that the EM algorithm itself is a generic framework for parameter estimation, and its optimization criterion is application specific. The notion of "incomplete data" is used to imply that the observable statistics are generated by an underlying hidden process controlled by a set of hidden variables.

It is handy to define the observable data as a random variable X, whose distribution is governed by an underlying process; and define the unknown underlying process as another random variable Y; thus we have the complete data denoted by $Z = (X, Y)$. We use corresponding lower case letters to denote instances of each random variable. In addition, we denote by θ the set of model parameters to be estimated. Thus we have the log-likelihood of the complete data based on the parameters:

$$L(\theta|Z) = \log p(Z|\theta) = \log p(X, Y|\theta). \qquad (2)$$

As an iterative algorithm, EM consists of two steps in each iteration. In each iteration, we optimize the following auxiliary function:

$$Q(\theta, \theta^{(t-1)}) = E_{Y|X,\theta^{(t-1)}} \log p(X, Y|\theta) \qquad (3)$$

where $\theta^{(t-1)}$ is the parameters obtained before this iteration, and θ is the parameters that will be obtained after this iteration. The expectation (E) step finds the expectation of 2 over posterior distribution of the hidden variable Y given the observable X and the fixed $\theta^{(t-1)}$, while the ensuing maximization (M) step finds the optimal θ using MLE. Each iteration can be encapsulated by the formula:

$$\theta^{(t)} = \underset{\theta}{\operatorname{argmax}} \left\{ E_{Y|X,\theta^{(t-1)}} \log p(X, Y|\theta) \right\}. \qquad (4)$$

As has been proved in many previous works, each iteration is guaranteed to improve the object function until convergence.

2.3. Hard EM

Before we can unveil how our PCM generation algorithm fits a general EM framework, it is important to address a variant of the basic EM algorithm, known as the "hard EM" which differs from the commonly used (soft) EM algorithms in the E step.

The conceptual difference between soft and hard EM has been introduced by Segal *et al.*in the book [13]. In soft EM, the E step accounts for the probability over all hidden variables, while in hard EM, the single most likely assignment of the hidden variable is selected. Effectively, the hard EM can be encapsulated by the formula:

$$\theta^{(t)} = \underset{\theta}{\operatorname{argmax}} \left\{ \max_{Y} [\log p(X, Y|\theta)|X, \theta^{t-1}] \right\}. \qquad (5)$$

Moreover, it has also been proved in [13] that the hard EM is also guaranteed to converge to a local optimum.

The adoption of the EM framework is motivated by the fact that there exists at least one alignment that optimally reflects how the ASR performs on the speech data, but that optimal alignment can not be obtained with out having a corresponding optimal PCM that we aim to learn, and vice versa. We only have the references and the hypotheses, without knowing which alignment is optimal among the exponentially many ones in the first place. Instead of exhausting the search space, which is infeasible, we can learn a rough PCM from the first round Viterbi alignments. The rough PCM, though imperfect, can shrink our search space effectively and allow us to use the Viterbi algorithm to find a better set of phoneme alignments, which will in turn guide us to a better PCM.

To formulate our method in a hard EM framework, we define our optimization criterion as the mean of the average alignment cost of each sentence over the entire speech data. The average alignment cost is obtained from the Viterbi algorithm, which is used to find the optimal alignment between a reference phoneme sequence and a hypothesized phoneme sequence. Since the optimization criterion is disjoint across sentences, the minimum overall cost can be obtained by summing up the costs of the optimal alignments for each sentence. Moreover, the MLE guarantees that each row of the PCM must sum up to one, which serves as a natural constraint for the optimization criterion.

We define the observed data as a set of reference-hypothesis (ref-hyp) phoneme sequence pairs. This is obtained by translating the word-based ground truth of the speech data and the word-based hypotheses generated by the ASR into phoneme sequences using a grapheme-to-phoneme (G2P) model [14].

Our hidden data is defined as all the unobserved possible alignments for all the reference-hypothesis phoneme sequence pairs. There are a finite number of but exponentially many possible alignments for each ref-hyp phoneme sequence pair. Clearly, enumerating through all of them and to obtain an expected cost is not scalable and even silly because the absolute majority of those enumerations have extremely low probabilities. This explains why we choose hard EM over soft EM: in soft EM one needs to find a weighted sum of the average alignment costs of all possible phoneme alignments, while in hard EM we only deal with the most probable alignment.

2.4. Expectation Step

Under a hard EM framework, the E step is effectively carried out by the Viterbi algorithm, with the cost of each phoneme-to-phoneme alignment given by the PCM obtained after each iteration. In particular, the Viterbi algorithm automatically applies the $\max$ operator in Eq. (5). This has also been discussed in [15] and [16] where the Viterbi algorithm was used. Notice that the $\max$ operator is a general notion; in our case it is implemented with a $\min$ operator because our optimization criterion is a cost function.

To elaborate on how Viterbi implements hard EM from a microscopic perspective, consider only one pair of ref-hyp phoneme sequences. There may be multiple unobserved alignments upon a substitution error, but the Viterbi algorithm will calculate the most likely alignment by preferring a particular alignment over others as the PCM provides numeric estimations of the probabilities that the ASR system will confuse a phoneme with other phonemes.

Upon each iteration, the PCM will be updated with a new set of values, which can be considered as the model parameters θ in the general EM framework. It is important to realize that our optimization space is not convex, and there might be multiple optimal PCMs for a particular set of ref-hyp phoneme sequences. For example, consider the reference sequence "A X D" and the hypothesis "A B C D". The optimal PCM learned may either assign $P_x(B) = 1$ or $P_X(C) = 1$. In other words, as long as the Viterbi algorithm encounters ties, the optimization surface may not be convex as the ties introduces indeterminacy. However, for larger training sets, this will not prevent learning a PCM that is reasonable enough since ties will never be dominant enough to affect the relative frequencies. For phones that appear rarely, however, this could be an issue. To resolve it, one needs to define rules that always break such ties in a consistent manner.

2.5. Maximization Step

The M step is rather straight forward both conceptually and computationally. We estimate the probabilities using MLE, and the calculation has already been illustrated in Eq. (1).

3. Experiments

3.1. Data and System Descriptions

We will experiment with our generated PCMs on four low-resource languages: Assamese, Bengali, Haitian and Lao. The STD tasks are powered by Probabilistic Phonetic Retrieval (PPR), which we will introduce in section 3.2. Our primary evaluation metric is Actual Term Weighted Value (ATWV) [17] on OOV queries, as the scoring for OOV queries particularly requires better phonetic similarity estimations. For all the detections for a particular query, Expected Count Thresholding (ECT) proposed in [18] is used to compute a dynamic threshold deciding which ones are to be scored. This method is optimal if the scores of the items in the posting list accurately reflect the probability of relevance. Since the purpose of using better PCMs is to provide better probabilistic estimations of phonetic similarity and resolve systematic biases, using PCM on PPR should improve ATWV. In addition, to prevent over-fitting, we will use two independent sets of speech data, decoded by the same recognizer.

Our datasets are provided by the IARPA Babel Program [19], and we will work under the Limited Language Pack (LimitedLP) condition defined in [17], where OOV words were a prominent issue. For each language, we had 10 hours of development (dev) data and a small part (evalpart1, 5 hours) of the 70-hour evaluation data. Table 1 presents the datasets and the word error rate (WER) of the decoding results, upon which PPR was used to run keyword search with various PCMs.

Language	Dataset ID (LimitedLP)	WER (%)
Assamese	IARPA-babel102b-v0.5a_conv-dev	60.3
	IARPA-babel102b-v0.5a_conv-evalpart1	58.7
Bengali	IARPA-babel103b-v0.4b_conv-dev	63.0
	IARPA-babel103b-v0.4b_conv-evalpart1	61.3
Haitian	IARPA-babel201b-v0.2b_conv-dev	59.1
	IARPA-babel201b-v0.2b_conv-evalpart1	57.7
Lao	IARPA-babel203b-v3.1a_conv-dev	60.5
	IARPA-babel203b-v3.1a_conv-evalpart1	56.6

Table 1: Data and ASR Performance

For each dataset, a massive word confusion network (WCN) [20] was generated with our state-of-the-art ASR system introduced in [21]. Therefore the quality of the WCNs was sufficient for PPR to reflect changes in the retrieval system.

3.2. Probabilistic Phonetic Retrieval (PPR)

We will evaluate our PCM generation in a STD task on low resource languages where out-of-vocabulary (OOV) words are the prominent issue. The STD task is formally defined by NIST in the OpenKWS14 Evaluation Plan [17]. Our underlying speech-to-text system, and the STD system based on Probabilistic Phonetic Retrieval (PPR) have been introduced in [21].

PPR was proposed to provide OOV handling capabilities for word-based STD tasks, and has proved to be effective on four low-resource languages in [21]. The idea is to consider fuzzy matches during the search, and reward fuzzy matches

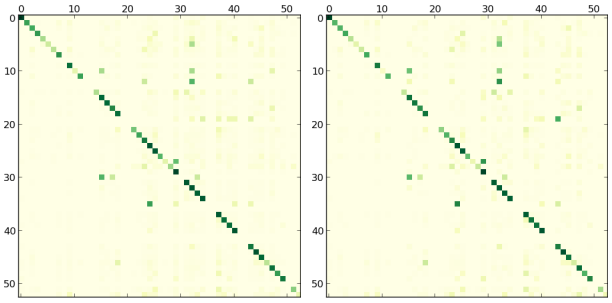

Figure 1: Two PCMs for Assamese. The left panel shows the PCM obtained after one iteration of the EM algorithm; the right panel shows the PCM obtained after the algorithm has converged.

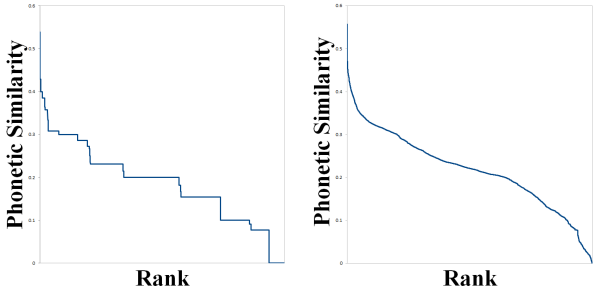

Figure 2: Ranked phonetic similarities for 5000 randomly sampled word pairs. Using a PCM (right) results in a much smoother curve than if using binary similarities (left).

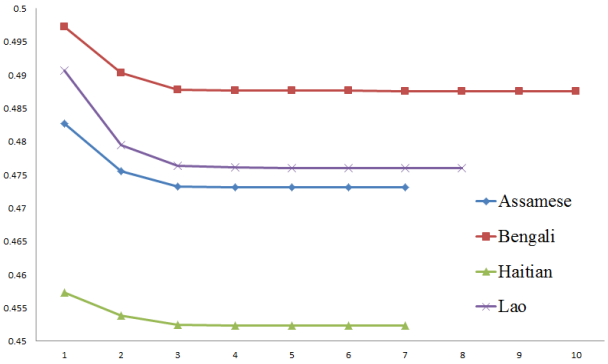

Figure 3: The optimization criterion (ordinate) after each iteration of EM (abscissa) for Assamese, Bengali, Haitian and Lao.

with high phonetic similarities to the target. The phonetic similarity used in [21] was the Levenshtein distance between the phoneme sequences of two words. However, this method considers only binary similarities on the phoneme level, which is equivalent of using a PCM with all diagonal entries being 1 and other entries being 0. This makes the phonetic similarity calculation systematically biased, leading to less accurate probabilistic relevance estimations. A better PCM is needed to improve the phonetic similarity estimation of PPR, and thus improve the overall retrieval performance.

We will show that our PCMs perform better than the conventional PCMs in terms of improving the OOV scoring quality of PPR. We will also illustrate that the hard EM based algorithm enhances the quality of the PCM upon the first couple of iterations, but this is followed by a saturation effect. We propose ways to avoid over-fitting.

3.3. Probabilistic Phonetic Similarity for Word-based Hypotheses

The word-to-word phonetic similarity calculation used in PPR was defined as the phoneme error rate of the hypothesis compared to the reference. To leverage the information provided by the PCM, we redefine this similarity as the likelihood that the target word is recognized as the given word hypothesis. We further assume that this quantity is consistent with the average likelihood of the phoneme hypotheses defined in formula 1. In particular, this is calculated by taking the geometric mean of the step costs used by the Viterbi algorithm when finding the optimal alignment between the two phoneme sequences. It is necessary to take the average likelihood as the joint likelihood is not directly comparable for hypotheses of different lengths.

Recall that in a probabilistic PCM, each row represents the probability distribution on how a reference phoneme may be correctly recognized or incorrectly recognized as other phonemes. This makes practical sense because our goal is to learn a PCM that best reflects how the ASR works on a particular set of speech data, rather than finding a PCM that reflects common sense. This is also one of the major differences between a data-driven matrix and expert-generated matrix. In Fig. 1, we can observe that a data driven PCM may distribute more probability mass to off-diagonal entries than diagonal ones if the ASR systematically mis-recognizes a phoneme as another. This also makes the phonetic similarity for exact word matches less than 1, since there is some probability that the word is mis-recognized as something else. As mentioned before, the use of PCM provides smoother and better approximations of phonetic similarity. This can be observed from Fig. 2.

3.4. Observations on EM-based PCM Generation

What we have observed from the iterations is typical for an EM algorithm. Table 3 shows the value of the optimization criterion after each iteration before convergence for the four experimented languages. We can observe that the number of iterations required to converge differs from one language to another. Bengali takes more iterations to converge, which is likely due to the fact that it has more phonemes than the other three languages. It is also noticeable that for all four languages, the optimization criterion does not change significantly after the third iteration. This observation reminds us that there may be a potential of over-fitting for PCMs generated after the third iteration of the EM algorithm. However, a more justified way of validation is to evaluate the PCM on a validation data and choose the best number of iterations, and then test it on a different dataset.

3.5. Baselines

Our primary baseline system was the original PPR system which does not require a PCM, labelled “PPR-original”. It can be regarded as using a PCM with all diagonal entries being 1 and all others being 0. In order to show that the EM-based PCM generation algorithm finds better PCMs after each EM iteration until convergence, we took the PPR system with a PCM generated after the first iteration of the EM algorithm as a secondary baseline, labeled “PPR-PCM-1st”. The PCM after the first iteration is equivalent to the PCM generated with the conventional

data-driven method originally purposed in [1], starting with binary phoneme-to-phoneme similarities.

We will first show that the secondary baseline outperforms the primary baseline on OOV queries, which proves that it is reasonable to apply EM-generated PCMs to a PPR powered STD system. We will then show that our EM-based algorithm is capable of improving the quality of the PCMs further from the conventional method.

3.6. ATWV Improvements with EM-Generated PCMs

For validation purposes, we ran PPR with and without EM generated PCMs on the dev dataset for the 4 languages. The results are listed in Table 2, in which the OOV ATWV for each system is presented first, followed by their relative improvement from the primary baseline (PPR-original). Apparently the secondary baseline (PPR-PCM-1st) outperforms the primary baseline, indicating that it is a valid approach to evaluate the quality of a PCM using PPR. It is obvious that the best ATWVs are achieved mostly in the second iteration of the PCM algorithm, suggesting that our EM-based PCM generation is able to find better PCMs than the conventional method (PPR-PCM-1st). It is also worth mentioning that the effect of over-fitting is present. Notice that for Haitian, the PCM obtained upon convergence is not better than the one obtained after the first iteration. This is expected because we didn't assume the training data to be perfect, and thus the EM algorithm was likely to learn noises in the data.

	Assamese	Bengali	Haitian	Lao
PPR-original	0.022363	0.033048	0.02727	0.010701
PPR-PCM-1st	0.025052	0.034658	0.03015	0.011024
	+11.9%	+4.9%	+10.6%	+3.0%
PPR-PCM-2nd	0.027737	0.035194	0.03241	0.015402
	+24.0%	**+6.5%**	**+18.8%**	**+43.9%**
PPR-PCM-3rd	0.026578	0.035060	0.03241	0.015008
	+18.8%	+6.1%	+18.8%	+40.2%
PPR-PCM-4th	0.025896	0.035060	0.03050	0.015008
	+15.8%	+6.1%	+11.8%	+40.2%
PPR-PCM-5th	0.025579	0.034926	0.00305	0.014979
	+14.4%	+5.7%	+11.8%	+39.9%
PPR-PCM-6th	0.025570	0.034926	0.03012	0.014979
	+14.4%	+5.7%	+10.4%	+39.9%

Table 2: Validation results over EM iterations based on ATWVs

Based on the validation results, we tested the selected PCMs generated from the 2nd iteration of the EM algorithm on the evalpart1 dataset, and we also present the results obtained with other PCMs to see if the validation process was effective. The results are presented in Table 3. Apart from that, we have also conducted the Wilcoxon signed-rank test [22] and reported the p-values in Table 4 to check if the selected PPR-PCM-2nd performs significantly better than the baselines.

The testing results pretty much agreed with what the validation outcomes suggested, and we can conclude that for the four languages and the underlying systems, the PCMs generated from the 2nd and 3rd iteration of the EM-based algorithm provide noticeable improvements over the both the primary and secondary baseline. In particular, according to the Wilcoxon signed-rank test, we have enough evidence to claim that on the given data PPR-PCM-2nd is significantly better than the primary baseline; we have enough evidence to claim that it is also significantly better than the secondary baseline on Assamese and Lao, and some evidence to say the same for Bengali and Haitian. Apart from that, we can also observe that without validation, the potential of over-fitting is still present and is likely to occur after the 5th iteration.

	Assamese	Bengali	Haitian	Lao
PPR-original	0.043426	0.052504	0.006306	0.016834
PPR-PCM-1st	0.049241	0.055882	0.007273	0.020139
	+13.4%	+6.4%	+15.3%	+19.6%
PPR-PCM-2nd	0.053744	0.058999	0.007660	0.022166
	+23.8%	+12.4%	**+21.5%**	+31.7%
PPR-PCM-3rd	0.049453	0.059078	0.007273	0.022233
	+14.2%	**+12.5%**	+15.3%	**+32.0%**
PPR-PCM-4th	0.049421	0.058808	0.007273	0.022033
	+13.8%	+12.0%	+15.3%	+30.8%
PPR-PCM-5th	0.049234	0.058514	0.007079	0.021793
	+13.4%	+11.4%	+12.3%	+29.5%
PPR-PCM-6th	0.049008	0.058011	0.007079	0.021793
	+12.8%	+10.5%	+12.3%	+29.5%

Table 3: Testing results for the selected PCMs (PPR-PCM-2nd) on evalpart1

P-value	PPR-PCM-2nd			
	Assamese	Bengali	Haitian	Lao
PPR-original	**0.011**	**0.048**	**0.021**	**0.013**
PPR-PCM-1st	**0.032**	0.087	0.068	**0.044**

Table 4: Wilcoxon Signed Rank test on PPR-PCM-2nd against the baselines

4. Conclusions

In this paper we have reviewed the EM algorithm and one of its variants, the hard EM. Based on that, we have introduced a novel data-driven PCM generation algorithm, demonstrated why this method fits a general (hard) EM framework with statistical elaborations and observations on the saturation pattern of the algorithm and the resulting PCMs. We have rigorously evaluated our method by applying the generated PCMs in a challenging STD task on low-resource languages to improve the OOV ATWV, and the results confirm our hypothesis that the EM-based algorithm is capable of generating better PCMs than the conventional method. Apart from that, we have also shown that our method, like many others, requires validation to prevent over-fitting. We consider our major contribution in this paper to be the formulation of the PCM generation method in an EM framework, thus expanding the applications of machine learning techniques in spoken language technologies.

5. Acknowledgements

This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense U.S. Army Research Laboratory (DoD/ARL) contract number W911NF-12-C-0015. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government.

6. References

[1] S. Srinivasan and D. Petkovic, "Phonetic confusion matrix based spoken document retrieval," in *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2000, pp. 81–87.

[2] N. Moreau, H.-G. Kim, and T. Sikora, "Phonetic confusion based document expansion for spoken document retrieval." in *INTERSPEECH*, 2004.

[3] A. Žgank, B. Horvat, and Z. Kačič, "Data-driven generation of phonetic broad classes, based on phoneme confusion matrix similarity," *Speech Communication*, vol. 47, no. 3, pp. 379–393, 2005.

[4] S. O. C. Morales and S. J. Cox, "Modelling confusion matrices to improve speech recognition accuracy, with an application to dysarthric speech." in *INTERSPEECH*, 2007, pp. 1565–1568.

[5] P. Zhang, J. Shao, J. Han, Z. Liu, and Y. Yan, "Keyword spotting based on phoneme confusion matrix," in *Proc. of ISCSLP*, vol. 2, 2006, pp. 408–419.

[6] U. V. Chaudhari and M. Picheny, "Matching criteria for vocabulary-independent search," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 5, pp. 1633–1643, 2012.

[7] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–38, 1977.

[8] T. K. Moon, "The expectation-maximization algorithm," *Signal processing magazine, IEEE*, vol. 13, no. 6, pp. 47–60, 1996.

[9] C. M. Bishop *et al.*, *Pattern recognition and machine learning*. springer New York, 2006, vol. 1.

[10] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the em algorithm," *SIAM review*, vol. 26, no. 2, pp. 195–239, 1984.

[11] L. Rabiner and B.-H. Juang, "An introduction to hidden markov models," *ASSP Magazine, IEEE*, vol. 3, no. 1, pp. 4–16, 1986.

[12] L. Si, R. Jin, J. Callan, and P. Ogilvie, "A language modeling framework for resource selection and results merging," in *Proceedings of the eleventh international conference on Information and knowledge management*. ACM, 2002, pp. 391–397.

[13] E. Segal, A. Battle, and D. Koller, "Decomposing gene expression into cellular processes," in *Proceedings of the Pacific Symposium on Biocomputing*, vol. 8, 2003, pp. 89–100.

[14] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434–451, 2008.

[15] V. I. Spitkovsky, H. Alshawi, D. Jurafsky, and C. D. Manning, "Viterbi training improves unsupervised dependency parsing," in *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 2010, pp. 9–17.

[16] A. Hofleitner and A. Bayen, "Optimal decomposition of travel times measured by probe vehicles using a statistical traffic flow model," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. IEEE, 2011, pp. 815–821.

[17] NIST, "Spoken term detection (std) 2014 evaluation plan," http://www.nist.gov/itl/iad/mig/openkws14.cfm, 2014.

[18] D. R. Miller, M. Kleber, C.-L. Kao, O. Kimball, T. Colthurst, S. A. Lowe, R. M. Schwartz, and H. Gish, "Rapid and accurate spoken term detection." in *INTERSPEECH*, 2007, pp. 314–317.

[19] M. Harper, "Iarpa solicitation iarpa-baa-11-02," *IARPA BAA*, 2011.

[20] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech & Language*, vol. 14, no. 4, pp. 373–400, 2000.

[21] D. Xu and F. Metze, "Word-based probabilistic phonetic retrieval for low-resource spoken term detection," in *15th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2014.

[22] R. H. Randles, "Wilcoxon signed rank test," *Encyclopedia of statistical sciences*, 1988.