Discriminative training of a phoneme confusion model for a dynamic lexicon in ASR

Penny Karanasou, François Yvon, Thomas Lavergne, Lori Lamel

Discriminative training of a phoneme confusion model for a dynamic lexicon in ASR

Penny Karanasou[1,2], *François Yvon*[1,2], *Thomas Lavergne*[1,2], *Lori Lamel*[1]

[1]LIMSI/CNRS, B.P. 133, 91403 Orsay, France
[2]Université Paris-Sud, 91403 Orsay, France
{pkaran, yvon, lavergne, lamel}@limsi.fr

10.21437/Interspeech.2013-468

Abstract

To enhance the recognition lexicon, it is important to be able to add pronunciation variants while keeping the confusability introduced by the extra phonemic variation low. However, this confusability is not easily correlated with the ASR performance, as it is an inherent phenomenon of speech. This paper proposes a method to construct a multiple pronunciation lexicon with a high discriminability. To do so, a phoneme confusion model is used to expand the phonemic search space of pronunciation variants during ASR decoding and a discriminative framework is adopted for the training of the weights of the phoneme confusions. For the parameter estimation, two training algorithms are implemented, the perceptron and the CRF model, using finite state transducers. Experiments on English data were conducted using a large state-of-the-art ASR system of continuous speech.
.

Index Terms: FST-based ASR decoding, dynamic recognition lexicon, phoneme confusion model, discriminative training

1. Introduction

Modern ASR systems rely on a set of parameterized knowledge sources, often taking the form of statistical models estimated on appropriate training data. Amongst those knowledge sources, the pronunciation dictionary has a peculiar role, and its optimization for a specific task is not so easily performed; yet many studies have found that integrating pronunciation variants in a lexicon without optimizing their weights could severely degrade the ASR performance. This explains the growing interest in the design of dynamic, speech-dependent lexica, and of subsequent weight training procedures. One such approach is to label existing speech training data at the phonemic level (for ex. using a phoneme recognizer), then to align these realizations with baseform pronunciations, thereby identifying new variants for known words. These methods are a priori limited to words present in the training set. To circumvent this limitation, it is possible to extract phonological rules once the alignment is done. These rules are not the result of linguistic knowledge as the ones used in knowledge-based approaches. They just adapt the baseform pronunciations to a transcription that better matches the spoken utterance. Some instanciations of this generic methodology are described in [1, 2, 3, 4, 5] to name a few.

Once these surface pronunciations or phonological rules are chosen, the next step is to add weights to them. A basic method is to extract pronunciation probabilities based on the frequency counts of each word [6]. Such weights can only be defined for words occurring in the training set and no further training of the weights is performed. Another method proposed in [7] and [8] is to use the EM algorithm to train these lexical probabilities, a method that seems prone to over-fitting the training data. This explains why the last years there is a turn towards discriminative models. In [9], a maximum entropy model is used to evaluate the pronunciation weights, and in [10] a minimum-classification-error approach is used. The drawback is that such methods are often computationally expensive and, are thus often applied to small data sets. Moreover, the latter works are once again limited to words present in the training set.

In this work, we develop a discriminative framework for training the weights of the pronunciation model and we evaluate the proposed method in a real-world task with experiments on large data sets. First, the output of a phoneme recognizer is aligned with the reference and a set of phoneme confusion pairs is extracted. These confusion pairs are used to expand the phonemic search space of pronunciations during the ASR decoding. In this way, we hope to have pronunciations that better reflect the actual spoken utterances. To train their weights, a discriminative training is performed so as to minimize the phoneme edit distance between the output of the phoneme recognizer and the reference. Two training criteria are implemented, the perceptron and the CRF model. The advantage of using a discriminative model is that the parameters of the model are adapted to minimize the recognition error rate. By contrast, the parameters of a maximum likelihood model are derived, as the name suggests, to maximize the likelihood of some data given the model; increases of the likelihood of the training data, however, do not always translate into reduced error rates.

Another way of seeing the application of our confusion model is as a corrector of the errors of the phoneme recognizer. The study of [11] has shown that phonetic and word errors are correlated, a fact that justifies our choice of an objective function in the phoneme level. This allows us to add variants to the baseform pronunciations of any word and not be limited to words that occur in the training data. Note also that in this way we do not add a fixed number of pronunciations per word, as is done with static g2p conversion.

2. Modelisation of the problem

In this section, we present our model, using concepts and notations borrowed in the structured learning litterature [12] and in the theory of Finite-State Transducers (see [13] for a review).

We first assume the availability of a phoneme recognizer (described in Section 4), turning input speech segments into phonemic lattices (aka weighted acyclic automata) Ph. Lattices contain acoustic scores, which are used during the training of the pronunciation weights, enabling us to integrate some phonemic information provided by the acoustic model. This can improve the results, as observed in [6, 14].

In this work, a unigram model of phonemic confusions (substitutions and deletions) is used, represented by an FST $C(\theta)$, where the vector θ has one component for each possible confusion. $C(\theta)$ is an one-state FST whose structure is designed as follows. We first run a forced alignment of the training data with the reference, yielding phonemic references; we then align the one-best outputs of the phoneme recognizer with the corresponding reference sequences and count the number of phoneme deletions and substitutions. Confusions that appear less than 20 times are not kept to avoid learning hazardous mistakes. The resulting FST contains $1,021$ confusion pairs, for which weights have to be trained. Each confusion is represented by one arc in the FST, whose input (resp. output) symbol corresponds to the recognized (resp. reference) symbol. No particular initialization of the weights θ is performed, since the training algorithms to be used maximize a convex objective function (see details below).

We assume a training set consisting of n examples $\{\langle x^{(i)}, y^{(i)}\rangle\}_{i=1}^{n}$, where $x^{(i)}$ is a phoneme lattice Ph and $y^{(i)}$ is the reference corresponding to the true phoneme sequence. The phoneme lattice $x^{(i)}$ can be expanded with the use of the confusion model through formal composition $Ph \circ C(\theta)$. Let $\mathcal{Y}(x^{(i)})$ be the set of phoneme sequences of the expanded phoneme lattice. Each path π in $\mathcal{Y}(x^{(i)})$ corresponds to an alignment (x, y) of a recognized phoneme sequence x and a possible correction y; the weight of a path is the sum over all arcs of (i) the acoustic score of x, and (ii) the parameters associated with the confusions along the sequences (x, y). The phoneme decoding problem requires solving

$$y^* = \arg\max_{y' \in \mathcal{Y}(x)} \theta^\top f(x, y'). \tag{1}$$

Decoding thus amounts to choosing the minimum-scoring path in the FST representing $\mathcal{Y}(x)$. By changing the weights θ, we also change the path weights and, thereby changes the best path in this FST. Discriminative training aims at changing the weights θ in such a way that the best path gets closer to the reference phoneme sequence. This approach can be viewed as a discriminative "reranking" of the pairs (x, y) that occur in $\mathcal{Y}(x)$.

3. Training criteria

We review two criteria for training the parameter vector θ corresponding to two popular models: first Conditional Random Fields, then the Perceptron model, using the notations of [15].

3.1. The CRF model

To derive the first training criterion, we can use the conditional log-linear model of Equation 2. Note that, in addition to the weights θ of the confusion model, we need to take the acoustic scores a_x for sequence x into account. These scores are independent of θ and appear as an additive factor. Since they do not depend on θ, they do not contribute to the derivatives as we will see below and, therefore do not complicate the optimization.

$$p_\theta(y|x) = \frac{\exp\{\theta^\top f(x, y) + a_x\}}{\sum_{y' \in \mathcal{Y}(x)} \exp\{\theta^\top f(x, y') + a_x\}} \tag{2}$$

The weights θ can then be trained by maximizing the conditional log-likelihood

$$\max_\theta \sum_{i=1}^{n} \big[\theta^\top f(x^{(i)}, y^{(i)}) + a_{x^{(i)}} - log \sum_{y \in \mathcal{Y}(x^{(i)})} \exp\{A_i\}\big], \tag{3}$$

where $A_i = \theta^\top f(x^{(i)}, y) + a_{x^{(i)}}$. Note that for the time being, no regularization term is used in the CRF model. Later, we plan to experiment on using L_2 and L_1 regularizations (see Section 6).

The CRF training criterion, originally proposed by [16], is equivalent to MMI training that is traditionally used in speech recognition to discriminatively train the acoustic model's weights [17]. Note that this framework is pretty generic and can be extended to more complex representations of (x, y), taking for instance the input or output context into account (subject to decomposability constraints, so as to keep the optimization tractable). We leave these extensions for future work.

3.2. Perceptron

The perceptron can be seen as an approximation of the online version of the CRF training criterion, where we equal the posterior probability of the most likely hypothesis to one and of all the other hypotheses to zero. The perceptron algorithm iteratively updates weights by considering each training example in turn. On each round, it uses the current model to make a prediction. If the prediction is correct, there is no change to the weights. If the prediction is incorrect, the weights are updated proportionally to the difference between the correct feature vector $f(x^{(i)}, y^{(i)})$ and the predicted feature vector $f(x^{(i)}, y^*)$. Following the presentation of [12], the weight update for each training example is:

$$\theta \leftarrow \theta + a\big(f(x^{(i)}, y^{(i)}) - f(x^{(i)}, y^*)\big), \tag{4}$$

where a is the learning rate.

It can be shown that this approach minimizes the following loss function, which approximates the zero-one loss:

$$\frac{1}{n} \sum_{i=1}^{n} \theta^\top \big(f(x^{(i)}, y^{(i)}) - f(x^{(i)}, y^*)\big), \tag{5}$$

Following common practice, we use the average version of the perceptron: denoting $\theta_t^{(i)}$ the parameter vector after the ith example is processed on the t^{th} pass through the data. Then the averaged parameters are defined as $\theta_{AVG} = \sum_{i,t} \theta_t^{(i)}/nT$, where n is the number of examples in the training set and T is the number of passes. The *averaged perceptron*, originally proposed by [18], has been shown to give substantial improvements in accuracy over the non averaged version for tagging tasks [19].

3.3. Optimization algorithms

For the perceptron, its built-in update formula is used as already mentioned. For the CRF model a gradient descent with learning rate a can be used as an optimization algorithm. The derivatives that need to be calculated are:

$$\begin{aligned} \frac{\partial CRF(\theta)}{\partial \theta_j} &= \sum_{i=1}^{n} \big[f_j(x^{(i)}, y^{(i)}) - \sum_{y \in \mathcal{Y}(x^{(i)})} f_j(x^{(i)}, y) p_\theta(y|x^{(i)})\big] \\ &= \sum_{i=1}^{n} \big[f_j(x^{(i)}, y^{(i)}) - \mathbb{E}_{p_\theta(y|x^{(i)})}[f_j(x^{(i)}, y)]\big] \end{aligned} \tag{6}$$

The feature expectation $\mathbb{E}_{p_\theta(y|x^{(i)})}[f_j(x^{(i)}, y)]$ is the averaged value of the feature f_j across all $y \in \mathcal{Y}(x^{(i)})$, with each y weighted by its conditional probability given $x^{(i)}$. Using the

log-linear form of the model (Equation (2)), the expectation equates:

$$\mathbb{E}_{p_\theta(y|x^{(i)})}[f_j(x^{(i)}, y)] = \frac{\sum_{y \in \mathcal{Y}(x^{(i)})} f_j(x^{(i)}, y) \exp\{A_i\}}{Z_{x^{(i)}}},$$

where $Z_{x^{(i)}} = \sum_{y' \in \mathcal{Y}(x^{(i)})} \exp\{\theta^\top f(x^{(i)}, y') + a_{x^{(i)}}\}$ is the normalization term, independent of y. The expectation is calculated using the standard forward-backward algorithm.

An additional comment regarding CRF training is in order: until now we presented a simple supervised learning setup where learning is done with gradient descent. However, in this work online training is chosen and stochastic gradient descent is used. Meaning that each iteration estimates this gradient on the basis of a single randomly selected example [20]. In the perceptron case, the stochastic gradient descent matches the original algorithm.

In online training, it has been found that is is better not to use a fixed learning rate a. Instead, learning rates are generally decreased according a schedule of the form $a = a_0/(1+a_0*t)$, where $t = 1, 2, ...n$ is the iteration of the learning algorithm (the example we are processing). This schedule was originally proposed by [21]. It is a gradually decaying learning rate, but smoother than $1/t$. The initial rate a_0 was heuristically set to $a_0 = 0.1$.

4. Experimental set-up

The phoneme recognizer used in these experiments is built using acoustic models that are tied-state, left-to-right 3-state HMMs with Gaussian mixture observation densities. The acoustic models are word position independent, gender-dependent, speaker-adapted, and Maximum Likelihood trained on about 500 hours of audio data. They cover about $30k$ phone contexts with a total of $11,500$ tied states. Unsupervised acoustic model adaptation is performed for each segment cluster using the CMLLR and MLLR techniques prior to decoding. A phonemic 3-gram language model is used in the construction of the phoneme recognizer to impose some constraints in the generated phonemic sequences.

Discriminative training is done on $40h$ of broadcast news (BN) data used for the Quaero projet (www.quaero.org). These data include around $5k$ phoneme lattices. Lattices with very high error rate were removed and the remaining $\sim 4k$ lattices were used for training. Reasons for the very high error rate on some lattices include lack of reference for the particular time segments, or other unpredictable factors (i.e., extreme presence of noise,...). The Phoneme Error Rate (PER) on the training data is 35%. Note that we are working with real-world continuous speech, segmented in particularly long sentences (on average 80 words/sentence).

The Quaero 2010 development data ($4h$) were equally subdivided into test and dev sets, each containing 350 lattices. This data set covers a range of styles, from BN to talk shows. Roughly 50% of the data can be classed as BN and 50% broadcast conversation (BC). These data are considerably more difficult than pure BN data.

An FST decoder is also needed for the experiments presented in Section 5. We use a simple one-pass decoder. The recognition dictionary used as a baseline is the LIMSI American English recognition dictionary with $78k$ word entries with 1.2 pronunciations per word. The pronunciations are represented using a set of 45 phonemes [22]. A 4-gram word LM is used, trained on a corpus of 1.2 billion words of texts from various LDC corpora , news articles downloaded from the web, and assorted audio transcriptions.

5. Results

5.1. Objective calculation

A first control of the correct functioning of the discriminative training is the calculation of the objective on the training data. Only one epoch on the training data is traversed to keep the time of computation low. This is why we actually chose to use online training which has been shown to be asymptotically efficient after a single pass on the training set [20]. The objective is calculated after each 50 iterations (examples) on a randomly chosen sub-set of the training data set.

In the case of the perceptron, the loss function is given in Equation 5. This loss function, in the ideal case, should be zero if no difference between the best hypothesis and the reference was observed. In our case, as can be seen in Figure 1, the loss function converges to a minimum after around 1,250 iterations of the training algorithm.

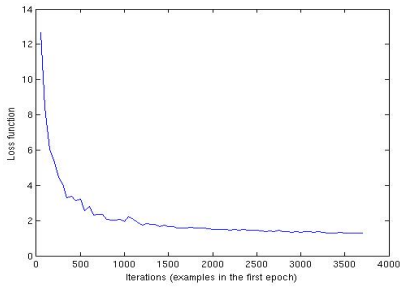

Figure 1: *Perceptron loss on training data*

In the case of the CRF model, we want to train the weights θ while maximizing the conditional log-likelihood (Equation 3). To see some improvement in the upper objective, some normalization of the acoustic weights a_x was necessary before combining them with the weights θ in order to have the weights in the same scale of values. After this normalization, the objective is indeed maximized as expected, though not presented here for lack of space. Note that, for both perceptron and CRF, a convergence towards a stable point is reached within the first epoch on the training data.

5.2. Phoneme Accuracy

Next the phoneme accuracy is calculated, a measure related to the objective function. Slight improvements are observed over the baseline for both the development (dev) and the test sets. Table 1 presents the results on the test set. Note that the proposed simple unigram model can surely not capture the phoneme context dependencies presented in pronunciation modeling. However, some partial improvements can be observed. For example, looking at the column "Deletions" of Table 1, the system with the CRF-trained confusion model reduces the deletion rate from 19% to 16%. The best performance is achieved by the averaged perceptron which slightly improves the phoneme accuracy from

Table 1: *Phoneme Accuracy of the phoneme recognizer on the test set*

System	Phon Acc(%)	Del(%)	Sub(%)	Ins(%)
Baseline	**55**	**19**	20	3
Perceptron	54	19	22	3
Av. Perceptron	**56**	19	20	3
CRF	52	**16**	25	5

55% to 56%. The online training is very sensitive to the order of processing of the examples and taking the average value circumvents this drawback.

Note that adding the confusion model without any training of its weights severely degrades the system's performance. This is because of an augmentation of 126% in the average number of paths in the phoneme lattices of the test set after the application of the confusion model, which adds a detrimental amount of confusability. However, the training of the weights of our confusion model manages to handle the confusability in this doubled search space.

Note also that the acoustic models we use are already context-dependent, plus a 3-gram phonemic LM is used in the phoneme recognizer. That means that a big part of the phonemic variation is already covered by the acoustic model and the phonemic LM. It would be maybe easier to see some improvement if a simple phoneme-loop phoneme recognizer was used to generate the phoneme hypotheses.

5.3. Decoding process

The next step is to introduce the confusion model into the decoding process of a word recognizer and expand the phonemic search space of pronunciations. Thus, instead of using a static recognition lexicon, a dynamically adapted lexicon is produced. To do so, an FST-based decoder is needed, which is not the case of the LIMSI decoder [23]. To circumvent this problem, we decided to add the confusion model in a post-processing step to the 1-best word output of the LIMSI decoder, expressed as an FST W. We compose it with the inverted FST of the pronunciation model Pr^{-1} and the result is a phoneme lattice $A = W \circ Pr^{-1}$.

The Phoneme Accuracy of the baseline phoneme lattice A is 70% (see Table 2). Note that this Accuracy is significantly higher than the Phoneme Accuracy of the phoneme recognizer, which is 55% for the same test set (see "Baseline" in Table 1). Meaning that using these lattices as an input to an FST word decoder will propagate less noise and will result in word sequences of better quality.

The phoneme lattice A is then expanded with the confusion model $C(\theta)$ and a new phoneme lattice $B = W \circ Pr^{-1} \circ C(\theta)$ is generated. The Phoneme Accuracy of the expanded lattice B is 77% (see Table 2), which corresponds to a significant improvement over the baseline. Note that the confusion model we apply to the experiments on the decoding process is the one trained with the CRF model.

Table 2: *Phoneme Accuracy of the word recognizer on the test set)*

	Phon Acc(%)	Del(%)	Sub(%)	Ins(%)
Lattice A	**70**	4	7	20
Lattice B	**77**	7	12	4

Then, we recompose with the pronunciation model Pr and the language model G to produce a new word sequence W_1. To sum up, the series of compositions to get to W_1 is:

$$W_1 = W \circ Pr^{-1} \circ C(\theta) \circ Pr \circ G \quad (7)$$

This series of inverse compositions and recompositions is based on the idea presented in [24], implemented to find the confusable words and predict ASR errors. Ideally the new word sequence W_1 would have a lower word error rate compared to W. However, the following problem occurs: comparing W and W_1 is not a fair comparison because they are not the outputs of the same decoder. Our FST decoder is surely a more simple one compared to the LIMSI decoder. It is an one-pass decoder, keeping no time information and applying no normalization on the output data before scoring. Moreover, since the inverted mappings are one-to-many (i.e., the lexicon Pr includes more than one pronunciations for certain words) and the word boundary information is lost after the compositions, the set W_1 will typically have more elements than W, meaning a lot of homophones. Last but not least, the acoustic scores are lost during the inverse composition. The baseline Word Accuracy of the FST decoder (before introducing the confusion model: "Lattice W_b" in Table 3) is thus lower than the one of the LIMSI decoder (around 70%). The lattice W_b is the result of the post-processing compositions $W_b = W \circ Pr^{-1} \circ Pr \circ G$.

Table 3: *Word Accuracy on the test set*

	Word Acc(%)	Del(%)	Sub(%)	Ins(%)
Lattice W_b	**61**	10	22	6
Lattice W_1	**62**	12	21	5

As can be seen in Table 3, using the confusion model ("Lattice W_1") results in a slight improvement over the baseline ("Lattice W_b"). However, the large improvement observed in the phoneme level (Phoneme Accuracy improved from 70% to 77%, see Table 2) is not propagated when passing to words. This can be again because of the characteristics of the FST-decoder mentioned in the above paragraphs (the acoustic model's infomation is lost, no word-boundaries,...). It is not straightforward though how to integrate the FST-based confusion model to a non-FST decoder.

6. Conclusion and Future Work

We close this paper by summarizing some interesting aspects of this work. Discriminative training of the weights of a phoneme confusion model used to expand the phonemic search space of pronunciations during decoding has been presented. A purely FST-based implementation of training enables us to integrate the training modules and the trained confusion model in any FST-based ASR system. Moreover, working at the phoneme level allows us to add pronunciation variants to any word, without limiting the method to words in the training set.

Experiments were conducted in the context of a continuous speech ASR system on English data segmented in long sentences, which is admittedly a difficult baseline. Even though we used a simple unigram confusion model, no additional confusability was introduced to the system and some improvements were observed. This suggests that this method and its possible extensions can be used to adapt the recognition dictionary to a particular data set .

In the future, we plan to experiment with different objective functions, such as cost-augmented CRF and large-margin methods. In addition, a regularization term will be added to the loss function so as to reduce overfitting and improve generalization performance. Another important direction in which we need to extend the approach is use contextual confusion models: in theory, extending the phonemic context is a simple matter of using a more complex confusion automaton; it remains to be seen how this increased complexity impacts the training and decoding times.

7. Acknowledgements

This work is partly realized as part of the Quaero Programme, funded by OSEO, the French State agency for innovation, and as part of the ANR EdyLex project.

8. References

[1] N. Cremelie and J.-P. Martens, "In search of better pronunciation models for speech recognition," *Speech Communication*, vol. 29, no. 2-4, pp. 115–136, 1999.

[2] M. Riley, W. Byrne, M. Finke, S. Khudanpur, A. Ljolje, J. McDonough, H. Nock, M. Saraclar, C. Wooters, and G. Zavaliagkos, "Stochastic pronunciation modelling from hand-labelled phonetic corpora," *Speech Communication*, vol. 29, no. 2-4, pp. 209–224, 1999.

[3] Q. Yang, J.-P. Martens, P.-J. Ghesquiere, and D. Van Compernolle, "Pronunciation variation modeling for asr: large improvements are possible but small ones are likely to achieve," in *Proc. of PMLA*, 2002, pp. 123–128.

[4] Y. Akita and T. Kawahara, "Generalized statistical modeling of pronunciation variations using variable-length phone context," in *ICASSP*, 2005, pp. 689–692.

[5] C. Van Bael, L. Boves, H. van den Heuvel, and H. Strik, "Automatic phonetic transcription of large speech corpora," *Computer Speech and Language*, vol. 21, no. 4, pp. 652–668, 2007.

[6] M. Weintraub, E. Fosler, C. Galles, Y.-H. Kao, S. Khudanpur, M. Saraclar, and S. Wegmann, "Ws96 project report: Automatic learning of word pronunciation from data," in *JHU Workshop Pronunciation Group*, 1996.

[7] H. Shu and I. Lee Hetherington, "Em training of finite-state transducers and its application to pronunciation modeling," in *Proc. of ICSLP*, 2002, pp. 1293–1296.

[8] I. Badr, I. McGraw, and J. Glass, "Learning new word pronunciations from spoken examples," in *Proc. of Interspeech*, 2010.

[9] O. Vinyals, L. Deng, D. Yu, and A. Acero, "Discriminative pronunciation learning using phonetic decoder and minimum-classification-error," in *ICASSP*, 2009, pp. 4445–4448.

[10] L. Adde, B. Rveil, j.-P. Martens, and T. Svendsen, "A minimum classification error approach to pronunciation variation modeling of non-native proper names," in *Proc. of Interspeech*, 2010, pp. 2282–2285.

[11] S. Greenberg, S. Chang, and J. Hollenback, "An introduction to the diagnostic evaluation of the switchboard-corpus automatic speech recognition systems," in *Proc. of NIST Speech Transcription Workshop*, 2000, pp. 16–19.

[12] N. A. Smith, *Linguistic Structure Prediction*. University ofToronto: Graeme Hirst, 2011.

[13] M. Mohri, "Weighted automata algorithms," *Handbook of weighted automata*, vol. 3, pp. 213–254, 2009.

[14] I. McGraw, I. Badr, and J. Glass, "Learning lexicons from speech using a pronunciation mixture model," *IEEE Transactions on audio, speech and language processing*, vol. 21, no. 2, pp. 357–366, 2013.

[15] K. Gimpel and N. Smith, "Softmax-margin crfs: Training log-linear models with cost functions," in *Proc. of HLT-NAACL*, 2010, pp. 733–736.

[16] J. Lafferty, A. McCallum, and P. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. of ICML*, 2001.

[17] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, Cambridge University Engineering Dept, 2003.

[18] Y. Freund and R. Schapire, "Large margin classification using the perceptron algorithm," *Machine Learning*, vol. 37, no. 3, pp. 277–296, 1999.

[19] M. Collins, "Discriminative training methods for HMMs: theory and experiments with perceptron algorithm," in *Proc. of ACL-02:EMNLP*, vol. 10, 2002, pp. 1–8.

[20] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. of the 19th International Conference on Computational Statistics (COMPSTAT'2010)*, Y. Lechevallier and G. Saporta, Eds. Springer, 2010, pp. 177–187.

[21] H. Robbins and S. Monro, "A stochastic approximation method," *Annals of Mathematical Statistics*, vol. 22, pp. 400–407, 1951.

[22] L. Lamel and G. Adda, "On designing pronunciation lexicons for large vocabulary, continuous speech recognition," in *Proc. of ICSLP*, 1996, pp. 6–9.

[23] J. Gauvain, L. Lamel, and G. Adda, "The limsi broadcast news transcription system," *Speech Communication*, vol. 37, no. 1, pp. 89–108, 2002.

[24] E. Fosler-Lussier, I. Amdal, and H. K. J. Kuo, "A framework for predicting speech recognition errors," *Speech Communication issue on Pronunciation Modeling and Lexicon Adaptation*, vol. 46, no. 2, pp. 153–170, 2005.