Discriminative Language Modeling Using Simulated ASR Errors

Preethi Jyothi[1], *Eric Fosler-Lussier*[1]

[1]Department of Computer Science and Engineering, The Ohio State University, USA
jyothi@cse.ohio-state.edu, fosler@cse.ohio-state.edu

Abstract

In this paper, we approach the problem of discriminatively training language models using a weighted finite state transducer (WFST) framework that does not require acoustic training data. The phonetic confusions prevalent in the recognizer are modeled using a confusion matrix that takes into account information from the pronunciation model (word-based phone confusion log likelihoods) and information from the acoustic model (distances between the phonetic acoustic models). This confusion matrix, within the WFST framework, is used to generate confusable word graphs that serve as inputs to the averaged perceptron algorithm to train the parameters of the discriminative language model. Experiments on a large vocabulary speech recognition task show significant word error rate reductions when compared to a baseline using a trigram model trained with the maximum likelihood criterion.

Index Terms: Language Modeling, Weighted Finite State Transducers, Confusion Matrix, Perceptron Algorithm

1. Introduction

Statistical n-gram language models play a significant role in the speech recognition process by constraining the vast search space of all possible word sequences. N-gram language models are obtained via maximum likelihood estimation from large bodies of text and aim at reducing the perplexity on unseen test data. However, when used in speech recognition, such an objective may not be optimal. Discriminative training algorithms are used to adjust a language model with an objective function that is optimized to improve speech recognition performance (word error rate).

[1] used a discriminative objective function to directly update n-gram counts and change the lexicon to add new words. [2] used the minimum classification error (MCE) training criterion to train the language model via the generalized probabilistic descent (GPD) algorithm. [3] describes a multi-pass algorithm based on global linear models that used the perceptron algorithm for feature selection and then used regularized maximum conditional likelihood to train the language model parameters. [4] proposes a WFST model that generalizes the discriminative linear model described in [3] and incorporates acoustic, duration and language components. For these systems, both acoustic waveforms and their corresponding transcriptions need to be available at training time.

We approach the problem of discriminatively training a language model using only the transcriptions and no acoustic data. Within this framework, we construct a confusion matrix that attempts at closely modeling the phonetic confusions prevalent in the recognizer. As detailed in our previous work [5], we combine counts of phonetic confusions derived from the recognition errors (information from the pronunciation model) along with phone Hidden Markov Model (HMM) distances (information from the acoustic model) to build this confusion matrix. The confusion matrix is a representation of the phonetic and acoustic confusability inherent in the recognizer. [6] adopts a similar motivation of not using acoustic data but use a conditional entropy criteria and a simple phone error model to update the language model parameters. We use a WFST framework to generate confusable hypotheses, that are further used to train our discriminative language model. Apart from the confusion matrix, we only need an initial language model (LM), a lexicon with pronunciations for all the words in the LM, and input transcriptions to train our discriminative language model.

The following section describes the WFST prediction framework used to generate confusable sentences for a given input utterance and details the learning algorithm used to train our discriminative language model. Section 3 describes our experimental setup and details of the task. Section 4 presents our analysis of two sets of experiments in detail. For the first set of experiments, we use training data from the same corpus that was used to train the baseline recognizer for our discriminative LM experiments. The second set of experiments use only transcripts, and no acoustic data, from a different corpus to train our LM. This strengthens our claim that our training model can be used effectively with only transcript data and that the confusion matrix within our WFST framework does a good job of capturing the acoustic errors made by the recognizer. Finally, Section 5 concludes with suggestions for future work.

2. Error Prediction and Learning Algorithms

2.1. Predictive WFST Framework

We employ the framework outlined in [7] that incorporates acoustic confusability, pronunciation model and language model information to generate a lattice of confusable word sequences for a given word sequence using WFSTs. Taking advantage of the invertible nature of transducers, we build a lattice of all possible confusable word hypotheses corresponding to the input utterance W using:

$$W_{\textbf{conf}} = (W \; o \; Lm^{-1} \; o \; P^{-1} \; o \; Ac^{-1} \; o \; Ac \; o \; P \; o \; Lm) \qquad (1)$$

where Ac is an FST that maps acoustic features to phones using the acoustic model scores, P is the pronunciation model FST mapping phones to words and Lm is the language model FSA (with ngram scores). Since Lm is a deterministic finite state automaton, Lm^{-1}=Lm; composition with Lm^{-1} is not necessary as it just provides a constant scaling for the correct string W. Given the infeasibility of the task of representing the continuous space of acoustic features ($Ac^{-1} \; o \; Ac$) within a WFST framework, [7] assumes that the acoustic errors made by a recognizer can be encapsulated within a confusion matrix (C) between phones derived from speech recognition errors. (1) now becomes:

$$W_{\textbf{conf}} = (W \ o \ P^{-1} \ o \ C \ o \ P \ o \ Lm) \tag{2}$$

In our previous work [5], we observed that providing the confusion matrix with acoustic model information in the form of distances between confusable phone HMMs, along with word-based phone confusion scores, improved the prediction capabilities of the confusion matrix. The distance between two phone HMMs was considered to be a weighted sum of the average distance between the Gaussian Mixture Models (GMMs) of the aligned states for every possible alignment of HMM states, normalized by the sum of all weights. The weights are derived from the probability of the alignment between the states of the HMMs of the two phones in question. In this paper, we make use of outputs from this WFST framework, in the form of confusable word graphs ($W_{\textbf{conf}}$), to train a linear discriminative language model built using the perceptron algorithm.

2.2. Linear Discriminative Language Model

We approach the problem of learning parameters of a discriminative language model using a global linear model as described in [8][3]. Given a set of training examples (x_i,y_i) where $x_i \in X$, $y_i \in Y$ for $i = 1 \ldots N$,a function **GEN** that lists a set of candidates **GEN**(x) for an input x, a feature vector $\Phi(x, y) \in \mathbb{R}^d$ for each $(x, y) \in X \times Y$ and a parameter vector $\overline{\alpha} \in \mathbb{R}^d$, there is a mapping from an input x to $F(x)$ given by the equation:

$$F(x) = \underset{y \in \textbf{GEN}(x)}{\text{argmax}} \ \Phi(x, y) \cdot \overline{\alpha} \tag{3}$$

where $\Phi(x, y) \cdot \overline{\alpha}$ is the dot product $\sum_i \alpha_i \Phi_i(x, y)$. We use the training examples (x_i,y_i) to learn the parameter values $\overline{\alpha}$ and the decoding algorithm searches for a value of y that maximizes (3). We use the perceptron algorithm shown in Fig. 1 [3] to train the parameters $\overline{\alpha}$ of the model. As in [8], we use the averaged parameter values during decoding; $\overline{\alpha}_{AVG} = \sum_{i,t} \overline{\alpha}_i^t / NT$, where $\overline{\alpha}_i^t$ is the parameter vector obtained after the t'th iteration on the i'th example in the algorithm described in Fig. 1.

2.2.1. WFST Implementation of the Perceptron Algorithm

Similar to [3], we use WFSTs to implement the perceptron algorithm. This fits in nicely with our prediction framework that outputs the final confusable word graph as a deterministic WFST. To apply the perceptron algorithm outlined in Fig. 1 to our generated confusion strings, we need to define the training examples (x_i, y_i), **GEN** and the feature vector mapping Φ. $x_i \in X$ are the input word sequences, **GEN**(x_i) is the set of word hypotheses obtained from the confusable word graph corresponding to x_i ($W_{\textbf{conf}}$ in (2)) and $y_i \in Y$ is the reference transcription.

The main difference in our model from the one proposed in [3] is that we use confusable word graphs generated from input transcriptions as opposed to word lattices produced by the recognizer and this is reflected in the first dimension, $\phi_0(x, y)$ of the feature vector Φ. We set $\phi_0(x$,$y)$ to be the sum of weights along the path of a confusable hypothesis in the confusable word graph and this serves as an acoustic surrogate to the log score output from the baseline recognizer word lattices used by [3]. We only require the reference transcriptions to generate our confusable strings and this eliminates the need for acoustic data.

```
Inputs: Training examples $(x_i, y_i)$
Initialization: Set $\overline{\alpha} = 0$
Algorithm:
    For t = $1 \cdots T, i = 1 \cdots N$:
    Calculate $z_i = \text{argmax}_{z \in \textbf{GEN}(x_i)} \Phi(x_i, z) \cdot \overline{\alpha}$
    If $(z_i \neq y_i)$ then $\overline{\alpha} = \overline{\alpha} + \Phi(x_i, y_i) - \Phi(x_i, z_i)$
Output: Parameters $\overline{\alpha}$.
```

Figure 1: *Collins' perceptron algorithm [3]*

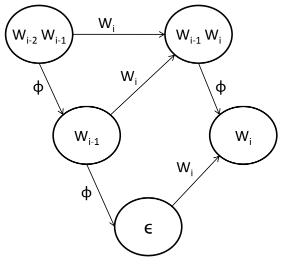

Figure 2: *Trigram model represented as a WFA with failure transitions*

The remaining features in Φ correspond to the unigram, bigram and trigram counts of all the n-grams (up to length 3) that appear in the confusable word graphs of all the training set sentences. These features can be efficiently represented as a deterministic weighted finite-state automaton (WFA) [9]. Fig. 2 shows how a trigram model can be represented as a WFA. There are transitions corresponding to each word w_i leaving a bigram history state $w_{i-2}w_{i-1}$ and there are failure transitions, labeled ϕ, that are traversed only if the next word in the input word sequence does not correspond to any of the word arcs leaving this history state. In such a representation, the weight of a word arc w_i leaving a history state $w_{i-2}w_{i-1}$ must account for the trigram($w_{i-2}w_{i-1}w_i$), bigram($w_{i-1}w_i$) and unigram (w_i) weights since encountering a trigram feature automatically implies that the corresponding bigram and unigram features also need to be taken into consideration. We note that for our experiments, the perceptron reached optimal performance after T = 1 or 2 iterations. Thus, the number of n-gram features with non-zero weights is very small compared to the total number of n-grams seen in the training word graphs.

Using the above definitions for (x_i, y_i), **GEN** and the feature vector mapping Φ, the algorithm in Fig. 1 is implemented using WFSTs as shown in Fig. 3. Since we generate the confusable word graphs using the reference transcriptions, the best path from $(\alpha_0 W_{\textbf{conf}_i} \ o \ D)$ returns the reference transcription. Thus, in order to account for more confusable words in the decoded sentence (z_i), we extract the n^{th} best confusable sentence from the word confusable graph where n=100, 500 or 1000. In Section 4, we will describe in detail the motivation behind

```
Inputs: Confusable Word Graphs $W_{\textbf{conf}_i}$ and
reference transcriptions $r_i$ for $i = 1 \ldots N$.
Value of $\alpha_0$ is chosen to be a fixed constant by
optimization on the development set ($\alpha_0 = 1$).
Initialization: $D$ is the discriminative language
model with all weights corresponding to the
ngrams in $D$ set to 0.
Algorithm:
  For $t = 1 \ldots T, i = 1 \ldots N$:
  Compute $z_i$ = $\textbf{n}^{\textbf{th}}\textbf{best}$ $(\alpha_0 W_{\textbf{conf}_i} \circ D)$
  Update $\alpha_j = \alpha_j + \phi_j(x_i, r_i) - \phi_j(x_i, z_i)$
  corresponding to the weights on the arcs in
  $D$ for all $j = 1 \cdots d$ where $\Phi(x, y) \in \mathbb{R}^d$.
```

Figure 3: *Perceptron Algorithm Implementation using WFSTs*

choosing a value for n and we present speech recognition results using different values of n.

3. Experimental Setup

We present preliminary results from experiments on the Switchboard-1 Telephone Speech Corpus, Release 2, 1997 that contains manually corrected word alignments and transcriptions. The training set consists of 154723 transcribed utterances. The development set contains 2235 sentences and the evaluation test set contains 638 sentences. The baseline recognizer for this task was built using the Hidden Markov Model Toolkit (HTK) [10]. The acoustic model was developed with tied-state intra-word triphones and the state output distributions from each of the phone HMMs were modeled as 16-component Gaussian mixtures. The pronunciation model was derived using the CMU dictionary. A bigram language model was built using the maximum likelihood criterion from all the sentences in the training set. The lattices derived from the recognizer were rescored using a trigram language model and the baseline word error rates (WERs) were generated from these rescored lattices. We build the confusion matrix utilizing phone confusion counts and HMM distances, as outlined in [5], using the ASR transcriptions of the 2235 sentences in the development set.

4. Experimental Design and Analysis

We conduct two sets of experiments to demonstrate the utility of confusable word graphs from our predictive WFST framework to train our discriminative language model without making use of any acoustic data. Due to time constraints, the first set of experiments uses a smaller subset of the Switchboard training set, comprising of 20000 sentences, to train our discriminative language model. To further reiterate our hypothesis that this framework is useful in cases where there is no real acoustic data, our second set of experiments use 10000 sentences from the Fisher corpus (Linguistic Data Consortium, LDC 2003) to train the discriminative language model. Both sets of experiments and the results are analyzed in detail in the following sections.

4.1. Error Prediction on the Switchboard Corpus

We evaluate the predictive capability of our framework by computing the fraction of errors that are correctly predicted when a threshold of "n" sentences from the word confusable graph is applied. We calculate the number of "error chunks" that are correctly predicted by the framework within the threshold of "n" for all the sentences in the development set. "Error chunks" are identified by removing the longest common subsequence of the correct sentence and the misrecognized sentence from each of these sentences and grouping together the remaining chunks of the two sentences. For example, if the reference sentence is "oh the last big lake i went to was Lake Fork" and the misrecognized sentence is 'all the last big lake going to was Lake Fork", the error chunks after eliminating the longest common subsequence ("the last big lake to was Lake Fork") would be "oh:all" and "i went:going". Recall ranks of the generated "error chunks" are an important measure in that it computes the capacity of our system to generate the misrecognized strings. From Fig. 4, we observe that only about 30% of the erroneous word chunks are correctly predicted for $n = 100$. For higher values of n ($n = 500/1000$), about 42% of the erroneous word subsequences are correctly predicted. This is closer to the word accuracy rates obtained from the recognizer on the development set (49%) indicating that a higher value of n should perform better for our task of discriminatively training the language model.

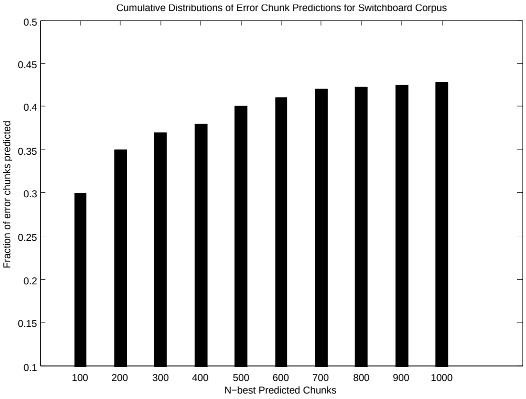

Figure 4: *Recall rank of recognized error chunks in the development set of the Switchboard Corpus*

Table 1: *Table of word error rates on the Switchboard test-set using different confusion matrices*

System	Word error rate (%)
Baseline	49.7
FST-Words	49.2
FST-Words+HMM	48.8

4.2. Discriminative LM Experiments on Switchboard

We consider the 100^{th}, 500^{th} and 1000^{th} best sentence from the confusable word graphs to update the ngram weights of those features that differ from the ngram features in the reference sentence. Table 1 shows us the word error rates (WERs) for two models based on two different confusion matrices. Both systems "FST-Words" and "FST-Words+HMM" use 10000 training sentences to train the discriminative LM and set $n = 1000$. "FST-Words" is built using a confusion matrix that sets the cost of each phone-phone mapping by computing an alignment between the phonetic transcriptions of the reference sentence and the recognized sentence. "FST-Words+HMM" uses a confusion matrix with its costs set to phone HMM distances along with the phone-phone log-likelihood scores to generate the confusable word graphs. We observe that "FST-Words+HMM" gives a larger reduction in WER as compared to "FST-Words" reiterating that adding information from the underlying HMM topology of the phones in the form of HMM distances to the confusion matrix helps provide a more accurate picture of the possible phone confusions [5]. In our previous work [5], we found that incorporating lexical constraints gave superior predictive performance over pure phone-recognitions with no lexicon.

Table 2 gives us the WERs corresponding to the three different values of n. The baseline model consists of a maximum likelihood trained trigram model. The system "With Lats" corresponds to using word lattices of the training sentences from the baseline recognizer to train the discriminative language model, as described in [3]. Keeping with the approach in [3], the lattices are produced with an acoustic model that was trained on the entire training set but with a language model that was trained on data portions that did not include the current utterance. Both "With Lats" and our system with $n = 1000$ give a statistically significant reduction ($p<0.05$) in WER over the baseline. Also, we observe that our system with $n = 1000$ performs comparably to "With Lats" that uses acoustic data in the form of word lattices from the recognizer. Additionally,

Table 2: *Table of word error rates on the Switchboard test-set building discriminative LM using the Switchboard training set*

System	Word error rate (%)
Baseline	49.7
With Lats (audio), $\alpha_0 = 2$ [3]	48.4
With FSTs (audio-free), n = 100	49.2
With FSTs (audio-free), n = 500	48.7
With FSTs (audio-free), n = 1000	48.5

Table 3: *Table of word error rates for experiments on the Switchboard test-set using sentences from the Fisher corpus*

System	Word error rate (%)
Baseline	49.7
Baseline(with Fisher)	49.4
With FSTs, n = 1000	48.8

our word graphs were generated an order of magnitude faster than generating training lattices using HTK. It should be noted that we used a reference gold standard in "With Lats" and not a minimum word error gold standard as mentioned in [8]. For $n = 500$, the drop in WER from the baseline is almost statistically significant and for $n = 100$, the fall in WER is not statistically significant compared to the baseline. This is in line with the observations from our previous evaluation task that for higher values of n, more acoustic errors of the recognizer are captured. This increases the number of ngrams with a non-zero weight in our newly trained discriminative language model and contributes to the reduction in WER of the recognizer.

4.3. Experiments Using Fisher Corpus

Our second set of experiments aim at evaluating the performance of our discriminative language modeling algorithm by using audio-free transcript data from a corpus different from the one used to train the baseline recognizer. We use 10000 transcripts from the Fisher corpus to train our model. The sentences were chosen such that there were no out of vocabulary words with regards to the Switchboard corpus and were of least perplexity when evaluated against the trigram language model built using all the training sentences from the Switchboard corpus. "Baseline(with Fisher)" consists of a trigram language model that was trained using all the training set sentences from the Switchboard corpus and the newly extracted 10000 Fisher transcripts. Table 3 shows that our system with $n = 1000$ gives a statistically significant improvement ($p<0.05$) over "Baseline". It also shows an improvement over "Baseline(with Fisher)" that is not statistically significant and we believe this is because we used only 10000 sentences to train the model which may not be sufficient for a large vocabulary task of spontaneous conversations. This result is promising in that we see improvements in the performance of the recognizer by using our system with audio-free transcripts from a corpus different from the one used to train the baseline recognizer.

5. Conclusions

We used a predictive WFST framework, consisting of a confusion matrix that uses pronunciation and acoustic model information from the recognizer to model possible phone confusions, to generate confusable word sequences corresponding to an input word utterance. We further used these erroneous word hypotheses to train a global linear discriminative language model using the averaged perceptron algorithm. We observe that our system performs significantly better than a baseline recognzer that uses a generatively trained trigram language model. Also, it performs comparably to a system that uses acoustic data in the form of word lattices from a recognizer and runs an order of magnitude faster. Though these results need to be validated with a stronger baseline system, they are promising in that they are indicative of the fact that the confusion matrix is a fitting model for the phonetic confusions prevalent in the recognizer and thus allows us to use only transcript data (no audio data) to get significant improvements in word accuracy rates from the recognizer. To further this point, we use transcripts from the Fisher corpus to train our discriminative language model and observe that we see a reduction in WER when compared to baselines that use generatively trained trigram LMs. This is an encouraging result in that it allows us to use audio-free transcripts from a different corpus than the one used to train the acoustic model of the baseline recognizer and still report improvements in the performance of the recognizer. Future work will include expanding the feature set beyond simple n-gram features and incorporating additional features such as part-of-speech information that can be extracted from just the transcripts. We also intend to experiment with different distributions around n rather than using a single n^{th} best hypothesis from our confusable word graphs.

6. Acknowledgements

This research was supported by the NSF Career grant IIS-0643901. The opinions expressed in this work are those of the authors and do not necessarily reflect the views of the funding agency.

7. References

[1] Chen, Z., Lee, K-F. and Li, M. J., "Discriminative training on language model", Proceedings of the Sixth International Conference on Spoken Language Processing, Beijing, 2000.

[2] Kuo, H-K. J., Fosler-Lussier, E., Jiang, H. and Lee, C-H., "Discriminative training of language models for speech recognition", Proceedings of the International Conference on Acoustics, Speech and Signal Processing, Florida, 2002.

[3] Roark, B., Saraclar, M., Collins, M. and Johnson, M., "Discriminative language modeling with conditional random fields and the perceptron algorithm", Proceedings of the 43rd Meeting of the Association for Compuational Linguistics, 507-514, 2005.

[4] Lehr, M., and Shafran, I., "Discriminatively estimated joint acoustic,duration and language model for speech recognition", Proceedings of the International Conference on Acoustics, Speech and Signal Processing, Dallas, 2010.

[5] Jyothi, P. and Fosler-Lussier, E., "A comparison of audio-free speech recognition error prediction methods", Proceedings of Interspeech, Brighton, 2009.

[6] Huang, J-T., Li, X. and Acero, A., "Discriminative training methods for language models using conditional entropy criteria", Proceedings of the International Conference on Acoustics, Speech and Signal Processing, Dallas, 2010.

[7] Fosler-Lussier, E., Amdal, I. and Kuo, H-K. J., "A framework for predicting speech recognition errors", Speech Communication, 46:153-170, 2005.

[8] Collins, M., "Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms", Proceedings of Empirical Methods in Natural Language Processing, 1-8, 2002.

[9] Allauzen, C., Mohri, M. and Roark, B., "Generalized algorithms for constructing language models", Proceedings of the 41st Meeting of the Association for Computational Linguistics, 40-47, 2003.

[10] Young, S., "The HTK Hidden Markov Model Toolkit: Design and Philosophy", Online: http://htk.eng.cam.ac.uk, 1993.