# Discriminative Training of N-gram Classifiers for Speech and Text Routing

*Ciprian Chelba, Alex Acero*

Microsoft Research
Microsoft Corporation, One Microsoft Way,
Redmond, WA 98052, USA
{chelba,alexac}@microsoft.com

## Abstract

We present a method for conditional maximum likelihood estimation of N-gram models used for text or speech utterance classification. The method employs a well known technique relying on a generalization of the Baum-Eagon inequality from polynomials to rational functions. The best performance is achieved for the 1-gram classifier where conditional maximum likelihood training reduces the class error rate over a maximum likelihood classifier by 45% relative.

## 1. Introduction

Speech utterance classification as well as text classification are an interesting subproblem in a growing trend of providing natural language user interfaces to automated systems. A straightforward application among many others is call-routing, a problem tackled by various research groups: [1], [2].

Text classification and/or categorization literature is far from scarce since it is a straightforward application of various classification techniques. A good starting point for further reading is [3].

The current work aims at comparing various design decisions when devising an N-gram utterance classifier: should one use a one-pass system — concurrent speech utterance classification and recognition — or a two-pass system — speech recognition followed by text classification? Are discriminative training techniques effective? Our goal is to provide experimental results that would guide various choices or further experiments when designing such a system.

The paper is organized as follows: Section 2 describes the problem setup and reviews the N-gram classifier as well as maximum likelihood (ML) versus conditional maximum likelihood (CML) training for N-gram classifiers. Section 3 outlines the re-estimation procedure for CML training of N-gram classifiers. Section 4 presents various experiments along the directions outlined previously. We conclude with a short analysis of the results and brief outline of future directions that we believe to be worthwhile exploring.

## 2. Utterance Classification

To fix notation, we denote a speech utterance with $A$, the word string that gave rise to it with $W = w_1 \dots w_n$ and the class of the utterance with $C$. The word vocabulary is denoted with $\mathcal{V}$ and the class vocabulary with $\mathcal{C}$. The corpus, split in training and test data, $\mathcal{T}$ and $\mathcal{E}$, respectively, consists of tuples (or samples) $s$ containing: utterance $A$, transcription $W$ and utterance class $C$. The performance of a given classifier is measured by the class error rate (CER):

$$CER = 1/|\mathcal{E}| \sum_{s \in \mathcal{E}} \delta(s.C, \widehat{C}(s.A)) \quad (1)$$

where $s.A$ denotes the acoustics in sample $s$, $s.C$ is the correct class for sample $s$, $\widehat{C}(s.A)$ is the class assigned to $s.A$ by the classifier and $\delta(\cdot,\cdot)$ is the Kronecker-$\delta$ operator.

### 2.1. N-gram Classifier

#### *2.1.1. One-pass Utterance Classification*

N-gram classifiers have the advantage that one can use them in a single pass system whereby a given utterance $A$ is assigned a class $\widehat{C}(A)$ concurrently with speech decoding that finds the word string $\widehat{W}$.

Assume one builds an N-gram model $P(w_i|w_{i-1}, \dots, w_{i-N+1}, C)$ for each class $C \in \mathcal{C}$ by pooling all the training transcriptions that have been labeled with class $C$.

In a one-pass scenario one builds a recognition network that stacks each of the language models $P(\cdot|C)$ in parallel, the transition into each language model $P(\cdot|C)$ having score $\log P(C)$. The decoder search for the most likely path will find

$$(\widehat{C}(A), \widehat{W}) = \arg\max_{(C,W)} \log P(A|W) + \log P(W|C) + \log P(C)$$

#### *2.1.2. Two-pass Utterance Classification*

In a two-pass scenario one builds a *pooled* N-gram language model $P_{pooled}(w_i|w_{i-1}, w_{i-N+1})$ from all the training transcriptions in addition to the class specific language models $P(\cdot|C)$. Each test utterance is then assigned a class by doing text classification on the 1-best recognition output using the *pooled* language model. The second stage of the two-pass approach implements an N-gram text classifier.

$$\widehat{W} = \arg\max_{W} \log P(A|W) + \log P_{pooled}(W)$$

$$\widehat{C}(A) = \arg\max_{C} \log P(\widehat{W}|C) + \log P(C)$$

### 2.2. Estimation of N-gram Classifier Parameters

#### *2.2.1. Maximum Likelihood Estimation of N-gram Classifiers*

We have found smoothing to be a very important issue for all classifiers that we experimented with. For estimating the N-gram models we use the recursive deleted interpolation scheme [4] between relative frequency estimates at different orders,

$f_k(\cdot), k = 0 \ldots N-1$:

$$P_n(w|h_n) = \lambda(h_n) \cdot P_{n-1}(w|h_{n-1}) + \overline{\lambda(h_n)} \cdot f_n(w|h_n), \quad P_{-1}(w) = uniform(\mathcal{V}) \quad (2)$$

where

$$\overline{\lambda(h_n)} = 1 - \lambda(h_n)$$
$$h_n = w_{-1}, \ldots, w_{-n}$$

The relative frequencies and interpolation weights are estimated using maximum likelihood from main and held-out data, respectively, obtained by a 70/30% random split of the training data available to a given language model. The interpolation weights are bucketed according to the context count: $\lambda(h_n) = \lambda(C_{ML}(h_n))$.

#### *2.2.2. Conditional Maximum Likelihood Estimation of N-gram Classifiers*

Training the class specific N-gram models using the maximum likelihood criterion is suboptimal when the set of N-gram models is used for classification. A better training criterion tunes all the language models $P(W|C), \forall C$ jointly such that the CER (1) is minimized.

Since the CER is not analytically tractable, a convenient substitute is the conditional probability $\prod_{i=1}^{T} P(s_i.C|s_i.A)$ for speech utterance classification or $\prod_{i=1}^{T} P(s_i.C|s_i.W)$ for text classification, where $T$ is the number of samples in the training data. The choice is justified by the fact that this is inversely correlated with the CER on the training data.

Restricting our attention to the text only case, we seek to tune the language models $P(W|C), \forall C$, to maximize the conditional log-likelihood:

$$L(C|W) = \sum_{i=1}^{T} \log P(s_i.C|s_i.W) \quad (3)$$
$$P(C|W) = P(C) \cdot P(W|C) / \sum_{L \in \mathcal{C}} P(L) \cdot P(W|L)$$

For efficiency reasons we wish to keep the same parameterization as in the ML case (see Eq. 2) for the class specific language models: the storage requirements and run time should be the same as for the ML models. This is a constraint that is not strictly necessary and could be removed, however it was our choice for the current approach.

Another important observation that needs to be made is that for speech utterance classification one wishes to tune the language models $P(W|C), \forall C$ such that the conditional likelihood $L(C|A)$ is maximized:

$$L(C|A) = \sum_{i=1}^{T} \log P(s_i.C|s_i.A) \quad (4)$$
$$P(C, A) = P(C) \sum_{W \in \mathcal{V}^*} P(W|C) \cdot P(A|W)$$
$$P(C|A) = P(C, A) / \sum_{L \in \mathcal{C}} P(L, A)$$

The main difference between tuning the language models for maximizing $L(C|A)$ (Eq. 4) versus $L(C|W)$ (Eq. 3) is that in the former case the language model will take into account the acoustic confusability between words and will try to discount the contribution of highly confusable words to the classification result. The current approach maximizes Eq. (3) which requires word transcriptions along with class labels for language model training (the acoustic data is not used for language model training).

## 3. Rational Function Growth Transform for CML N-gram Parameter Estimation

As explained in Section 2.2.2 each class specific N-gram model is parameterized according to Eq. (2). The goal of the CML estimation procedure is to tune the relative frequency values $f_n(w|w_{-1}, \ldots, w_{-n})$ at all orders n = $0 \ldots N-1$ such that the conditional likelihood of the training data $L(C|W)$ (see Eq. 3) is maximized.

The optimization technique used in our approach is the rational function growth transform (RFGT) [5]. The main reason for our choice is the fact that the RFGT procedure operates on probability distributions and thus it enforces at each iteration the proper normalization of the probability distributions underlieing the model parameterization. Due to lack of space we omit the derivation of the reestimation equations; under the parameterization in Eq. (2), the relative frequency $f_k(w|h_k, c)$ of order $k$ is estimated iteratively using:

$$f_k^+(w|h_k, c) = f_k(w|h_k, c) \frac{1 + \beta(h_k, c) \frac{\alpha(h_k, c)}{P_n(w|h_n, c)} \cdot f_k^{CML}(w|h_k, c)}{norm(h_k, c)} \quad (5)$$

where

$$norm(h_k, c) = 1 + \beta(h_k, c) \sum_{w \in \mathcal{V}} \frac{\alpha(h_k, c)}{P_n(w|h_n, c)} f_k(w|h_k, c) \cdot f_k^{CML}(w|h_k, c)$$
$$\alpha(h_k, c) = \overline{\lambda_k(h_k, c)} \prod_{l=k+1}^{n} \lambda_l(h_l, c)$$
$$f_k^{CML}(w|h_k, c) = \frac{C_{CML}(w, h_k, c)}{C_{ML}(h_k, c)}$$

$C_{ML}(w, h_k, c)$ denotes the ML count of $(w, h_k)$ in sentences of class $c$ gathered from the training data:

$$C_{ML}(w, h_k, c) = \sum_{i=1}^{T} C((w, h_k) \in s_i.W) \cdot \delta(c, s_i.C)$$

$C_{CML}(w, h_k, c)$ denotes the "CML count" of $(w, h_k)$ in sentences of class $c$:

$$C_{CML}(w, h_k, c) = \sum_{i=1}^{T} C((w, h_k) \in s_i.W) \cdot [\delta(c, s_i.C) - P(c|s_i.W)]$$

with the probability $P(c|W)$ of a class $c$ given a sentence $W = w_1, \ldots, w_q$ being assigned using the class specific N-gram models $P_n(w|h, c)$ and the prior probability on classes $P(c)$:

$$P(c|W) = \frac{P(c) \cdot P(W|c)}{\sum_{d \in \mathcal{C}} P(d) \cdot P(W|d)} \quad (6)$$
$$P(W|c) = \prod_{i=1}^{length(W)} P_n(w_i|h_i, c)$$

The context sensitive "CML weight" $\beta(h_k, c)$ is set to $\beta_{max} > 0$ at each iteration. Following the development in [5],

its value is then lowered for each context $\beta(h_k, c)$ separately such that the numerator in Eq. (5) is non-negative for all events $(w, h_k, c)$:

$$1 + \beta(h_k, c)\frac{\alpha(h_k, c)}{P_n(w|h_n, c)} \cdot f_k^{CML}(w|h_k, c) > \epsilon$$

We note that for most contexts $(h_k, c)$ this adjustment is not necessary if a low enough $\beta_{max}$ value is chosen to start with. The downside of choosing a small $\beta_{max}$ value is that the relative increase of $L(C|W)$ at each iteration is smaller.

The model parameters are estimated by splitting the training data randomly in main and held-out data; the same partition as the one for ML training is used. The main data is used for estimating the relative frequencies, whereas the held-out data is used for tuning the number of RFGT iterations and the optimal CML weight $\beta_{max}$. The class priors and the interpolation weights are not re-estimated. The initial values for the class priors, relative frequencies and the interpolation weights are the ML ones.

The number of RFGT iterations and the optimal CML weight $\beta_{max}$ are determined as follows:

- fix a preset number N of RFGT iterations to be run
- fix a grid (range of values and step) to be explored for determining the $\beta_{max}$ parameter value
- for each value $\beta_{max}$ run as many RFGT iterations as possible (no more than N) such that the conditional likelihood $L(C|W)$ of the main data increases at each iteration;
- retain the (no iterations, $\beta_{max}$) pair that maximizes the conditional likelihood $L(C|W)$ of the held-out data as the desired values

After determining the number of RFGT iterations to be run and the $\beta_{max}$ value, the main and held-out data are pooled and the model is trained using these values.

# 4. Experimental Results

We have employed the CML training technique described in Section 2.2.2 for training N-gram classifiers. The classifiers were evaluated on both speech and text in the same ATIS setup as the one used in [6].

## 4.1. Experimental Setup

All experiments were carried out on the ATIS corpus [7]. We have pooled the ATIS II and ATIS III data after which we extracted the type A utterances (that can be interpreted independent of context) along with their transcriptions. We have used the ATIS III dev94 class A utterances as a development set for tuning the speech recognition system. The test set was obtained by pooling the ATIS III 93 and 94 test sets such that enough utterances of class A were available for testing our classifier.

The acoustic model was trained on all the ATIS II and III training data, irrespective of utterance class (A, D or X). We have built a standard tied-state cross-word triphone HMM acoustic model using the HTK [8] training tools. The model uses 3 states per triphone. After tri-phone clustering there were 1979 states. The model uses a total of 35,661 Gaussians, each modeled using diagonal covariance matrix. The acoustic feature vector consists of 13 Mel-frequency static cepstral coefficients (MFCC) derived using cepstral mean normalization, together with delta and delta-delta coefficients, thus resulting in a 39-dimensional feature vector.

For language model and classifier training we have used only type A utterances along with a class label assigned by taking the argument of the first `SELECT` statement in the SQL query associated with the utterance. There were 14 classes derived in this manner, their distribution being highly skewed towards the `FLIGHT` class which covers about 74% of the utterances. The training data consisted of 5,822 class A utterances (74,442 words). The test data consisted of 914 class A utterances (10,673 words). The development data consisted of 410 utterances (5,326 words).

The vocabulary derived from the training data had size 780 and out-of-vocabulary rate on test data 0.24%. The pooled 1,2,3-gram language models built on the above vocabulary had test set perplexity 149, 19, 15, respectively.

## 4.2. Classification Experiments

### *4.2.1. One Pass*

Table 1 contrasts the performance of N-gram one-pass classifiers of different orders (n=1,2,3) when trained under the ML and CML criteria, respectively. Each classifier is also evaluated on the correct transcription for the test utterances.

| N-gram | CER(1-best) | WER | runtime |
|---|---|---|---|
| ML 1gram | 11.8% | 12.3% | 6.6hrs |
| Transcript | 10.6% | 0% | —- |
| ML 2gram | 8.5% | 6.3% | 3.15hrs |
| Transcript | 9.3% | 0% | —- |
| ML 3gram | 9.0% | 5.5% | 3.36hrs |
| Transcript | 9.6% | 0% | —- |
| CML 1gram | 6.6% | 12.6% | 6.3hrs |
| Transcript | 6.7% | 0% | —- |
| CML 2gram | 7.4% | 6.3% | 3.2hrs |
| Transcript | 8.6% | 0% | —- |
| CML 3gram | 8.3% | 5.5% | 2.8hrs |
| Transcript | 9.2% | 0% | —- |

Table 1: Classification error rate (CER), word error rate (WER) and decoding time (1GHz Pentium) for a one-pass system using an **N-gram classifier**; CER(1-best): the class label along the most likely path in the 1-pass $(W, C|A)$ lattice is retained as the class for the utterance; the $(W, C|A)$ lattice is obtained using a **maximum likelihood and conditional maximum likelihood N-gram classifier** $P(W|C)P(C)$ as the language model, respectively

The CML trained classifiers outperform their ML counterpart at all orders. The best performance is achieved by the CML 1-gram classifier where CML training reduces the CER over a ML classifier by 45%. Since the CML models have the same number of parameters as their ML counterpart, the run-times of CML models are roughly the same as the ones for the ML models.

### *4.2.2. Two Pass*

We have also evaluated the N-gram classifiers in a two-pass scenario.

The first pass uses "pooled" N-gram language models of various orders (1,2,3) built from the entire training data, irrespective of class. Table 2 presents the WER results for the first pass, along with run-times.

| 1st pass LM | WER | runtime |
|---|---|---|
| 1gram | 13.0% | 4.2hrs |
| 2gram | 6.0% | 1.8hrs |
| 3gram | 5.1% | 1.8hrs |

Table 2: Word Error Rate (WER) and decoding time (1GHz Pentium) for various N-gram decoders used in the first pass of various two-pass classifiers

The second pass employs N-gram classifiers of different orders, trained under ML and CML. As it can be easily seen by comparing the results in Tables 1 and 3, the one-pass N-gram classifiers outperform their two-pass counterpart.

| 1st pass | 2nd pass | CER | |
|---|---|---|---|
| | | ML | CML |
| 1gram | 1gram | 12.0% | 7.5% |
| 2gram | 1gram | 11.1% | 7.5% |
| 3gram | 1gram | 10.7% | 7.2% |
| Transcript | 1gram | 10.6% | 6.7% |
| 1gram | 2gram | 11.2% | 9.3% |
| 2gram | 2gram | 9.6% | 8.8% |
| 3gram | 2gram | 10.1% | 9.0% |
| Transcript | 2gram | 9.3% | 8.6% |
| 1gram | 3gram | 11.2% | 10.3% |
| 2gram | 3gram | 9.4% | 9.1% |
| 3gram | 3gram | 9.4% | 9.2% |
| Transcript | 3gram | 9.6% | 9.2% |

Table 3: Classification error rate (CER) for a two-pass system: 1-best word string from N-gram decoder is fed to N-gram classifier trained under maximum likelihood (ML) or conditional maximum likelihood (CML)

The effectiveness of CML training decreases with the n-gram order for both one- and two-pass classifiers. We attribute this to the fact that the higher order n-gram classifiers are "overtrained": if the term corresponding to the correct class dominates all the others in the sum $\sum_{d\in\mathcal{C}} P(d)\cdot P(W|d)$ (see Eq. 6) then the CML training cannot be very effective. Table 4 summarizes the values of the conditional cross-entropy $H(C|W) = -1/T\sum_{i=1}^{T}\log P(s_i.C|s_i.W)$ on the training data before (ML) and after CML training (CML).

The conditional cross-entropy for the ML 2/3-gram models is an order of magnitude smaller than the one for the 1-gram model so the effectiveness of the CML training procedure can be expected to be smaller in those cases.

| n-gram order | $H(C\|W)$ (nats) | |
|---|---|---|
| | ML | CML |
| 0 ($P(C)$ prior) | 1.03 | 1.03 |
| 1 | 0.24 | 0.028 |
| 2 | 0.03 | 0.0036 |
| 3 | 0.007 | 0.0016 |

Table 4: Conditional cross-entropy $H(C|W)$ on training data before and after CML training for different n-gram orders

## 5. Conclusions

Discriminative training under the CML criterion is an effective technique for improving classification accuracy. For N-gram classifiers it is more effective in a one-pass classification scenario than in a two-pass one.

The RFGT training technique doesn't perform equally well for all N-gram orders when the parameterization of the N-gram model is fixed such that only relative frequencies for N-grams observed in a given class are stored as model parameters and the interpolation weights are fixed to their maximum likelihood value. An undesirable result is that the higher order N-grams (2,3) do not perform as well as the 1-gram model after training under the CML criterion using RFGT.

Future research in this area should explore ways of tieing the interpolation weights that would allow higher order N-gram information to be incorporated constructively into the CML 1gram.

Another interesting research direction would be to relax the constraint of having the same number of parameters as the ML models and reduce the model size by pruning after completing CML training on a fully parameterized model.

Yet another direction worth exploring is the training of the classifier such that the conditional likelihood $L(C|A)$ of class given the acoustics is maximized, and thus the acoustic model is taken into account when making the classification. The fact that the 1-pass N-gram classifiers perform better when run on speech utterances $A$ than on transcription $W$ indicates that further improvements could be obtained this way.

## 6. Acknowledgments

The authors wish to thank Milind Mahajan, Asela Gunawardana and Julian Odell for useful discussions as well as help in bringing the ATIS system up.

## 7. References

[1] A. L. Gorin, G. Riccardi, and J. H. Wright, "How may I help you?" *Speech Communication*, vol. 23, no. 1/2, pp. 113–127, 1997.

[2] J. Chu-Carroll and B. Carpenter, "Vector-based natural language call routing," *Computational Linguistics*, vol. 25, no. 3, pp. 361–388, 1999.

[3] C. D. Manning and H. Schutze, *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts: The MIT Press, 2001.

[4] F. Jelinek and R. Mercer, "Interpolated estimation of Markov source parameters from sparse data," in *Pattern Recognition in Practice*, E. Gelsema and L. Kanal, Eds., 1980, pp. 381–397.

[5] P. S. Gopalakrishnan, D. Kanevski, A. Nadas, and D. Nahamoo, "An inequality for rational functions with applications to some statistical estimation problems," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 107–113, January 1991.

[6] C. Chelba, M. Mahajan, and A. Acero, "Speech utterance classification," in *Proceedings of ICASSP*.

[7] D. Pallet, J. Fiscus, W. Fisher, J. Garofolo, B. Lund, A. Martin, and M. Przybocki, "1993 benchmark tests for the ARPA spoken language program," in *Proceedings of the Human Language Technology Workshop*, C. Weinstein, Ed. Plainsboro, NJ: Morgan Kaufmann, March 1994.

[8] S. Young, "The HTK hidden Markov model toolkit: design and philosophy," Department of Engineering, Cambridge University, UK, Tech. Rep. TR.153, 1993.