MAXIMUM ENTROPY CONFIDENCE ESTIMATION FOR SPEECH RECOGNITION

Christopher White

Center for Language and Speech Processing
JHU, Baltimore, MD, 21218

Jasha Droppo, Alex Acero, Julian Odell

Microsoft Research
Redmond, WA 98052

ABSTRACT

For many automatic speech recognition (ASR) applications, it is useful to predict the likelihood that the recognized string contains an error. This paper explores two modifications of a classic design. First, it replaces the standard maximum likelihood classifier with a maximum entropy classifier. The maximum entropy framework carries the dual advantages discriminative training and reasonable generalization. Second, it includes a number of alternative features. Our ASR system is heavily pruned, and often produces recognition lattices with only a single path. These alternate features are meant to serve as a surrogate for the typical features that can be computed from a rich lattice. We show that the maximum entropy classifier easily outperforms the standard baseline system, and the alternative features provide consistent gains for all of our test sets.

Index Terms— Speech recognition, Speech processing, Maximum entropy methods

1. INTRODUCTION

In automatic speech recognition (ASR), confidence measures (CM) predict the reliability of the recognition result. They enable the system to discard a result, or prompt the user to repeat herself, rather than to act on an incorrect transcription.

The standard confidence estimation design consists of a classifier that predicts the probability of error using several observations taken from the recognition lattice emitted by the ASR engine.

If a rich lattice is available, it can be renormalized to provide a good confidence estimate (CE)[1, 2]. Alternately, an ASR engine can produce many types of scores which are used as observations to train a statistical model. In addition to a typical ASR observation such as acoustic score, decoders may produce a variety of observations based on the language model, articulatory observations[3] or discourse events[4].

The framework of observing events from an output (lattice or otherwise) to train a model for estimating confidence is also used in the fields of information extraction[5] and machine translation[6, 7]. The models to be trained have included Gaussian mixture models (GMM)[8], generalized linear models (GLM)[9], decision trees[10], support vector machines[11], maximum entropy (MaxEnt) trained models[12], model combination[13], or a hybrid of these[6, 7, 14]. While the recent trend has been toward discriminative systems[11, 12], many systems still train a generative model based on observations pulled from a lattice[8, 3].

This paper details two improvements over the standard design. Both are motivated by the desire to create a system that can reliably and efficiently estimate confidence for an optimized ASR engine, across utterances that vary in duration, grammar, and vocabulary.

The first improvement provides significant gains in overall accuracy, as well as good generalization behavior. This is accomplished with the introduction of a maximum entropy classifier. The references listed above provide ample motivation for selecting a discriminatively trained system, and maximum entropy has the added benefit of producing a classifier that is likely to generalize well. Producing good maximum entropy features from the raw observations is not trivial, and this paper explores the relative merits of several approaches.

The second improvement allows the system to provide good confidence estimates, even when a rich recognition lattice is not available. In a deployed system, the ASR engine may be set to heavily prune its search for the best transcription. While this makes the overall system faster, the resulting recognition lattices tend to contain only a single path. This precludes the use of many standard confidence estimation observations. The solution presented here is to produce alternate features designed to contain information similar to what has been pruned from the lattice.

This paper is organized as follows: Section 2 describes the data set which has been compiled to represent many types of relevant tasks, the baseline GMM system, and observation selection. Section 3 presents the MaxEnt systems and a comparison of different methods for generating features from the raw observations. We show that the maximum entropy classifier easily outperforms the standard baseline system, and that the alternative features provide consistent gains for all of our test sets.

2. BASELINE SYSTEM

Our baseline system consists of a Gaussian mixture model (GMM) classifier, built on a data set specifically constructed for this task.

Performance is measured by the sum of false accepts and false rejects divided by the total number of instances in the set.

2.1. The Data Set

Our goal was to build a system that generates good confidence estimates. This means that it should work transparently across a variety of recognition grammars. It should be robust to duration, speaker, channel, and other irrelevant factors. And, it should produce reliable output, even if rich lattices of the recognition result are unavailable.

To meet and prove these design requirements, we merged existing data to construct a new corpus. It contains over 250,000 utterances pulled from source corpora covering different acoustic channels, additive noise, and accents. The utterances are parsed by 280 grammars, including everything from spelling and digits to "how may I help you" style grammars.

The utterances in the corpus were divided according to Table 1. Approximately 80% of the utterances used for training, 10% for designing our system (development), and 10% were reserved for a held-out evaluation set.

Data Partition	*Utterances*	*Unq*	*Alt*
Training	199,282	157,372	41,910
Development	26,023	20,840	5,183
Evaluation	29,551	20,759	5,232

Table 1. The distribution of data used for our experiments.

Each major division of the corpus is further divided into two parts, according to the result returned by our recognition engine. If our engine returns more than one alternate hypothesis for an utterance, it is placed in the 'Alt' partition. Otherwise, there is a unique recognition result, and it belongs in the 'Unq' partition.

The data in the 'Unq' and 'Alt' partitions behave quite differently. For example, in the training set, approximately 85% of the 'Unq' examples were correct. In the 'Alt' partition, the top hypothesis was only 50% of the time. Because of this discrepancy, we build a separate classifier for each partition.

2.2. Observation Selection

The topology of the engine and the constraints of the task limit observation selection options. Due to our desire for language independence, many popular observations in the literature which depend on the lexicon are excluded. Also, we focus on observations extracted from the 1-best hypothesis for most of our experiments including many typical observations such as acoustic score. In general the model should get most of the information from core features and feature processing. The features used are listed below in Table 2, with lattice features denoted with an *, and augmented-set features denoted with a **. Features used in the 'Unq' case are denoted with a 'U', 'Alt' with a 'A'.

Observation	*Set*	*Description*
AN	AU	Acoustic Normalized
BN	AU	Background Normalized
NN	AU	Noise Normalized
AAN	AU	Arc Acoustic Normalized
LMN	AU	Language Model Normalized
DN	AU	Duration Normalized
LMP	AU	Language Model Perplexity
LMF	AU	Language Model Fanout
AS	AU	Active Senones
AC	AU	Active Channels
C	*A	First Alternate Delta AN
NB	**A	Number NBest
CN	**A	Number of Nodes
CA	**A	Number of Arcs
CB	**A	Number of Bytes
EMD	**A	E[MaxAN - MinAN] (node)

Table 2. Observations for all systems

In Table 2, some of the scores are normalized. The acoustic scores are normalized by subtracting the likelihood generated by a "best senone" model. It is the likelihood that would have been produced by the engine, if it weren't as constrained by grammar and pronunciation rules. Each normalized observation likelihood is further divided by its duration.

Most of the observations in Table 2 are self-explanatory.

The acoustic scores measure how well the acoustic data matches the grammar and acoustic model, unconstrained speech-like sounds, and noise. There is an acoustic score associated with the most likely transcription, an acoustic score generated by a monophone-loop (background) acoustic model, one generated by a noise model, and for the 'Alt' case, a measure of how different the first and second most likely hypotheses were (First Alternate Delta AN).

Observations taken from the language model are included to help the classifier adapt to different recognition grammars. There are observations taken from the language model scores, as well as observations of its perplexity and average fanout.

Some observations are included to measure the state of the recognition process itself. For instance, how hard the recognizer had to work to produce the result (active senones and channels), how many entries are in the n-best list, and the size (in nodes, arcs, and bytes) of the recognition result.

2.3. The GMM Baseline

The baseline system consists of two GMMs, one that models correctly recognized utterances (c) and one that models incorrectly recognized utterances (i). The GMM models the output distribution as being generated from a linear combination of M components. For example, the p.d.f. of the observations $\mathbf{y}$ given the class c is given by

$$p(\mathbf{y}|c) = \sum_{k=1}^{M} p(\mathbf{y}|k,c)p(k|c), \tag{1}$$

where $p(k|c)$ is the prior probability of being generated from the kth mixture component for class c, and $p(\mathbf{y}|k,c)$ has the form

$$p(\mathbf{y}|k,c) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\mu_{k,c})^T\Sigma_{k,c}^{-}1(\mathbf{x}-\mu_{k,c})\right)}{(2\pi)^{D/2}\,|\Sigma_{k,c}|^{\frac{1}{2}}} \tag{2}$$

where $\mu_{k,c}$ and $\Sigma_{k,c}$ are the mean and covariance matrix of the kth mixture component of the class c.

Both c and i models use a full covariance matrix and have been trained using the expectation maximization (EM) algorithm. The performance was insensitive to the number of mixtures. It produced reasonable results, as can be seen in the "GMM" row of Table 3.

3. MAXIMUM ENTROPY SYSTEM

This section describes several discriminative classifiers for confidence estimation. In these experiments, the classifiers are conditional models trained using the maximum entropy (MaxEnt) criterion. Conditional maximum entropy models were chosen based on their history of good performance for speech and language related tasks including language modeling[15], parsing[16], etc. They have been applied with mixed results to confidence estimation in information extraction[5] and machine translation[6].

Although MaxEnt models have been applied to estimating posterior phone probabilities during a lattice search[12], this work differs in many respects. For example, they are concerned with estimating absolute posterior probability for a lattice phone search while we are estimating a general model of error. Also, they adjust the maximum entropy criterion to approach precision on a training set using weakened constraints and features entirely based on the word lexicon.

Our model is of the form $p(y|\mathbf{x})$. Here, y is a discrete random variable representing the class 'correct' or 'incorrect', and $\mathbf{x}$ is a vector of discrete or continuous random variables. In the conditional MaxEnt framework, the model interacts with the random variables $\mathbf{x}$ and y through a vector of feature functions $f_i(\mathbf{x}, y)$ and parameters λ_i.

$$p_\Lambda(y|\mathbf{x}) = \frac{\exp\left(\sum_{i=1}^{F} \lambda_i f_i(\mathbf{x}, y)\right)}{\sum_{y'} \exp\left(\sum_{i=1}^{F} \lambda_i f_i(\mathbf{x}, y')\right)} \quad (3)$$

This conditional MaxEnt model is regularized by using a Gaussian prior on the parameters λ_i. The performance on the development data was insensitive to the variance of this prior, which is not surprising given the size of our training data set. As a result, it was fixed at a value of 100 for all of our experiments.

	Training		*Development*		*Evaluation*	
	Unq	*Alt*	*Unq*	*Alt*	*Unq*	*Alt*
GMM	-	-	10.92	21.03	10.96	21.25
11c	6.59	20.74	6.26	19.49	6.69	20.30
11b	5.61	16.83	5.86	18.77	6.06	19.55
121b	4.47	10.53	5.75	20.91	6.07	19.71
11+b	5.45	16.15	5.69	18.35	5.97	17.91

Table 3. Error rate of GMM and MaxEnt systems. The error rate is defined as the total number of false accepts and false rejects, divided by the number of training examples.

3.1. A Simple MaxEnt System

In building a MaxEnt model, the system designer is free to choose from a number of methods of generating feature functions f to represent the observations $\mathbf{x}$. For this paper, we explored four different choices, with varying levels of complexity and parameter count.

The first system, '11c' in Table 3, is meant to approximate a linear classifier. It has by far the fewest number of parameters out of any other system presented in this paper, and represents the gain that can be achieved without the more involved binning techniques presented later.

The observations for this system consist of the base set of observations which have been normalized by mean and variance. This normalization equalizes the dynamic range of each type of observation, which helps our training process to converge on a good parameter estimate.

There are four feature functions created for each dimension of the observation vector. Because our trainer does not accept negative features, we create symmetric features based on whether the original observation was positive or negative:

$$f_+ = max(x, 0) \quad (4)$$
$$f_- = max(-x, 0) \quad (5)$$

For each of these, another pair of symmetric features is created: one for the correct class, and one for the incorrect class.

After adding 1 indicator feature for each class to build a truth-based prior there are a total of 42 and 46 features for the 'Unq' and 'Alt' case respectively.

Results for this system are displayed in the "11c" row of Table 3. Even though the "11c" system has far fewer parameters than the "GMM" system, it shows a marked improvement over the baseline.

3.2. Improved Results with Binning

The second system, row '11b' in Table 3, like '11c', uses the base set of 10 and 11 observation dimensions. But, instead of using features that are linear functions of the observations, it creates a set of histogram-based binary features. As a result, they allow the model to take advantage of nonlinear relationships in the data.

These features are created by sorting each of the observation dimensions by value and creating bins based on a uniform-occupancy partitioning. [1] When an observation value falls within the range associated with one of its features, that feature is activated with a value of one. Otherwise, its value is zero.

With a maximum of 100 bins (chosen experimentally, see Section 3.5 below) and a minimum occupancy of 100, there were 2246 and 1984 binary MaxEnt feature functions for 'Alt' and 'Unq' respectively.

The '11b' row of Table 3 shows a consistent improvement over the '11c' case for both 'Unq' and 'Alt' on both dev and eval sets. We conclude that binning, which allows the MaxEnt classifier to develop a nonlinear decision surface, is preferable to the simpler system presented in Section 3.1.

3.3. Quadratic Observation Vector

The third system, $121b$, attempts to mimic the full covariance aspect of the GMM system. Instead of the base set of 10 and 11 observation dimensions, it uses the outer product consisting of 100 and 121 dimensions. After binning, with minimum and maximum occupancy set as above, there were 26,414 and 21,556 features in the two systems.

The results in row '121b' of Table 3 show no consistent improvement, which is most likely due to over-training. The marked decrease in training set error rate supports this hypothesis. Future efforts could attempt to balance the very low training error with the higher development error through different MaxEnt feature selection strategies such as pruning.

3.4. Incorporating Augmented Features

The fourth system, '11b+' in Table 3, augments the original feature set with additional lattice based observations (see Table 2).

As the quality of the observation depends on the depth and quality of the lattice, the augmented set includes mostly observations which are descriptive of the lattice as a whole rather than individual pieces or hypotheses. Most of the lattices generated by our engine on this test have a very small depth, with only 1 or 2 alternates.

After trying several non-lexical observations, those in the augmented set were found to improve the system the most on the development set. Therefore this system has 14 observation dimensions for both cases producing approximately 2800 features after binning as above.

Results in the '11b+' row of Table 3 show improvement in both cases on both the development and evaluation set. The success of this experiment highlights the potential for small but significant improvements from smart observation selection.

3.5. Feature Processing

There have been few informative methods in the literature for feature processing using MaxEnt models. At first, it is not clear that quantizing observations would help, but the results above show significant

[1] A minimum occupancy of 100 training examples per bin is enforced for all MaxEnt experiments using quantization.

improvement with uniform binning and constraints on occupancy and number of bins.

Figure 1 shows experiments on the development set designed to find the proper maximum number of bins. Both curves reach their minimum error rate on the development set near a setting of 100 bins.

To the left and right of this minimum, both systems exhibit over-training and under-parameterization, respectively. As the number of bins increases, the model over-trains, increasing the development set error, while decreasing the training set error. As the number of bins decreases, eventually the system doesn't have enough parameters to do optimal classification. As a result, both training and development set error increase.

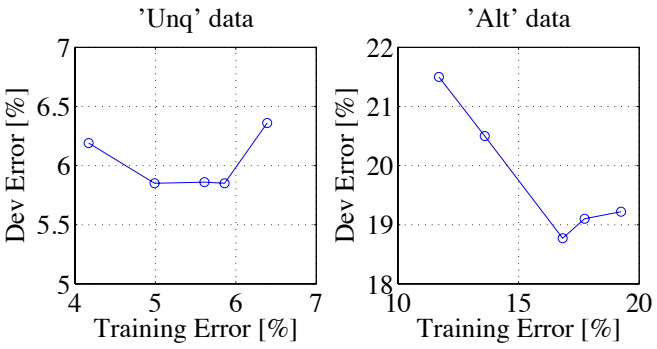

Fig. 1. Setting Maximum Bins. Experiments were run with minimum bin sizes of 10, 50, 100, 300, and 1000. Development set accuracy was minimized with 100 bins in both cases. The 'Alt' partition of the data appears to be more sensitive to over-training than the 'Unq' partition.

4. CONCLUSIONS

This paper describes how a maximum entropy model can be used to generate confidence scores for a speech recognition engine on an array of grammars. Results on an evaluation set of 25,991 examples that span 280 grammars demonstrate that the methods of observation selection, feature generation, and model training in this paper provide a significant improvement over a standard baseline. The systems might be enhanced with additional observations including those derived from the grammar, or transformations such as linear discriminant analysis (LDA). This work presents a base set of non-lexical observations principally derived from a 1-best recognition output. It also demonstrates the effectiveness of quantizing continuous observations. Finally, it outlines a successful strategy for building a confidence estimate which can work on a variety of language independent tasks even in the absence of a lattice.

5. ACKNOWLEDGEMENTS

The authors wish to thank our colleagues Milind Mahajan for his help with the conditional MaxEnt training, as well as Aslea Gunawardana and Xiaodong He for sharing their time and ideas.

This research was performed while one of the authors was on appointment as a U.S. Department of Homeland Security (DHS) Fellow under the DHS Scholarship and Fellowship Program, a program administered by the Oak Ridge Institute for Science and Education (ORISE) for DHS through an interagency agreement with the U.S Department of Energy (DOE). ORISE is managed by Oak Ridge Associated Universities under DOE contract number DE-AC05-06OR23100. All opinions expressed in this paper are the author's and do not necessarily reflect the policies and views of DHS, DOE, or ORISE.

6. REFERENCES

[1] G. Evermann and P. Woodland, "Large vocabulary decoding and confidence estimation using word posterior probabilities," in *International Conference on Acoustics, Speech, and Signal Processing*, 2000.

[2] F. Wessel, R. Schlüter, K. Macherey, and H. Ney., "Confidence measures for large vocabulary continuous speech recognition," *IEEE Trans. on Speech and Audio Processing*, vol. 9, no. 3, pp. 288–298, 2001.

[3] K.-Y. Leung and M. Sui, "Articulatory-feature-based confidence measures," *Computer Speech and Language*, 2005.

[4] T. Hazen, S. Seneff, and J. Polifroni, "Recognition confidence scoring and its use in speech understanding systems," *Computer Speech and Language*, vol. 16, pp. 49–67, 2002.

[5] A. Culotta and A. McCallum, "Confidence estimation for information extraction," in *NAACL*, 2004.

[6] Blatz et al., "Confidence estimation for machine translation," Tech. Rep., Final report, JHU/CLSP Summer Workshop, 2003.

[7] C. Quirk, "Training a sentence-level machine translation confidence measure," in *LREC*, 2004.

[8] T. Kim and H. Ko, "Bayesian confidence scoring and adaptation techniques for speech recognition," *IEEE Transactions on Communication*, vol. E88-B, no. 4, 2005.

[9] M. Siu and H. Gish, "Evaluation of word confidence for speech recognition systems," *Computer Speech and Language*, vol. 13, no. 4, pp. 299–319, 1999.

[10] J. Xue and Y. Zhao, "Random forest-based confidence annotation using novel features from confusion networks," in *International Conference on Acoustics, Speech, and Signal Processing*, 2006.

[11] D. Hillard and M. Ostendorf, "Compensating for word posterior estimation bias in confusion networks," in *International Conference on Acoustics, Speech, and Signal Processing*, 2006.

[12] P. Yu, D. Zhang, and F. Seide, "Maximum entropy based normalization of word posteriors for phonetic and lvcsr lattice search," in *International Conference on Acoustics, Speech, and Signal Processing*, 2006.

[13] A. Sankar, "Bayesian model combination (baycom) for improved recognition," in *International Conference on Acoustics, Speech, and Signal Processing*, 2005.

[14] S. Gandrabur and G. Foster, "Confidence estimation for text prediction," in *Conference on Natural Language Learning*, 2003.

[15] R. Rosenfeld, *Adaptive Statistical Language Modeling: A Maximum Entropy Approach*, Ph.D. thesis, Carnegie Mellon University, 1994.

[16] A. Ratnaparkhi, *Maximum Entropy Models for Natural Language Ambiguity Resolution*, Ph.D. thesis, University of Pennsylvania, 1998.