Estimating probability of correctness for ASR N-Best lists

Jason D. Williams and Suhrid Balakrishnan
AT&T Labs - Research, Shannon Laboratory, 180 Park Ave., Florham Park, NJ 07932, USA
{jdw,suhrid}@research.att.com

Abstract

For a spoken dialog system to make good use of a speech recognition N-Best list, it is essential to know how much trust to place in each entry. This paper presents a method for assigning a probability of correctness to each of the items on the N-Best list, and to the hypothesis that the correct answer is not on the list. We find that both multinomial logistic regression and support vector machine models yields meaningful, useful probabilities across different tasks and operating conditions.

1 Introduction

For spoken dialog systems, speech recognition errors are common, and so identifying and reducing dialog understanding errors is an important problem. One source of potentially useful information is the *N-Best list* output by the automatic speech recognition (ASR) engine. The N-Best list contains N ranked hypotheses for the user's speech, where the top entry is the engine's best hypothesis. When the top entry is incorrect, the correct entry is often contained lower down in the N-Best list. For a dialog system to make use of the N-Best list, it is useful to estimate the probability of correctness for each entry, and the probability that the correct entry is not on the list. This paper describes a way of assigning these probabilities.

2 Background and related work

To begin, we formalize the problem. The user takes a communicative action u, saying a phrase such as "Coffee shops in Madison New Jersey". Using a language model g, the speech recognition engine processes this audio and outputs an ordered list of N hypotheses for u, $\tilde{\mathbf{u}} = \{\tilde{u}_1, \ldots \tilde{u}_N\}$, $N \geq 2$. To the N-Best list we add the entry $\tilde{u}_*$, where $u = \tilde{u}_*$ indicates that u does not appear on the N-Best list.

The ASR engine also generates a set of K recognition features $\mathbf{f} = [f_1, \ldots, f_K]$. These features might include properties of the lattice, word confusion network, garbage model, etc. The aim of this paper is to estimate a *model* which accurately assigns the $N + 1$ probabilities $P(u = \tilde{u}_n | \tilde{\mathbf{u}}, \mathbf{f})$ for $n \in \{*, 1, \ldots, N\}$ given $\tilde{\mathbf{u}}$ and $\mathbf{f}$. The model also depends on the language model g, but we don't include this conditioning in our notation for clarity.

In estimating these probabilities, we are most concerned with the estimates being *well-calibrated*. This means that the probability estimates we obtain for events should accurately represent the empirically observed proportions of those events. For example, if 100 1-best recognitions are assigned a probability of 60%, then approximately 60 of those 100 should in fact be the correct result.

Recent work proposed a generative model of the N-Best list, $P(\tilde{\mathbf{u}}, \mathbf{f} | u)$ (Williams, 2008). The main motivation for computing a generative model is that it is a component of the update equation used by several statistical approaches to spoken dialog (Williams and Young, 2007). However, the difficulty with a generative model is that it must estimate a joint probability over all the features, $\mathbf{f}$; thus, making use of many features becomes problematic. As a result, discriminative approaches often yield better results. In our work, we propose a discriminative approach and focus on estimating the probabilities *conditioned on* the features. Additionally, under some further fairly mild assumptions, by applying Bayes Rule our model can be shown equivalent to the generative model required in the dialog state update. This is a desirable property because dialog systems using this re-statement have been shown to work in practice (Young et al., 2009).

Much past work has assigned meaningful proba-

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 132–135,
Queen Mary University of London, September 2009. ©2009 Association for Computational Linguistics

bilities to the top ASR hypothesis; the novelty here is assigning probabilities to *all* the entries on the list. Also, our task is different to *N-Best list re-ranking*, which seeks to move more promising entries toward the top of the list. Here we trust the ordering provided by the ASR engine, and only seek to assign meaningful probabilities to the elements.

3 Model

Our task is to estimate $P(u = \tilde{u}_n | \tilde{\mathbf{u}}, \mathbf{f})$ for $n \in \{*, 1, \ldots, N\}$. Ideally we could view each element on the N-Best list as its own class and train an $(N+1)$-class regression model. However this is difficult for two reasons. First, the number of classes is variable: ASR results can have different N-Best list lengths for different utterances. Second, we found that the distribution of items on the N-Best list has a very long tail, so it would be difficult to obtain enough data to accurately estimate late position class probabilities.

As a result, we model the probability P in two stages: first, we train a (discriminative) model P_a to assign probabilities to just three classes: $u = \tilde{u}_*$, $u = \tilde{u}_1$, and $u \in \tilde{\mathbf{u}}_{\mathbf{2}+}$, where $\tilde{\mathbf{u}}_{\mathbf{2}+} = \{\tilde{u}_2, \ldots, \tilde{u}_N\}$. In the second stage, we use a separate probability model P_b to distribute mass over the items in $\tilde{\mathbf{u}}_{\mathbf{2}+}$:

$$P(\tilde{u}_n = u | \tilde{\mathbf{u}}, \mathbf{f}) = \qquad (1)$$
$$\begin{cases} P_a(u = \tilde{u}_1 | \mathbf{f}) & \text{if } n = 1, \\ P_a(u \in \tilde{\mathbf{u}}_{\mathbf{2}+} | \mathbf{f}) P_b(u = \tilde{u}_n | \mathbf{f}) & \text{if } n > 1, \\ P_a(u = \tilde{u}_* | \mathbf{f}) & \text{if } n = * \end{cases}$$

To model P_a, multinomial logistic regression (MLR) is a natural choice as it yields a well-calibrated estimator for multi-class problems. Standard MLR can over-fit when there are many features in comparison to the number of training examples; to address this we use ridge *regularized* MLR in our experiments below (Genkin et al., 2005).

An alternative to MLR is support vector machines (SVMs). SVMs are typically formulated including regularization; however, their output scores are generally not interpretable as probabilities. Thus for P_a, we use an extension which re-scales SVM scores to yield well-calibrated probabilities (Platt, 1999).

Our second stage model P_b, distributes mass over the items in the tail of the N-best list ($n \in \{2, \ldots, N\}$). In our exploratory analysis of N-Best lists, we noticed a trend that facilitates modeling this distribution. We observed that the distribution of the *fraction* of the correction position n/N was relatively invariant to N. For example, for both short ($N < 100$) and long ($N \geq 100$) lists, the probability that the answer was in the top half of the list was very similar (see Figure 1). Thus, we chose a continuous distribution in terms of the fractional position n/N as the underlying distribution in our second stage model. Given the domain of the fractional position $[0, 1]$, we chose a Beta distribution. Our final second stage model is then an appropriately discretized version of the underlying Beta, namely, P_b:

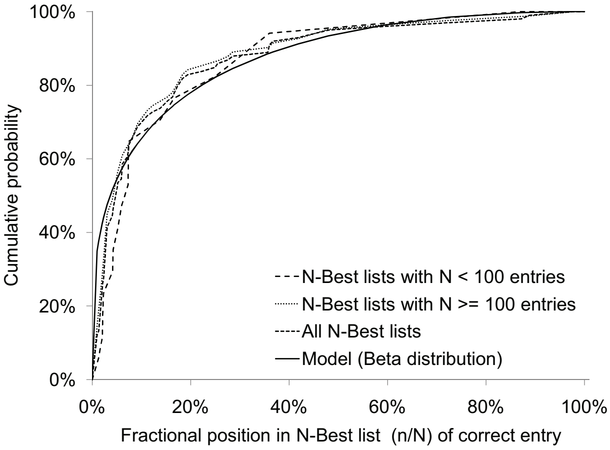

Figure 1: Empirical cumulative distribution of correct recognitions for N-Best lists, and the Beta distribution model for P_b on $1,000$ business search utterances (Corpus 1 training set, from Section 4.)

$$P_b(u = \tilde{u}_n | \mathbf{f}) = P_b(u = \tilde{u}_n | N) =$$
$$P_{\text{beta}}(\frac{n-1}{N-1}; \alpha, \beta) - P_{\text{beta}}(\frac{n-2}{N-1}; \alpha, \beta)$$

where $P_{\text{beta}}(x; \alpha, \beta)$ is the standard Beta cumulative distribution function parametrized by α and β. Figure 1 shows an illustration. In summary, our method requires training the three-class regression model P_a, and estimating the Beta distribution parameters α and β.

4 Data and experiments

We tested the method by applying it to three corpora of utterances from dialog systems in the business search domain. All utterances were from

Corpus	WCN	SVM	MLR
1	-0.714	**-0.697**	-0.703
2	-0.251	-0.264	**-0.222**
3	-0.636	-0.605	**-0.581**

Table 1: Mean log-likelihoods on the portion of the test set with the correct answer on the N-Best list. None of the MLR nor SVM results differ significantly from the WCN baseline at $p < 0.02$.[2]

Corpus	WCN	SVM	MLR
1	-1.12	**-0.882**	-0.890
2	-0.821	-0.753	**-0.734**
3	-1.00	**-0.820**	-0.824

Table 2: Mean log-likelihoods on the complete test set. All MLR and SVM results are significantly better than the WCN baseline ($p < 0.0054$).[2]

users with real information needs. Corpus 1 contained $2,000$ high-quality-audio utterances spoken by customers using the Speak4It application, a business search application which operates on mobile devices, supporting queries containing a listing name and optionally a location.[1] Corpus 2 and 3 contained telephone-quality-audio utterances from $14,000$ calls to AT&T's "411" business directory listing service. Corpus 2 contained locations (responses to "Say a city and state"); corpus 3 contained listing names (responses to "OK what listing?"). Corpus 1 was split in half for training and testing; corpora 2 and 3 were split into $10,000$ training and $4,000$ testing utterances.

We performed recognition using the Watson speech recognition engine (Goffin et al., 2005), in two configurations. Configuration A uses a statistical language model trained to recognize business listings and optionally locations, and acoustic models for high-quality audio. Configuration B uses a rule-based language model consisting of all city/state pairs in the USA, and acoustic models for telephone-quality audio. Configuration A was applied to corpora 1 and 3, and Configuration B was applied to corpus 2. This experimental design is intended to test our method on both rule-based and statistical language models, as well as matched and mis-matched acoustic and language model conditions.

We used the following recognition features in $\mathbf{f}$: f_1 is the posterior probability from the best path through the word confusion network, f_2 is the number of segments in the word confusion network, f_3 is the length of the N-Best list, f_4 is the average per-frame difference in likelihood between the highest-likelihood lattice path and a garbage model, and f_5 is the average per-frame difference in likelihood between the highest-likelihood lattice path and the maximum likelihood of that frame on *any* path through the lattice. Features are standardized to the range $[-1, 1]$ and MLR and SVM hyperparameters were fit by cross-validation on the training set. The α and β parameters were fit by maximum likelihood on the training set.

We used the BMR toolkit for regularized multinomial logistic regression (Genkin et al., 2005), and the LIB-SVM toolkit for calibrated SVMs (Chang and Lin, 2001).

We first measure average log-likelihood the models assign to the test sets. As a baseline, we use the posterior probability estimated by the word confusion network (WCN), which has been used in past work for estimating likelihood of N-Best list entries (Young et al., 2009). However, the WCN does not assign probability to the $u = \tilde{u}_*$ case – indeed, this is a limitation of using WCN posteriors. So we reported two sets of results. In Table 1, we report the average log-likelihood given that the correct result is on the N-Best list (higher values, i.e., closer to zero are better). This table includes only the items in the test set for which the correct result appeared on the N-Best list (that is, excluding the $u = \tilde{u}_*$ cases). This table compares our models to WCNs on the task for which the WCN is designed. On this task, the MLR and SVM methods are competitive with WCNs, but not significantly better.

In Table 2, we report average log-likelihood for the entire test set. Here the WCNs use a fixed prior for the $u = \tilde{u}_*$ case, estimated on the training sets ($u = \tilde{u}_*$ class is always assigned 0.284; other classes are assigned $1 - 0.284 = 0.716$ times the WCN posterior). This table compares our models to WCNs on the task for which our model is designed. Here, the MLR and SVM models yielded

[1] `http://speak4it.com`

[2] 2-tailed Wilcoxon Signed-Rank Test; 10-way partitioning.

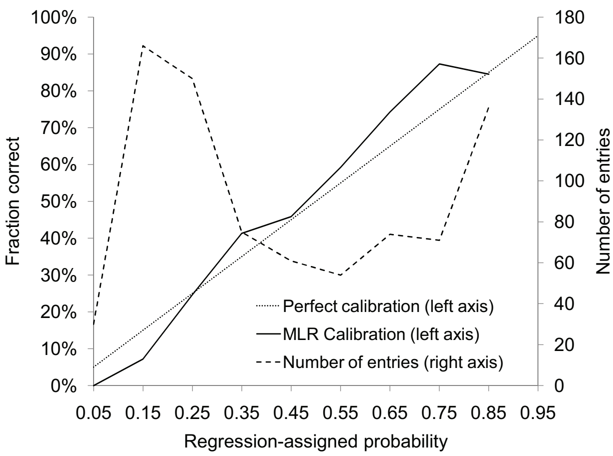

Figure 2: Calibration and histogram of probabilities assigned by MLR on corpus 1 (test set).

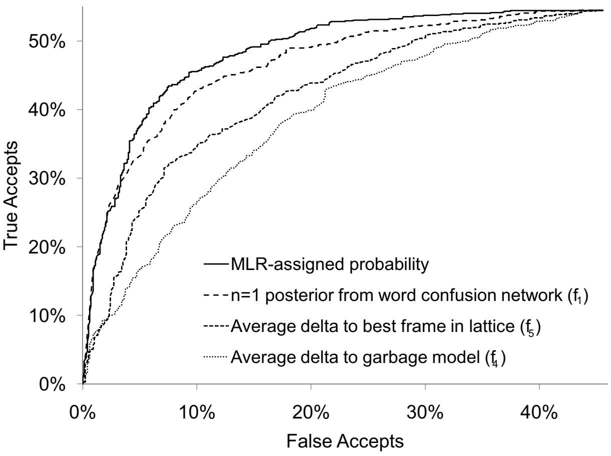

Figure 3: ROC curve for MLR and the 3 most informative input features on corpus 1 (test set).

significantly better results than the WCN baseline.

We next investigated the calibration properties of the models. Results for the MLR model on the $u = \tilde{u}_1$ class from corpus 1 test set are shown in Figure 2. This illustrates that the MLR model is relatively well-calibrated and yields broadly distributed probabilities. Results for the SVM were similar, and are omitted for space.

Finally we investigated whether the models yielded better accept/reject decisions than their individual features. Figure 3 shows the MLR model a receiver operating characteristic (ROC) curve for corpus 1 test set for the $u = \tilde{u}_1$ class. This confirms that the MLR model produces more accurate accept/reject decisions than the individual features alone. Results for the SVM were similar.

5 Conclusions

This paper has presented a method for assigning useful, meaningful probabilities to elements on an ASR N-Best list. Multinomial logistic regression (MLR) and support vector machines (SVMs) have been tested, and both produce significantly better models than a word confusion network baseline, as measured by average log likelihood. Further, the models appear to be well-calibrated and yield a better indication of correctness than any of its input features individually.

In dialog systems, we are often more interested in the concepts than specific words, so in future work, we hope to assign probabilities to concepts. In the meantime, we are applying the method to our dialog systems, to verify their usefulness in practice.

References

CC Chang and CJ Lin, 2001. *LIBSVM: a library for support vector machines*. `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

A Genkin, DD Lewis, and D Madigan, 2005. *BMR: Bayesian Multinomial Regression Software*. `http://www.stat.rutgers.edu/~madigan/BMR/`.

V Goffin, C Allauzen, E Bocchieri, D Hakkani-Tur, A Ljolje, S Parthasarathy, M Rahim, G Riccardi, and M Saraclar. 2005. The AT&T Watson speech recognizer. In *Proc ICASSP, Philadelphia*.

JC Platt. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74. MIT Press.

JD Williams and SJ Young. 2007. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech and Language*, 21(2):393–422.

JD Williams. 2008. Exploiting the ASR N-best by tracking multiple dialog state hypotheses. In *Proc ICSLP, Brisbane*.

SJ Young, M Gašić, S Keizer, F Mairesse, J Schatzmann, B Thomson, and K Yu. 2009. The hidden information state model: a practical framework for POMDP-based spoken dialogue management. *Computer Speech and Language*. To appear.