Don't Multiply Lightly: Quantifying Problems with the Acoustic Model Assumptions in Speech Recognition

Dan Gillick[1], Larry Gillick[2], Steven Wegmann[1,3]

[1] *International Computer Science Institute, Berkeley, CA, USA*
[2] *EnglishCentral, Inc., Arlington, MA, USA*
[3] *Cisco Systems, Inc., San Jose, CA, USA*

{dgillick, swegmann}@icsi.berkeley.edu, lsgillick@gmail.com

***Abstract*—We describe a series of experiments simulating data from the standard Hidden Markov Model (HMM) framework used for speech recognition. Starting with a set of test transcriptions, we begin by simulating every step of the generative process. In each subsequent experiment, we substitute a real component for a simulated component (real state durations rather than simulating from the transition models, for example), and compare the word error rates of the resulting data, thus quantifying the relative costs of each modeling assumption. A novel sampling process allows us to test the independence assumptions of the HMM, which appear to present far more serious problems than the other data/model mismatches.**

I. Introduction

There are two main failures of the standard Hidden Markov Model (see Figure I) to model speech data: the form of the output distribution and the statistical independence assumptions.

Nearly all HMM-based speech recognition systems model speech frames, the output of the underlying or *hidden* states, with a Gaussian Mixture Model (GMM), typically using diagonal covariance matrices. Much of the progress in recognition accuracy over the last twenty years is typically attributed to improving the accuracy of this approximation: adding mixture components, compensating for the diagonal covariance assumption [1], employing "discriminative" estimation criteria [2], and training with larger datasets [3], for example.

The HMM structure also induces a set of assumptions based on the conditional independence of the outputs. Specifically, the model assumes that conditional on a state, the frames in that state segment are independent of each other and independent of frames generated by other states. For example, in Figure I, conditional on the value of state s_5, output frame o_5 is independent of every other output frame. Such strong independence claims are clearly false, partly because of the mechanics of speech production, and partly because of the way speech features are computed (for example, 10 ms frames are computed from a 25 ms window; delta and double-delta features—literally functions of features in adjacent frames—are tacked on to the standard features). However, attempts to relax the HMM's independence assumptions [4], [5] or replace the HMM with a model that captures more temporal structure [6], [7], [8], [9] have met with limited success and have not been adopted in practice.

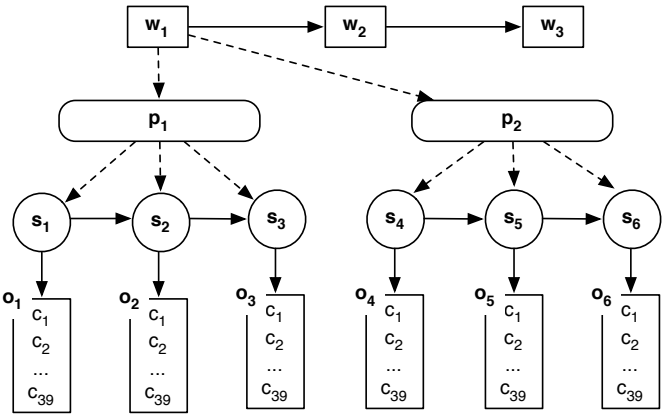

Fig. 1. A depiction of the standard generative speech model. An HMM models the continuous output distribution $P(o_t|s_t)$ with a GMM and the discrete transition distribution $P(s_t|s_{t-1})$. Each phoneme p is modeled by an HMM, typically with three sub-phoneme states, and each word w consists of a sequence of phonemes; these pronunciations are derived from a dictionary.

Given this record, we might be tempted to conclude that properly modeling the output distribution is more important than addressing the broken independence assumptions. The goal of this paper is to quantify losses incurred by making these standard modeling assumptions. Of course, there are other failings of speech recognition to model speech data, including shortcomings of the front-end, the language model, and the pronunciation dictionary. We do not intend to suggest these are minor issues, but choose to focus specifically on the statistical properties of the standard acoustic model.

Why are we doing this? First, we want to better understand the failings of the model. Second, we found many of the results surprising—for example, the effect of dependence is much stronger than we expected—and may suggest ways to refocus

future research.

Inspired by results reported in [10], we describe a series of *simulation* and *resampling* experiments—using the model to create alternate versions of a test set and then re-running recognition—to measure the impact of each assumption. Section II outlines the data and models we built for the series of experiments described in Section III; we conclude with a discussion in Section IV.

II. Data and models

We show experimental results on both Wall Street Journal (WSJ), carefully read news reports in controlled quiet conditions, and Switchboard (SWB), spontaneous telephone conversations in uncontrolled environments. The training data include 66 hours of the WSJ SI-200 dataset and 300 hours of Switchboard I. To ensure the purity of our results, we split each dataset into two speaker-disjoint pieces, and train a simulation model and a recognition model.

For WSJ experiments, we use the standard 5k bigram language model created at Lincoln Labs for the 1992 evaluation. For SWB experiments, we build a standard 20k trigram language model with absolute discounting using SRILM [12] from all the utterances used to build the recognition acoustic model. We use version 0.6 of the CMU pronunciation dictionary (stress removed) for both WSJ and SWB models.

The WSJ test set consists of the 1992 ARPA evaluation set (often called Nov-92) and a set referred to as si_dt_05.odd in [13]. The SWB test set is assembled from all the speakers in Switchboard I who participated in only one call (these conversations are removed from the training sets). To avoid out-of-vocabulary issues, all test utterances containing some word not present in the relevant language model are removed. Statistics of the training and test sets are given in Table I.

Dataset	Speakers	Utterances	Words	Hours
WSJ sim	100	13,857	249,557	32
WSJ rec	100	14,852	250,904	32
SWB sim	256	105,629	1,366,704	135
SWB rec	255	100,750	1,343,286	132
WSJ test	18	576	9,381	1.2
SWB test	23	954	10,727	1.1

TABLE I
Training set (top) and test set (bottom) statistics.

A. Training

We use version 3.4 of the HTK toolkit to train and test our models [14]. In particular, we use the standard HTK front-end to produce a 39 dimensional feature vector every 10 ms: 13 Mel-cepstral coefficients, including energy, plus their first and second differences. The cepstral coefficients are mean-normalized at the utterance level.

The acoustic models are cross-word triphones estimated with maximum likelihood[1]. Except for silence, each triphone is modeled using a three-state HMM with a discrete linear transition structure that prevents skipping. The resulting triphone states are clustered using decision trees to 2500 tied states for WSJ models and 5000 tied states for SWB models. The output distribution for each tied state is a single, multivariate Gaussian with diagonal covariance.

While significantly better performance can be achieved with mixtures of more components (16 components are sufficient for WSJ; 32 for SWB), the simplicity of a single component is preferable for our analysis; it also helps highlight the performance differences between our experiments.

[1]The model training procedure roughly follows the HTK tutorial: we estimate monophone models from a "flat start", then duplicate these to form triphone models, cluster, and re-estimate.

B. Decoding

We use HTK's HDecode for decoding, employing a wide search beam (300), a word-insertion penalty of -4, and the language model scale factor set to 15. Recall that the decoder is trying to find the sequence of words $\mathbf{w}$ with the largest score (log probability) according to the model:

$$\operatorname*{argmax}_{\mathbf{w}} \log P(\mathbf{o}|\mathbf{w}) + \kappa \log P(\mathbf{w})$$

Due to incorrect model assumptions, the two components, the acoustic model $P(\mathbf{o}|\mathbf{w})$, and the language model $P(\mathbf{w})$, do not produce scores on the same scale. The language model scale factor κ increases the weight of the language model with respect to the acoustic model, a kind of coarse compensation. When we dramatically improve the acoustic model's fit to the data, the best scale factor is much smaller. The first three lines of Table II (below) use a language model scale factor of 4.

III. Experiments

The standard acoustic model for speech recognition is *generative*: it describes a process by which acoustic features arise from a sequence of words. As a result, we can follow the generative process to simulate data:

1) Start with the test transcriptions.
2) Look up each word in the pronunciation dictionary to create phoneme-level transcriptions.
3) Simulate the sequence of subphone states and their durations for each triphone from the appropriate HMM transition model.
4) Simulate pseudo speech frames matching the duration of each state from the appropriate HMM emission model.

In the following sections, we'll use the simulation model to create pseudo speech data in accordance with this generative story, and then decode it using the recognition model. With each subsequent experiment, we'll replace a simulated component with the corresponding component from the actual test data. Recall that the simulation models are used to create the simulated test data, so simulation is done completely independently from recognition. Refer to table II for the resulting WERs; each line is annotated with the section label (A-F) in which the experiment is described. Because random sampling is involved, we repeat each experiment 5 times and report mean WER along with a standard error (SE).

Before describing the results in detail, it is important to reflect on what kind of experiments we are doing, as they

Test	WSJ test WER	SE	SWB test WER	SE	Model assumptions satisfied $P(s_i\|s_{i-1}) \sim T$	$P(o) \sim E$	$o_i \perp\!\!\!\perp o_j \| s$
(A) Simulate from T and E	0.2	0.01	2.4	0.04	yes	yes	yes
(B) Simulate from E	0.3	0.01	3.0	0.05	no	yes	yes
(C) Resample frames	0.5	0.02	4.5	0.09	no	no	yes
(D) Resample states	2.8	0.06	28.2	0.12	no	no	with i and j in different states
(E) Resample phonemes	6.2	0.10	42.1	0.19	no	no	with i and j in different phonemes
(F) Resample words	13.6	0.10	56.4	0.22	no	no	with i and j in different words
Original data	15.2		61.5		no	no	no

TABLE II
ALL SIMULATION RESULTS; EXPERIMENTS ARE ANNOTATED WITH THE SECTION IN WHICH THEY ARE DESCRIBED. T AND E REFER TO THE STANDARD HMM TRANSITION AND EMISSION MODELS.

are a bit unusual. Speech research usually involves one of two sorts of experiments: (1) improvement due to a new method is measured by WER, or (2) a "cheating" experiment that addresses how much WER could improve if only some unknown variable were revealed (e.g. What if we could always re-rank an n-best list of hypotheses perfectly?). This work does not fit into either category. We are creating artificial speech data and using WER as a familiar metric to compare the effects of the model assumptions. Thus, the results are purely diagnostic and should not be misconstrued as remedies.

A. Simulating transitions and emissions

In our first experiment, we'll follow the full generative process (above) so that the simulated data matches all the assumptions of the model: The state durations and outputs respect the model transition and emission distributions, and the frames are conditionally independent because they are simulated only with respect to their generating state.

Note that we are simulating a 39-dimensional frame that, in real data, included delta and double-delta features computed from a local window. When we simulate from the model, the structural dependence in these delta features is ignored. This is intentional: the Gaussian distribution used to model speech frames knows nothing about delta features, and the typical diagonal covariance matrix further assumes that each feature is independent. Simulation shows just how extreme the mismatch is between real data and the standard model.

So how well does recognition work if the data completely respects the model assumptions? It turns out that even if each state is modeled by just one diagonal covariance Gaussian, the WER of simulated data is extremely low: 0.2 for WSJ and 2.4 for SWB, and most of these errors are homophones (see Table III). If only real speech data matched the HMM assumptions, recognition would work very well, even with very limited training data.

How many of these errors are due to mismatch between the simulation model and the recognition model? If we use the recognition model for simulation, the WER for the SWB test drops to 1.5; if, alternatively, we force the simulation model to use the same decision tree (for state tying) as the recognition model, and run the rest of the training process as usual, the WER again drops to 1.5. These results show that while there is indeed a difference between the two models, it is mostly due to variability in the decision tree clustering; holding the decision tree constant, the Gaussian parameters of the models are well-estimated.

Confusion	Count	Confusion	Count
uh ==> a	22	than ==> then	4
a ==> uh	17	huh ==> oh	3
uh ==> the	7	to ==> into	3
too ==> to	5	two ==> to	3
are ==> our	4	wines ==> wine	3

TABLE III
THE MOST COMMONLY CONFUSED WORD PAIRS IN THE FULLY SIMULATED SWB TEST SET.

B. Simulating emissions

Next, we'd like to replace the simulated state durations with real durations to measure the mismatch between the transition model and actual speech data. We do this by deriving a state-level forced alignment of the real test data using the simulation model. Now we have the simulation model's best guess of the test data's actual state sequence. As before, we simulate speech frames from the appropriate state models, thus creating pseudo utterances that are now exactly as long as the original test utterances.

The WERs for both WSJ and SWB tests increase by about 25%. What's wrong with the transition models? The most obvious issue is that speaking rate differs substantially for different speakers, but the transition models do not reflect this variability—a kind of speaker dependence. Figure 2 shows that the number of frames per phoneme varies between 7 and 13 for the 23 speakers in the SWB test set. In addition, the per-speaker standard deviation of this speaking rate varies between 6 and 14, a measure of speaker "burstiness".

To test whether these factors account for the gap between simulated and real durations, we estimate a linear regression model. Let Y be a vector containing the ratio $^{\textit{simulated duration WER}}/_{\textit{real duration WER}}$ for each speaker s in the test set. Usually, $Y_s < 1$ because the simulated duration WER tends to be lower. Let M be a vector of the corresponding mean speaking rates, measured in frames per phoneme, and S be a vector of the corresponding log standard deviations. We set up a regression of the form:

$$Y = \alpha + \beta_1 M + \beta_2 S + \epsilon$$

We find that β_1 has a highly significant ($p = 0.007$) positive coefficient: Slower speech makes recognition easier—more frames provide more information—so the fastest speakers benefit from simulating state durations while the slowest speakers are better off with their original durations. In addition, β_2 has a marginally significant ($p = 0.08$) negative coefficient: More variability in speaking rate tends to make recognition with real state durations harder relative to simulated durations. Together, these variables explain nearly 40% of the variance ($R^2 = 0.39$) in Y.

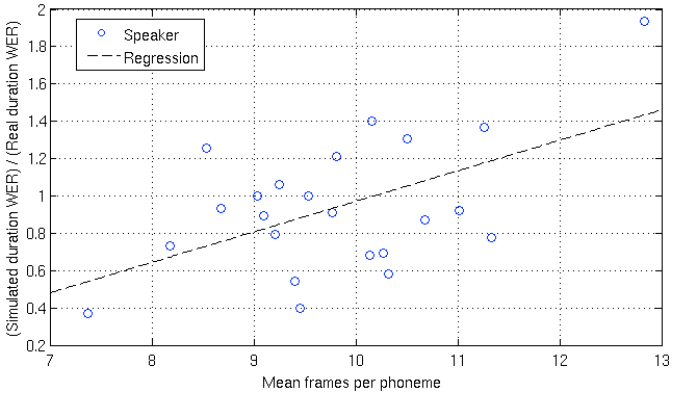

Fig. 2. In the SWB test data, there is a strong relationship between speaking rate, measured in frames per phoneme, and the ratio of per-speaker WER for simulated state durations and real state durations.

C. Resampling frames

The next model assumption we want to address is the output distribution specified by the emission models. Certainly, real speech data is not well modeled by a single diagonal Gaussian per state. To help understand how harmful this assumption is, we'd like to replace the simulated frames with real frames, but keeping intact the independence among frames.

To create data that matches the output distribution of real data but respects the model's independence assumptions, we introduce the notion of *resampling*, an application of Bradley Efron's work on Bootstrapping [11]. Rather than simulate a pseudo frame directly from the appropriate emission model, we draw an actual speech frame from an urn filled with examples of the relevant state. Figure 3 diagrams the resampling process.

Specifically, we create a state-level Viterbi alignment of the data used to train the simulation model. That is, given the trained simulation model and the correct transcriptions of the simulation data, we output the model state most likely to have generated each frame. Then we step through this alignment, placing each frame from the training data in an urn corresponding to its most likely generating state. At the end of this process, each urn represents a sample of the actual distribution of speech frames assigned to each state. Of course, since there is a limited quantity of training data, the number of representative examples in each urn varies considerably.

Resampling is the process of drawing a frame at random (with replacement) from the appropriate urn instead of simulating it from the emission model. As in the previous section, we start with the state-level Viterbi alignment of the test data (created using the simulation model). As before, we walk through the alignment one frame at a time. But now, rather than simulating each frame from its aligned state's model, we draw a sample from the urn labeled with the aligned state.

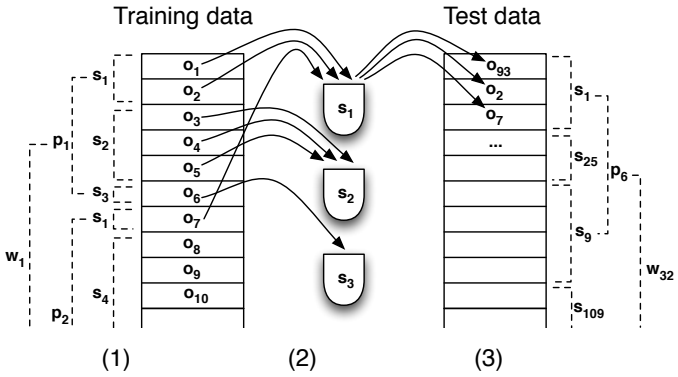

Fig. 3. A depiction of the frame-level resampling process. (1) represents the Viterbi time alignment of the simulation model's data, (2) shows how frames are placed into urns, here based on their state identity, and (3) shows the creation of resampled test data by drawing at random (with replacement) from the appropriate urn.

Resampling is a non-parametric analog to simulation. As in simulation, the independence assumptions of the HMM are satisfied by construction of the data: each frame is drawn randomly, that is, without respect to any previously drawn frame. However, while simulation creates frames that match the Gaussian model assumption, resampling results in frames matching the output distribution of real data. Note that this sort of resampling draws adjacent frames from different contexts and different speakers, both of which are sources of dependence in real data.

Perhaps the most startling result in this study is that resampling at the frame level gives WERs almost as low as simulation. This suggests that real data's violation of the model's independence assumptions is a far more serious problem than the mismatch between the output distribution and a single diagonal Gaussian. Quantitatively, creating data that satisfies the independence assumptions improves the SWB test WER from 61.5 to 4.5 (93% improvement); creating data that additionally fits the model's output distribution improves the WER from 61.5 to 3.0 (95% improvement). The relative improvements are 97% (resampling) and 98% (simulation) for the WSJ test set.

D. Resampling states

By simply changing the way we populate the urns, we can create resampled data that is locally dependent, but has longer-range independence. We'll start by placing full state segments (sequences of frames in the simulation model's data) in the urns—on average, 2-3 frames in length for non-silence states. In Figure 3, this would involve inserting the sequences (o_1, o_2) and (o_7) into urn s_1, (o_3, o_4, o_5) into urn s_2, and so on. Then to resample, we draw full state segments from the

urns, again following the state sequence determined by forced alignment of the test data (note that the durations in the forced alignments are no longer relevant and the resulting resampled data will likely be of a different length than the original). This resampled data is dependent within state regions but independent across states.

This shift from frame-level independence to state-level independence causes the WERs to jump by nearly a factor of 6. Thus the model assumption, that frames within a particular state are independent, is particularly problematic. Still, a considerable gap remains with respect to the original test data, so dependence across state boundaries is a serious issue.

E. Resampling phonemes

We can extend the reach of the data dependencies to the phoneme level by resampling phonemes (the triphone context is included in the urn labels). In Figure 3, this would involve inserting the sequence $(o_1, o_2, ..., o_7)$ into an urn labeled "$silence - p_1 + p_2$" (assuming that silence precedes phoneme p_1 in this example). Now the average length of a resampled unit is about 7-8 frames, and the WERs increase by a factor of 2.2 for WSJ data and 1.5 for SWB data. Since the error rates are still considerably lower than the original test data, cross-phoneme dependence is a significant source of mismatch.

F. Resampling words

When we further increase the size of the resampled segments to words (an average of 21-25 frames), we include the triphone context (one phoneme on each side) to respect the structure of the cross-word triphone models. For example, the word-level urns have labels like "n-THE-f" or "er-HUNDRED-ae". This makes for a very large number of urns, many of which have only one example. A few sparsely populated urns do not present a problem for resampling—subsequent words in the resampled test utterances are just as likely to be unrelated (thus preserving independence) as they would be if these rare words had more support in the urns. However, nearly a fifth of the SWB test word-level resamples and nearly a third of the WSJ test resamples turn up empty urns. In these cases, we simply keep the original segment. Still, an asymptotic analysis—using increasingly large fractions of the simulation data to populate the urns—suggests that the WERs reported in Table II for resampling words are probably nearly as low as we could expect if we had infinite simulation data.

When we resample at the word level, the WER almost returns to the rate of the original test data. This is reassuring: as we allow more and more structure from the test data into our simulation, the resulting WER increases; when we resample whole words, almost all of the important structure has returned and the WER parallels the WER of the original. So while local dependence proves quite problematic, dependence across word boundaries is of relatively little consequence.

G. Speaker dependence

In the frame resampling experiments described above (Section III-C), neighboring resampled frames tend to be uncorrelated for two main reasons. First, they probably come from different contexts—acoustic or linguistic. Second, they probably come from different speakers. If we restrict the collection of frame samples to a single training speaker, and then resample the test data as before, we can start to get a sense for how much of the data/model mismatch is due to each sort of dependence (contextual or speaker). In particular, if the WER of single-speaker resampled test data is close to the WER of the original test data, then there is little hope for improving speech recognition by modeling the temporal dependence of frames. On the other hand, if there is a relatively small increase in WER over the multi-speaker resampled test data, then the mismatch between real data and the model is probably due to the model's failure to capture temporal dependence.

We chose 10 different speakers in the SWB simulation dataset (who participated in the most conversations), and performed this single-speaker resampling experiment using each one[2]. Figure 4 shows that while the resulting WERs vary considerably, the median is just below 8, still quite small relative to the WER of the original (61.5). The variability is not particularly surprising; after all, some speakers have much lower WERs than others. But the small WER gap that separates the case where the urns are filled using all speakers from the case where the urns are filled using a single speaker suggests that the model's failure to capture speaker-based dependence is of less concern than its failure to capture contextual dependence.

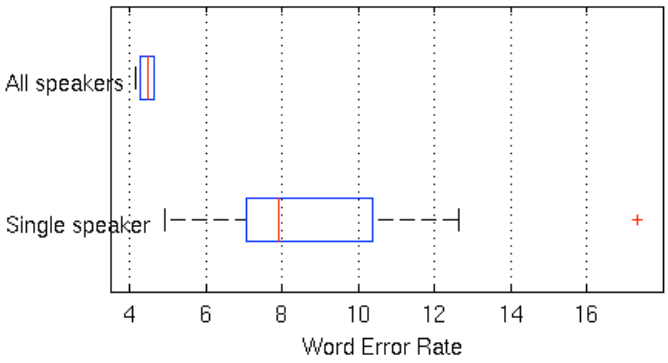

Fig. 4. Comparing WERs for two kinds of frame-resampling experiments. Urns are filled with frames from all training speakers (above) and urns are filled with frames from a specific speaker (below). The red lines indicate the median WERs.

IV. Discussion

Let's review the sequence of experiments. Simulating state durations from the transition models, and frames from the emission models, results in very low WERs for both WSJ and SWB test sets. WERs increase by 25% when we instead use real state durations derived from a forced alignment of the test data (forced alignment is done by the simulation model).

[2]We did not repeat this experiment for WSJ because the simulation error rates are so low that it is hard to compare results.

This change is due in part to speaking rate variability in real data that is not captured by the transition models.

Next, we resampled frames—drawing actual speech frames from an urn rather than simulating from the emission models—so that the output distributions match the output distributions of real speech. This increases the WERs by a factor of about 1.5. Presumably this factor would decrease if, for example, we modeled states with full rather than diagonal covariance Gaussians, or the traditional mixture of Gaussians.

Next we resampled whole states—drawing entire state sequences rather than each frame independently—and the WERs increased by a factor of nearly 6. Resampling phonemes and then finally words showed dramatic degradation in WER as well, though unequally for WSJ and SWB. Perhaps this divergent behavior is due to differences in the datasets: As Table IV shows, SWB exhibits 6% faster speaking rates than WSJ (measured in frames per phoneme). In addition, SWB tends to have shorter words (2.9 phonemes per word for SWB versus 3.2 phonemes per word for WSJ).

	WSJ test		SWB test	
Unit length (frames)	**Mean**	**Std.**	**Mean**	**Std.**
States	2.8	2.1	2.7	3.3
Phonemes	7.6	4.0	7.2	4.6
Words	24.5	17.5	21.2	17.9

TABLE IV
MEAN AND STANDARD DEVIATION OF THE NUMBER OF FRAMES PER STATE, PHONEME, AND WORD FOR THE WSJ AND SWB TEST SETS (NON-SILENCE DATA ONLY).

Finally, we returned to resampling frames, but filled the urns using a single speaker. This tended to increase the WER relative to the case where the urns were filled using many speakers, but by less than a factor of 2.

Together, this collection of experiments shows, at least for SWB and WSJ, that the mismatch between real speech data and standard models (1) is quite substantial, (2) is mostly the result of dependence among output frames, even across state and phoneme boundaries, and (3) has more to do with temporal dependence than speaker-related dependence.

Let's be more concrete. The HMM treats each output frame as a piece of independent evidence. For example, consider a situation where the decoder is trying to discern the word "she" from the competing hypothesis "sea" where the "sh" phoneme spans five very similar (and thus highly dependent) frames. Since these individual observations may be close to the decision boundary between "sh" and "s" states, the decoder ought to remain uncertain about the correct state identity, but in multiplying together essentially the same probability five times, it arrives at a dramatically over-confident result. We would expect this problem to affect long vowels in particular like "ey" (as in "ate") or "iy" (as in "eat").

This work is the beginning of an attempt to understand precisely where the standard model assumptions deviate from real speech data. While previous work on segmental models may have been headed in the right direction, perhaps it was cut short because it took a top-down, rather than a bottom-up approach to addressing temporal dependence. That is, while much segmental model research started by hypothesizing a new model for speech recognition and attempted to fit it to data, we feel it is prudent to start with the data. What kinds of dependence cause the most difficultly for current HMM systems? And, how have current methods explicitly or inadvertently helped address such dependence?

Regardless of what we find, it seems fitting to close with this moral, borrowed from William Kruskal's 1988 address to the American Statistical Association [15]: *Do not multiply lightly*.

V. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. IIS-1015930.

REFERENCES

[1] Gales, M.J.F., "Semi-Tied Covariance Matrices For Hidden Markov Models", IEEE Transactions on Speech and Audio Processing, **7**:272–281, 1999.

[2] Woodland, P. C., and Povey, D., "Large scale discriminative training of hidden Markov models for speech recognition", Computer Speech and Language, **16**(1):25–47, 2002.

[3] Chen, S., Kingsbury, B., Mangu, L., Povey, D., Saon, G., Soltau, H., and Zweig, G., "Advances in speech transcription at IBM under the DARPA EARS program", IEEE Transactions on Audio, Speech, and Language Processing, **14**(5):1596–1608, 2006.

[4] Russell, M., "A segmental HMM for speech pattern matching", Proceedings of ICASSP, 499–502, 1993.

[5] Gales, M. J. F., and Young, S. J., "Segmental HMM's for speech recognition", Proceedings of EUROSPEECH, 1579–1582, 1993.

[6] Ostendorf, M., Digalakis, V., and Kimball, O. A., "From HMM's to segment models: a unified view of stochastic modeling for speech recognition", IEEE Transactions on Speech and Audio Processing, **4**(5):360–378, 1996.

[7] Bilmes, J. A., "Buried Markov models: a graphical-modeling approach to automatic speech recognition", Computer Speech and Language, **17**:(2-3):213–231, 2003.

[8] Gunawardana, A., Mahajan, M., Acero, A., and Platt, J. C., "Hidden conditional random fields for phone classification", Proceedings of INTERSPEECH, 1117–1120, 2005.

[9] Kuo, H.–K. J., and Gao, Y., "Maximum entropy direct models for speech recognition", IEEE Transactions on Audio, Speech, and Language Processing, **14**(3):873–881, 2006.

[10] McAllaster, D., Gillick, L., Scattone, F., and Newman, M., "Studies with fabricated Switchboard data: exploring sources of model-data mismatch", Proceedings of DARPA Broadcast News Transcription and Understanding Workshop, 306–310, 1998.

[11] Efron, B., "The Jackknife, the Bootstrap and Other Resampling Plans", SIAM CBMS-NSF Monographs, 38, 1983.

[12] Stolcke, A., "SRILM-an extensible language modeling toolkit", Proceedings of ICSLP, 901–904, 2002.

[13] Woodland, P. C., Odell, J. J., Valtchev, V., and Young, S. J., "Large vocabulary continuous speech recognition using HTK", Proceedings of ICASSP, 306-310, 1994.

[14] Young, S. J., Evermann, G., Gales, M. J. F., Kershaw, D., Moore, G., Odell, J. J., Ollason, D. G., Povey, D., Valtchev, V., and Woodland,. P. C., "The HTK Book Version 3.4", Manual, Cambridge University Engineering Department, 2006.

[15] Kruskal, W., "Miracles and statistics: The casual assumption of independence", Journal of the American Statistical Association, **83**(404):929–940, 1988.