UNCERTAINTY IN TRAINING LARGE VOCABULARY SPEECH RECOGNIZERS

*Amarnag Subramanya, Chris Bartels, Jeff Bilmes and Patrick Nguyen**

Department of Electrical Engineering University of Washington Seattle, WA 98195-2500
*Speech Research Group, Microsoft Research, Redmond, WA 98052.

ABSTRACT

We propose a technique for annotating data used to train a speech recognizer. The proposed scheme is based on labeling only a single frame for every word in the training set. We make use of the virtual evidence (VE) framework within a graphical model to take advantage of such data. We apply this approach to a large vocabulary speech recognition task, and show that our VE-based training scheme can improve over the performance of a system trained using sequence labeled data by 2.8% and 2.1% on the dev01 and eval01 sets respectively. Annotating data in the proposed scheme is not significantly slower than sequence labeling. We present timing results showing that training using the proposed approach is about 10 times faster than training using sequence labeled data while using only about 75% of the memory.

1. INTRODUCTION

One of the obstacles to large scale adoption of speech recognition technology is lack of robustness in current state-of-the-art speech recognizers. In order for recognizers to be practical, it is important that they are robust towards various types of noise, speaker specific variations, changes in recording device setting, etc. One of the simplest ways of building robustness into a speech recognition system is to increase the amount of training data. Today state-of-the-art speech recognizers use thousands of hours of training data, collected from a large number of speakers with various backgrounds [1]. Yet another way to build robustness into a recognition system is to train it on hand-transcribed data with all appropriate word level segmentations (i.e. the exact time of the word boundaries are given). In [2], we showed that phone recognition systems can benefit from being trained on such data. However in the case of LVCSR systems, such segmentations are extremely hard to get and thus training using sequence labeled data (see below) been used extensively.

There are three ways to annotate data used to train a speech recognizer in a non unsupervised fashion: (a) fully-labeled (FL): all appropriate word level time segmentations (i.e., all word boundary points) are known, (b) sequence labeled (SL): only the sequence of words in an utterance is given, which implies their segmentations are unknown during training, and (c) a technique introduced by us in [2], which we call partially-labeled (PL): in addition to the word sequence, we also know the word identity of at least one frame (the acoustic observation) that was produced by each word in every utterance in the training set. In terms of human supervisory effort, FL and SL cases represent the extremes, whereas our PL method, lies somewhere in between the two. In the case of FL data, learning usually involves tuning emission distributions (the model may need to learn intra-word segmentations). On the other hand, in the SL case, in addition to the intra-word segmentation and the emission distributions, the recognizer also learns the inter-word segmentations.

In the general context of learning, training using FL data falls more into the category of supervised learning.[1] Training a speech recognizer using SL data is an instance of semi-supervised learning [3], and a more general semi-supervised learning setting involves training using both labeled and unlabeled data. In the case of speech recognition, this means we have transcripts (in most cases in the SL form) for a subset of the training data, and no annotations for other parts of the training data. One popular approach to semi-supervised learning is *self-training*. Self-training has been used in the past to train speech recognizers [4, 5, 6, 7]. In most of the above approaches, a previously trained recognizer is used to generate transcripts for unlabeled data, which are then used to re-train the recognizer after rejecting the erroneous transcripts based on some measure of recognizer confidence. The algorithms usually differ in the way the recognizer confidence is measured and the manner in which erroneous parts of the transcript are handled. For example, in the case of lightly supervised training [7], the output of the recognizer is compared against closed-captions to determine the reliable regions. In addition, a language model is also used to generate confidence values. Such approaches are particularly useful while developing recognition systems in languages for which large amounts of annotated data do not exist. The success of self-training based approaches largely depends on accurate estimation of recognizer confidence. While in tasks such as broadcast news, we can make use of closed-captions to estimate these confidence values, in the case of conversational speech, we do not have access to closed-captions and thus confidence estimation is a challenging problem.

While the techniques proposed in this paper may be extended for semi-supervised learning, the focus of this paper is to introduce a new method for annotating speech data and show how it can be used to train large vocabulary speech recognition (LVCSR) systems. If the amount of training data is fixed and finite, the FL case contains at least as much information about the hidden variables as SL data. In addition, under the above assumptions, a learner trained on FL data can potentially outperform a similar learner trained on SL data [2]. In the case of speech recognition, however, SL data is usually employed for training as obtaining FL data involves significant human effort. Further, in the case of continuous speech, accurate word segmentations are sometimes difficult to obtain as a result of co-articulation and/or word-boundary ambiguity. To illustrate this difficulty, consider the spectra shown in figure 1. In the first spectrogram, as a result of co-articulation, the boundary between the words "one" and "winter" is not clearly defined. In the second case, the boundary between all the three words is ambiguous at best. Listening to these utterances only strengthens this point [2]. In such cases, pro-

This work was supported by an ONR MURI grant, No. N000140510388.

[1] Of course, since only word and not phone segment information is known, this still would not be a fully supervised learning setting.

[2] Manual segmentation of word segmentation for the above two exam-

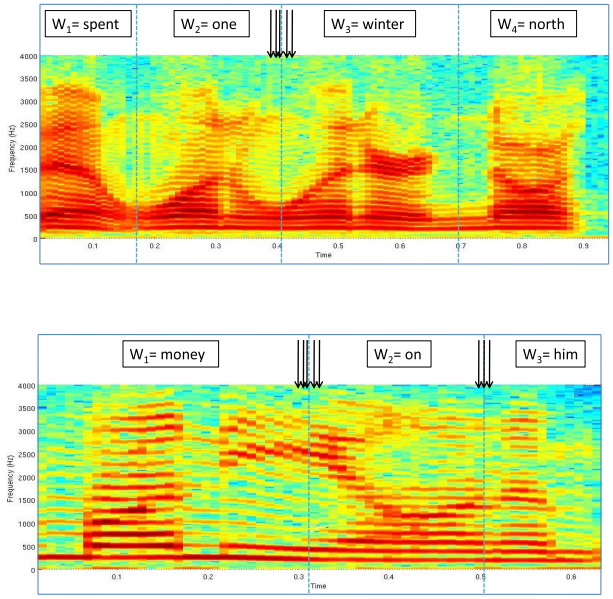

Fig. 1. Spectra of cuttings obtained from Switchboard conversation sw02423. The first figure was obtained by cutting the B-channel from 1:44:741s to 1:45:704s and the second one was obtained by cutting the A-channel from 8:05:530s to 8:06:622s. The arrows show word boundaries hypothesized by speech researchers asked to annotate these cuttings.

viding accurate word level segmentations with high inter-annotator agreement is difficult, certainly extremely costly, and might be impossible. On the other hand, it is extremely easy to label a frame that belongs somewhere in the middle of each of these words.

In this work, we propose a technique to label training data for speech recognition that does not require the annotator to provide accurate segmentation information (i.e. the FL case). The proposed scheme requires the annotator to provide only labels for at least one frame associated with each word in all the utterances. We show that annotating speech data using the proposed approach is about 3 times as fast as fully-labeling, and only about 2 times slower than sequence labeling. Thus, in comparison to providing accurate segmentation information, this method involves a significantly smaller amount of human effort, but only slightly more effort than annotating the sequences. Furthermore, it removes the possibility of labeling errors at word boundaries since the annotator need only provide labels on or near the center of the word. In the past, this annotation technique has been successfully applied to activity recognition [8] and phone recognition domains [2]. The training algorithm uses the notions of virtual evidence (VE) [9]. VE based ASR systems have been successfully used in the past to help decoding [10, 11], but to the best of our knowledge this paper presents the first system to express training uncertainty and show improvements using VE in the LVCSR domain.

It is important to highlight that, while all models in this paper are generatively trained using the expectation maximization (EM) algorithm [12], data in the proposed approach can easily be used for discriminative training. Further it is possible to train systems using a combination of data in FL, SL and PL formats. For example, if we had access to say, 2000 hours of SL data, and, 200 hours of PL data, we can use both the SL and PL data to train a recognizer (see section 7).

ples by 5 seasoned speech researchers yielded boundaries with a standard deviation of 95ms. The speech files used in figure 1 may be obtained here: http://ssli.ee.washington.edu/~asubram/annotation-examples

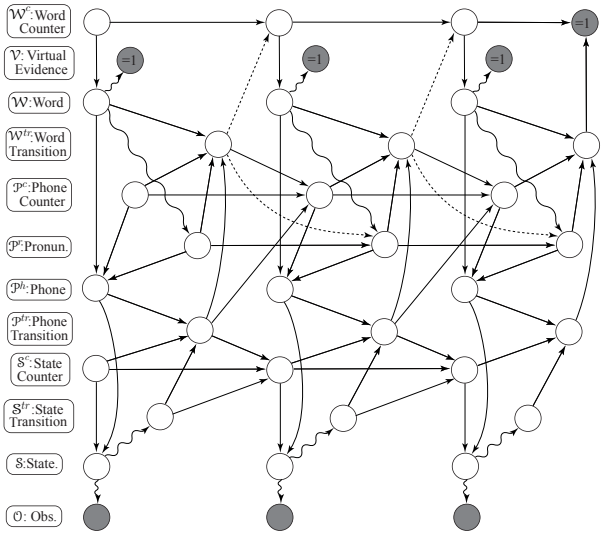

Fig. 2. Virtual Evidence Training Graph.

2. BASELINE SYSTEM

The probabilistic models used for parameter training are expressed using Dynamic Bayesian Networks (DBNs). The baseline model is equivalent to a standard speech recognition Hidden Markov Model (HMM), but expressing it as a DBN allows us to extend the baseline to include Virtual Evidence training.

The training graph is given in Figure 2. For a detailed description of how speech recognition systems can be represented using DBNs see [13], a brief introduction will be given here. The shaded circles represent observed variables and non-shaded circles represent hidden variables. Deterministic relationships are given by solid arrows, random dependencies are wavy, and value specific "switching" dependencies are dashed arrows. The Word Counter, labeled $\mathcal{W}^c$, keeps track of the position in the current word sequence. Variable $\mathcal{W}$ is a deterministic hidden variable that represents the identity of the word. The value of $\mathcal{W}$ can be uniquely determined from $\mathcal{W}^c$ since the word sequences are known. Word Transition, $\mathcal{W}^{tr}$, is a binary variable that indicates if the graph is currently on the last frame of a word. When $\mathcal{W}^{tr}$ is false, $\mathcal{W}^c$ gets its value from the $\mathcal{W}^c$ in the previous frame. When $\mathcal{W}^{tr}$ is true, $\mathcal{W}^c$ changes its value to the next word in the sequence. Pronunciation, $\mathcal{P}^r$, is a random variable that chooses what dictionary pronunciation is being used for the given word. $\mathcal{P}^c$ is the Phone Counter and it indicates the current position in the sequence of phones associated with the given word and pronunciation. The Phone variable, $\mathcal{P}^h$, gives the identity of the current phone. Phone Transition, $\mathcal{P}^{tr}$, is a binary variable that indicates if the graph is in the last frame of the current phone. Each phone model is represented by a sequence of three states, and the State Counter, $\mathcal{S}^c$, keeps track of what state the model is in. State Transition, $\mathcal{S}^{tr}$, is a binary random variable that determines if the model should stay in the same state or transition to the next. The State variable, $\mathcal{S}$, de-

termines what mixture model to use, and the Observation, $\mathcal{O}$, is the observed feature vector. The variable $\mathcal{V}$ is always observed to be 1 and is introduced into the model so that we can represent VE. It is not used in the baseline (SL) system. It is described in detail in the following section.

3. VIRTUAL EVIDENCE

In this section we introduce the notion of VE. Consider a DBN over n random variables (rv) $\{X_1, \ldots, X_n\}$. Evidence simply means that, by some external process, we have come to know the value of a set of rvs in the model. For example, if without the loss of generality (w.l.o.g) $X_1 = \bar{x}_1$ is given, the joint distribution is no longer a function of x_1 and is given by $p(\bar{x}_1, \ldots, x_n)$. Such evidence is sometimes also referred to as *specific evidence*. Specific evidence in a model can also be represented in another way by treating x_1 as hidden, but introducing a new variable $\mathcal{V}$ into the network ($\mathcal{V} \notin \{X_1, \ldots, X_n\}$). The variable $\mathcal{V}$ is made the child of x_1 (or in general the child of the sets of variables on which we have evidence) and their relationship is expressed as

$$p(\mathcal{V} = 1|x_1, \ldots, x_n) = \delta(X_1 = \bar{x}_1) \tag{1}$$

where $\delta(x, y)$ returns a 1 when x is equal to y, and 0 on all other occasions. As a result we have that,

$$\sum_{x_1} p(\mathcal{V} = 1, x_1, \ldots, x_n) \tag{2}$$

$$= \sum_{x_1} p(\mathcal{V} = 1|x_1, \ldots, x_n) p(x_1, \ldots, x_n) \tag{3}$$

$$= \sum_{x_1} p(\mathcal{V} = 1|x_1) p(x_1, \ldots, x_n) \tag{4}$$

$$= \sum_{x_1} \delta(x_1 = \bar{x}_1) p(x_1, \ldots, x_n) \tag{5}$$

$$= p(\bar{x}_1, \ldots, x_n). \tag{6}$$

Now consider setting $p(\mathcal{V} = 1|x_1) = \kappa f(x_1)$, where $f()$ is an arbitrary non-negative function and κ is a normalization factor so that $p(\mathcal{V} = 1|x_1)$ is a valid probability density function (pdf). With this, different treatment can be given to different assignments to x_1, but unlike hard evidence, we are not necessarily insisting on only one particular value. This is referred to as *virtual evidence* (VE). In practice, the value of κ does not effect the results of inference (see [14] for details). In essence, the VE framework allows us to deal with situations when we have evidence represented as a distribution over the domain of a set of rvs [3]. There is in fact a relationship between VE and priors used in Bayesian inference, but they are not exactly the same. More details about this can be obtained in [14].

4. PROPOSED ANNOTATION SCHEME

Figure 3 shows the time and frequency domain renditions of a speech segment obtained from Switchboard conversation sw40046_B. The utterance in this segment is "what was the other" [4]. If the training data included the time points t_1, t_4, t_7, t_{10}, and, t_{13}, and also that the word "what" started at time t_1 and ended at time t_4, "was" started at t_4, and so on, then this would be FL data. In other words, we have the exact start and end times of all the words in the utterance. This is depicted below the spectrogram in figure 3 where the shaded regions mark the start and end of each word. Annotating thousands of hours of data with such word segmentation information is not only time consuming, but in many circumstances may be impossible. As shown in figure 1, co-articulation effects in conversational speech lead to fuzzy word boundaries. Thus, the general training scenario in most large vocabulary speech recognition systems does not have access to these starting/ending times, and they are trained knowing only the sequence of word labels (e.g., that the word "other" follows the word "the" follows the word "was" and so on).

Consider a new transcription based on Figure 4 [5], where the annotator, for every word in the corpus, only labels a region somewhere within the start and end of the word. For example, in the case of the word "was", whose actual start and end times are t_4 and t_7 respectively, we are given that a *part* this word occurred in the region $[t_5, t_6]$, $t_4 \leq t_5 < t_6 \leq t_7$. Similarly we are given that in the region $[t_8, t_9]$, a part of the word "the" was uttered. The region $[t_6 + 1, t_8 - 1]$ is left unlabeled. Thus, in the proposed scheme, the annotator no longer labels frames in the word transition regions, but on the other hand, provides labels for the unambiguous (and therefore more reliable) parts (i.e. on or near the center of the word). This technique of annotation results in PL data. Given the annotations in figure 4, we know that $\mathcal{W}_t =$ "was", $\forall\ t_5 \leq t \leq t_6$, $\mathcal{W}_t =$ "then", $\forall\ t_8 \leq t \leq t_9$, and no other word, except for "was" or "then", was uttered in the region $[t_6 + 1, t_8 - 1]$. It is clear that the word "was" ended at some $t' \in [t_6 + 1, t_8 - 1]$, and the word "the" began at time $t' + 1$. This implies that $\mathcal{W}_t \in$ {"was","the"}, $\forall\ t_6 + 1 \leq t \leq t_8 - 1$. In other words, the value of the word variable in the unlabeled region ($[t_6 + 1, t_8 - 1]$) must be either "was" or "the". Thus in the case of PL data, in the labeled regions we know the identity of the word variable, whereas in the unlabeled regions, we know a set of possible values that the word variable could take on.

Next we address how PL data can be used within the VE framework. We first introduce an observed child $\mathcal{V}_t$ of the word variable in the training graph in figure 2. In the following we define $W_1 \triangleq$ "was", and, $W_2 \triangleq$ "the". The conditional probability table (CPT) for $\mathcal{V}_t$, $t_5 \leq t \leq t_9$ is given by

$$p(\mathcal{V}_t = 1|\mathcal{W}_t) \tag{7}$$

$$= \begin{cases} 1 & \text{if } \mathcal{W}_t = W_1 \ \&\ t_5 \leq t \leq t_6, \\ f_t(W_1) & \text{if } \mathcal{W}_t = W_1 \ \&\ t_6 + 1 \leq t \leq t_8 - 1, \\ g_t(W_2) & \text{if } \mathcal{W}_t = W_2 \ \&\ t_6 + 1 \leq t \leq t_8 - 1, \\ 1 & \text{if } \mathcal{W}_t = W_2 \ \&\ t_8 \leq t \leq t_9, \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

where $f_t(W_1)$ and $g_t(W_2)$ represent our relative beliefs in whether the value of $\mathcal{W}_t$, $t_5 \leq t \leq t_9$ is either "was" or "the". It is important to highlight that rather than the absolute values of these functions, it is their relative values that have an effect on inference [14]. Note that the above CPT can be defined for any two consecutive words in a similar manner.

There are number of different ways of choosing $f(.)$ and $g(.)$. We could set $f_t(W_1) = g_t(W_2) = \beta, \beta > 0$. This encodes our uncertainty regarding the identity of the word in unlabeled region while still forcing it to be either W_1 or W_2, and equal preference is

[3] Does not necessarily have to be a probability measure, any Lebesgue measure will suffice.

[4] Note that in figures 3, 4, the solid-blue vertical lines showing the segmentations between words are not necessarily the best segmentations. In fact, the "best" segmentation might not even exist (see Figure 1). Rather, these figures illustrate the basics of the proposed algorithm.

[5] Same utterance as shown in Figure 3.

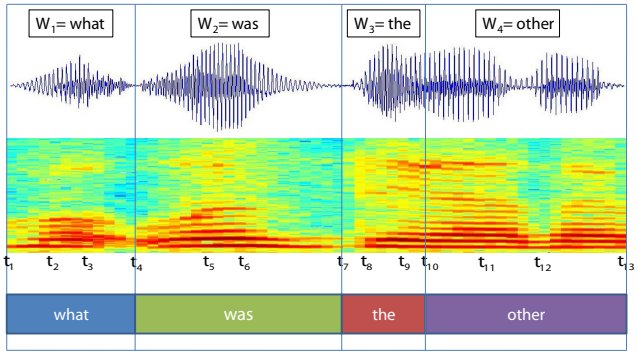

Fig. 3. Fully-Labeled (FL) case

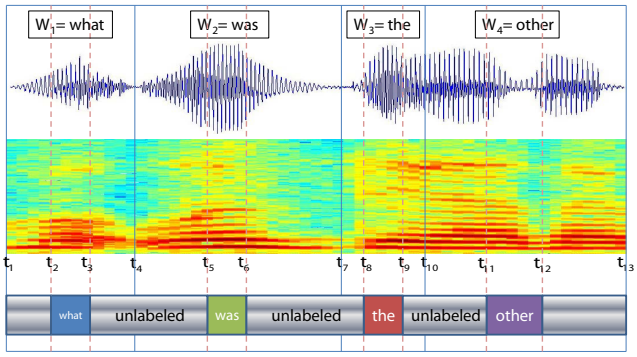

Fig. 4. Proposed Partially-Labeled (PL) case

given for both (referred to as "uniform over two assignments" in the rest of the paper). Alternatively, other functions could take into account the fact that, in the frames 'close' to t_6, $\mathcal{W}_t$ is likely to be W_1, whereas in the frames 'close' to t_8, W_2 is more likely. This can be represented by using a decreasing function of time for $f_t(W_1)$ and an increasing function of time for $g_t(W_2)$ (for example linearly increasing or decreasing with time). In the past, we have found that the "uniform over two assignments" approach performs better than the interpolation based approaches for both activity and phone recognition tasks (see [8, 2]). The success of interpolation based approaches relies on having access to (a) an estimate of the word durations, and (b) the position of the labeled frame relative to the start or end of the word (e.g. middle of the word). In the absence of the above, interpolation based approaches can lead to reduced performance as a result of over (or under)-weighting a particular assignment. Thus, in this paper we only use the "uniform over two assignments" technique in all our experiments (see section 7 for a further discussion on this topic).

In the above we suggested one way of generating the proposed PL data. PL data can also be generated by taking FL data and then dropping labels of frames around word transitions. As more labels are dropped around transitions (e.g., as $t_6 - t_5$ decreases), we use smaller amounts of labeled data. In an extreme situation, we can drop all the labels ($t_6 < t_5$) to recover the case where only sequence and not segment information is available. Alternatively, we can have $t_6 = t_5 + 1$, which means that only one frame is labeled for every word in an utterance — all other frames of a word are left un-transcribed. Once again, note that, from the perspective of a transcriber, this simulates the task of going through an utterance and identifying only one frame that belongs to each particular word without having to identify the (potentially ill-defined) word boundaries. In contrast to the task of determining the word boundaries, identifying one frame per word unit is much simpler and less prone to error [15, 16].

5. ANNOTATION TIMING EXPERIMENT

In order to compare the annotation times for SL, PL and FL formats, we asked 8 native American English speakers to annotate Switchboard utterances in the three formats. In each case, the annotators were given 9 utterances (each of length $\approx$ 15 seconds) chosen randomly from the Switchboard training set. They were instructed to annotate 3 utterances each in the SL, PL and FL formats. In the case of SL, the annotators simply listened to the speech file and gave the sequence of words, in the FL case, the annotators were asked to label the start and end times of all the words in the speech file as accurately as possible, and for the PL case, the annotators were instructed to simply mark one time point that belonged to each word. The means and standard deviations of the number of words annotated per second by the annotators for the three formats is shown in table 1. As expected annotation time in the case of SL is the smallest (i.e. most number of words per second), FL the largest and PL lies between the two. Also, given a fixed amount of time, if one can annotate X words in FL format, it is possible to annotate about $3X$ words in PL format, and about $5X$ words in SL format. Further we found that in the PL case, a large majority of annotations were close to the center of the word (even though there were no explicit instructions to do so). We also collected the following informal feedback from the participants: (a) SL case was the easiest to annotate, both in terms of time taken and the amount of effort involved, (b) most annotators felt that it was very tedious to do annotations in the FL format, and finally, (c) some annotators felt that the PL case was not much more difficult than the SL case.

Annotation Type	FL	PL	SL
Mean	0.052	0.134	0.272
Std. Dev.	0.056	0.097	0.302

Table 1. Mean and Standard Deviations of number of words annotated per sec for Switchboard. There were a total of 8 annotators in the test.

6. EXPERIMENTAL RESULTS

All systems in this paper were trained using 248 hours of Switchboard I [17] data. One issue with applying the proposed technique is the unavailability of PL Switchboard I data. In order to overcome this impediment, we used the following procedure to mimic a human annotator who labels speech data in the proposed approach: word level time annotations for Switchboard I were determined from a forced alignment using the state-of-the-art Microsoft Research large vocabulary decoder. Next, as explained in section 4, PL data was generated by dropping labels for frames around word transitions (see figure 4). For example, in order that there be n unlabeled frames in a word, we dropped labels on the first and last $n/2$ frames of that word (assuming n is even). If the total number of frames in the training set is Δ, and we drop labels on δ frames, the amount of

	FL U = 0%	PL U = 96.8%	SL U = 100%
Dev Set (dev01)	54.9	**53.3**	56.1
Eval Set (eval01)	53.1	**51.8**	53.9

Table 2. WER obtained on the 2001 Development and Evaluation sets. U represents the amount of unused labeled data.

unused data is given by $U = \frac{\delta}{\Delta} * 100$. It is important to note that PL data generated using the procedure described above differs from manually generated PL data due to two reasons, (a) there are inherent segmentation errors in the forced alignments, and, (b) the labeled frame(s) is(are) always at the center of each word. While this cannot be guaranteed in practice, we found that a large majority of the PL data in the annotation experiment had labels close to the center of the word (see section 5). In addition, our proposed PL approach is robust to word segmentation errors as only labels on or near the center of each word are necessary. To summarize, we ran forced alignment to obtain word level segmentation information. These segmentations were used as FL data. We dropped labels from the FL data to yield PL data. Forced alignment was done using transcriptions obtained from [18]. The same transcriptions were also used as SL data. Note that in the case of FL data word level segmentations were fixed during training, whereas in the case of both PL and SL systems the model had to learn the word level segmentations.

To obtain the acoustic observations, the conversations were first segmented, and then windowed using a Hamming window of size 25ms at 100Hz. We then extracted 13 PLP coefficients from these windowed features. Deltas and double deltas were appended to the above observation vector. All features were mean and variance normalized on a per-conversation side basis. The acoustics were modeled using 10,117 Gaussian mixtures, each representing a state-clustered within-word triphone [19]. All results reported in this paper were obtained using a system with 32 Gaussians per mixture. The language model was a bigram trained using approximately 22M words from Fisher and 3M words from Switchboard. The vocabulary was chosen as the 64,000 most frequent words in the training data. All training was performed using GMTK [13] and decoding was done using HTK [20]. In the case of the Switchboard corpus, using a single labeled frame for each word in every utterance corresponds to $U = 96.8\%$[6]. Note that $U = 100\%$ is the SL case, while $U = 0\%$ is the FL case. In each case the systems were trained using the EM algorithm.

The results of our experiments are shown in Table 2. All WER numbers are a result of first pass decoding (i.e. no re-scoring). In order to ensure a rapid turn-around time for our experiments, we do not use any form of adaptation (e.g. MLLR, SAT or VTLN), nor any of the standard front-end procedures (e.g. HLDA) that are common in LVCSR systems [1]. The language model (LM) scale and word insertion penalty (WIP) values were obtained by performing a grid search to optimize the performance on the development set. The results show that the system trained on PL data improves over the performance of the SL system by 2.8% on the development set and 2.1% in the case of the evaluation set. The FL system showed an improvement of about 1.2% over the SL system. This indicates that speech recognition systems can benefit from being trained on fully-labeled data. Further, it can be seen that the PL system outperforms the FL system by 1.6%. While this could be due to errors in the word segment boundaries generated using forced alignment, it is probably the case that even human transcriptions will not fix these errors since there is much inter-annotator disagreement at these word boundaries (see Section 1). These results suggest that even when one has access to automatically generated word level segmentation information (from a state-of-the-art system), it is advantageous to transform the data into PL form for system training.

[6]Average Number of Frames per word in Switchboard I is 31.78, while average number of words per "utterance" is 13.85.

	FL	PL	SL
Time Speed-Up	13.23	9.49	1
Relative Memory Usage	0.534	0.745	1

Table 3. Comparison of per-utterance inference time speed-ups and memory usage in the cases of using fully-labeled, proposed partially-labeled and sequence-labeled data for training. All entries in the table are shown relative to the corresponding SL case result. The inference times and memory usage statistics were measured using GMTK [13].

We also ran an experiment to quantitatively determine the equivalence between SL and PL data. In other words, find x and y such that, a system trained on x hours of SL data, and a system trained on y hours of PL data, yield similar performances. In the SL case, we used the system trained on all of Switchboard I data whose WER results were 56.1% and 53.9% on the dev01 and eval01 sets respectively (see Table 2). We then constructed a new training set by randomly selecting 60% of the Switchboard I corpus (approximately 148 hours of data). These utterances along with their labels in PL format were used to train a system whose WER results were 55.9% and 53.9% on the dev01 and eval01 sets respectively. Clearly, this is very similar to the performance in the case of the system trained using all of Switchboard I data in SL format. This implies that in the case of the Switchboard I corpus, for example, 60 hours of PL data is equivalent to 100 hours of SL data. Note that this is not a formal proof of SL and PL data equivalence and we plan on investigating this further in our future work.

In order to compare inference times and memory requirements to train systems using FL, the proposed PL and SL data, we randomly selected 100 utterances from the training set and ran inference on the these utterances using labels in the three formats on a 3.4MHz Intel Pentium D machine with 2GB of RAM. This was repeated 25 times, and we computed the minimum inference time over these runs in each of the three cases. We also measured the memory used in each of the cases. The results of these experiments are reported in table 3. Rather than absolute inference times and memory usage numbers, we present the performance of each system relative to the SL case. Thus in the case of time speed-up, a number larger than 1 implies that the system was faster than the SL case and in the case of memory usage, a number smaller than 1, implies it used less memory than the SL case. As expected training using FL data is the fastest and consumes the least amount of memory. It can be seen that training using PL data is about 10 times faster than SL data and requires only 74.5% of the memory used by the SL case. Also training a system using PL data is neither significantly slower, nor does it require significantly larger memory than the FL case.

7. DISCUSSION AND FUTURE WORK

We have proposed a method for labeling data used to train a LV system and shown that it can yield significant improvements over systems trained using SL data. The proposed labeling technique involves smaller amounts of human supervisory effort in comparison to labeling all word level segmentations. In addition, it also

overcomes some of the problems associated with annotating continuous speech at word boundaries. While it is the case that sequence-labeling speech data is twice as fast as the proposed approach, we have shown (see section 6) that we need about 1.67 (= 100 hours/60 hours) times as much SL data as PL data to obtain similar performance.

In the future, we plan on investigating other methods to generate the VE weights (i.e. $f(.)$ and $g(.)$). Another avenue for future work is to look at using a combination of data in different formats to train a recognizer. While the structure of the DBN used in the three cases has some differences, the decoding-time distributions learned during the training process are exactly the same and thus it is possible to share accumulators. This is particularly useful as there already exists large amounts of SL speech data (e.g. the Fisher corpus [21]). We would like to show that using small amounts of PL data in addition to large amounts of SL data can lead to improved performance. We can also get massive amounts of speech data annotated using the proposed scheme by designing an ESP-like game [22], wherein the players are instructed to label the center of the word and are rewarded for producing labels on frames that are close to each other. Also, in general, it is the case that recognizers use exponentially more SL data for linear relative gains. By making use of PL data, if a linear increase in the amount of data yields linear relative gains, this can be a huge win for training systems using this PL data. We hope to investigate the above in our future work.

8. REFERENCES

[1] G. Evermann, H. Y. Chan, and M.J.F. Gales, "Training LVCSR systems on thousands of hours of data," in *Proc. of ICASSP*, 2000.

[2] A. Subramanya and J. Bilmes, "Virtual evidence for training speech recognizers using partially labeled data," in *Proc. of the Human Language Technologies Conference (HLT-NAACL)*, 2007.

[3] O. Chapelle, A. Zien, and B. Schölkopf, Eds., *Semi-supervised learning*, MIT Press, 2006.

[4] G. Zavaliagkos and T. Colthurst, "Utilizing untranscribed training data to improve performance," in *Proc. of the broadcast news transcription and understanding workshop, Landsdowne Conference Resort, VA*, 1998.

[5] T. Kemp and A. Waibel, "Unsupervised training of a speech recognizer: Recent experiments," in *Proceedings of Eurospeech*, 1999.

[6] F. Wessel and H. Ney, "Unsupervised training of acoustic models for large vocabulary continuous speech recognition," in *Automatic Speech Recognition and Understanding Workshop, Trento, Italy*, 2001.

[7] L. Lamel, J. Gauvian, and G. Adda, "Lightly supervised and unsupervised acoustic model training," *Computer Speech and Language*, 2002.

[8] A. Subramanya, A. Raj, J. Bilmes, and D. Fox, "Recognizing activities and spatial context using wearable sensors," in *Proc. of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2006.

[9] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers, Inc., 1988.

[10] A. Subramanya, J. Bilmes, and C. P. Chen, "Focused word segmentation for ASR," in *Proc. of the Interspeech*, 2005.

[11] C. Bartels and J. Bilmes, "Focused state transition information in ASR," in *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, San Juan, Puerto Rico, November/December 2005.

[12] Dempster, Laird, and Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[13] J. Bilmes and C. Bartels, "Graphical model architectures for speech recognition," *IEEE Signal Processing Magazine*, vol. 22, no. 5, pp. 89–100, September 2005.

[14] J. Bilmes, "On soft evidence in Bayesian networks," Tech. Rep. UWEETR-2004-0016, University of Washington, Dept. of EE, 2004.

[15] S Greenberg, "The Switchboard transcription project," Tech. Rep., The Johns Hopkins University (CLSP) Summer Research Workshop, 1995.

[16] M. Lee, J. van Santen, B.Mobius, and J. Olive, "Formant tracking using segmental phonemic information," in *Proc. Eur. Conf. Speech Communication and Technology (Eurospeech)*, 1999.

[17] J. Godfrey, E. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, San Francisco, California, March 1992, vol. 1, pp. 517–520.

[18] N. Deshmukh, A. Ganapathiraju, A. Gleeson, J. Hamaker, and J. Picone, "Resegmentation of switchboard," in *Proceedings of the International Conference on Spoken Language Processing*, Sydney, Australia, November 1998, pp. 1543–1546.

[19] S. J. Young, J. Odell, and P. C. Woodland, "Tree-based state tying for high accuracy acoustic modeling," in *Proc. ARPA Human Lang. Technol. Workshop*, 1994, pp. 307–312.

[20] S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland, *The HTK Book, version 3.3*, Cambridge, U.K.: Cambridge Univ. Eng. Dept., 2005.

[21] C. Cieri, D. Miller, and K. Walker, "From Switchboard to Fisher: Telephone collection protocols, their uses and yields," in *Proc. of the Interspeech*, 2003.

[22] L. V. Ahn, "Games with a purpose," *IEEE Computer Magazine*, 2006.