LARGE SCALE DEEP NEURAL NETWORK ACOUSTIC MODELING WITH SEMI-SUPERVISED TRAINING DATA FOR YOUTUBE VIDEO TRANSCRIPTION

Hank Liao, Erik McDermott, and Andrew Senior

Google Inc.

{hankliao, erikmcd, andrewsenior}@google.com

ABSTRACT

YouTube is a highly visited video sharing website where over one billion people watch six billion hours of video every month. Improving accessibility to these videos for the hearing impaired and for search and indexing purposes is an excellent application of automatic speech recognition. However, YouTube videos are extremely challenging for automatic speech recognition systems. Standard adapted Gaussian Mixture Model (GMM) based acoustic models can have word error rates above 50%, making this one of the most difficult reported tasks. Since 2009, YouTube has provided automatic generation of closed captions for videos detected to have English speech; the service now supports ten different languages. This paper describes recent improvements to the original system, in particular the use of owner-uploaded video transcripts to generate additional semi-supervised training data and deep neural networks acoustic models with large state inventories. Applying an "island of confidence" filtering heuristic to select useful training segments, and increasing the model size by using 44,526 context dependent states with a low-rank final layer weight matrix approximation, improved performance by about 13% relative compared to previously reported sequence trained DNN results for this task.

Index Terms— Large vocabulary speech recognition, deep neural networks, deep learning, audio indexing.

1. INTRODUCTION

More than one billion people come to YouTube every month to access news, information, and entertainment. In doing so, they watch six billion hours of video every month. While much of the content may be lovable cats riding on Roombas, or the latest Psy video raking in the views, there is still a significant amount of important spoken content on YouTube from informal video blogs to high quality broadcast news. Hillary Clinton has held a global town hall live on YouTube. Khan Academy, an educational organization, provides over 4000 videos on a variety of subjects and has a million subscribers on YouTube. Some producers pay for hand-transcribed closed captions, however this can be expensive, time-consuming, and does not scale to the more than 20 hours of content being uploaded every minute to YouTube [1]. In addition to improving accessibility for the hearing impaired and non-native speakers, closed captions can be used to improve search for videos and search within videos. Thus, providing automatic closed captions using speech recognition technology can be an attractive and useful service. Figure 1 shows an example of a YouTube video with automatically generated captions. The captions are superimposed on the video content by clicking the CC button in the bottom right of the video window. Captions also allow the video to be browsed by clicking on a given segment in the scrollable part of the entire caption track, as shown in the bottom part of the figure. There are over 300 million videos with captions on YouTube, the vast majority produced using automatic speech recognition (ASR) or alignment. Due to the heterogeneity of videos found on the web [2], automatic transcription of YouTube videos is challenging.

Recently the application of multilayer perceptrons, or artificial neural networks, has become popular for acoustic modeling in ASR [3]. Still trained using standard back-propagation [4], lately the large improvements have been due to the use of many hidden layers and nodes, a large number of context dependent hidden Markov model (HMM) states, and training of these "deep" neural networks (DNNs) on fast graphical processing units [5]. Impressive gains of more than 15% relative over GMM-based acoustic models have been shown on a variety of tasks by many research groups, first on the small vocabulary TIMIT task [6], and then on larger Broadcast News [7], Switchboard/Fisher [8], and Voice Search [9] tasks. In [9], while the improvements on VoiceSearch by using a DNN acoustic model were more than 20% relative compared to a highly tuned GMM- acoustic model, the YouTube improvements were modest at less than 6%. Although it may be argued that YouTube is a more difficult task, and the comparison GMM system baseline applied CMLLR and MLLR adaptation techniques, in this work we show that significant further improvements can be made using DNN modeling

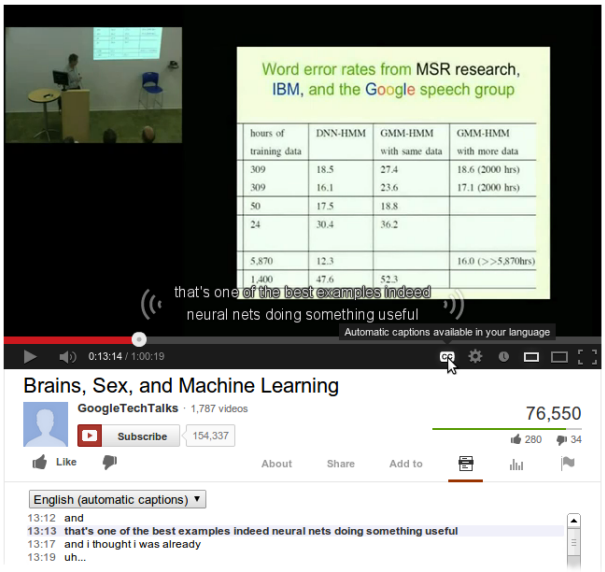

Fig. 1. The automatic captions interface on YouTube.

for speech acoustics.

Since 2006 it has been possible for YouTube video owners to upload their own closed caption files as subtitles to YouTube videos. A number of standard formats, such as SubRip (.srt) and SubViewer (.sub) exist as part of off-the-shelf video editing software. These formats specify start and end times for each caption to be displayed over the video in Closed Caption mode. Furthermore, in 2009, YouTube launched a new feature, dubbed "Autosync", which allows video owners to upload a simple text transcript of the spoken content of the video as an alternative to automatically-created closed captions using speech recognition [1]. The transcript, containing no timing information, is force-aligned ("auto-sync'ed") on servers using standard ASR algorithms to generate start and end times for reasonably spaced chunks of audio, over the course of the video [10]. This feature has proved extremely popular, as it saves the video owner the trouble of transcribing the spoken video content segment-by-segment and marking the start/end times explicitly. These captions could be a useful source of transcripts for training [11, 12].

This paper describes the YouTube speech recognition system used for generating automatic captions and reports improved results on the same YouTube test set used in [9]. The main contributions of this paper are the description of a large-scale application of ASR for transcribing YouTube videos, the use of semi-supervised data in training models, and experiments conducted on large data sets with the use of a low-rank approximation of the final weight matrix with the largest number of context dependent state targets that we are aware of being reported. It is organized as follows. First, details on how user-supplied captions can be used for training are discussed in section 2. Second, some techniques that are applied in this paper to improve recognition are reviewed in section 3. Experiments are then presented followed by conclusions drawn from this work.

2. LEVERAGING AUTOSYNC DATA UPLOADED BY YOUTUBE VIDEO OWNERS AS ADDITIONAL SOURCES OF AM TRAINING DATA

Both of these sources of data, Captions and Autosync, potentially offer a rich source of additional training for improved acoustic modeling in the context of automatic caption generation for YouTube videos. This study focuses on the use of Autosync only. For purposes of ASR training, the critical question is how reliably the owner-uploaded transcript data actually matches the underlying acoustics of the videos in question.

2.1. Non-acoustic pre-filtering of owner-uploaded Autosync data

Much of the data uploaded by video owners as Captions/Autosync data bears no useful resemblance to the true transcript. It is desirable to filter out as much junk as possible early, before turning to an acoustic confidence measure. In this study, the following simple filters were applied:

1. Language mismatch. Using a text-based language detector, captions that are ostensibly in the target language but actually in another language are removed from consideration. This occurs in the context of an overall pipeline that attempts to generate pronunciations for the target language (e.g. American English) for any given word, whether or not the uploaded transcript word exists in the target language.
2. Likely text mismatch. A large number of owner-uploaded captions are product ads, with no relation to the underlying audio. A useful heuristic was to remove all captions containing URLs. Additional filters were used, e.g. detecting non-ASCII characters in the captions data.

Of a total of 26,000 hours of videos with auto-aligned captions, 14,000 hours were accepted by these filters.

2.2. Confidence filtering Autosync data using an existing acoustic model and the "islands of confidence" filter

Having applied the non-acoustic filtering described above, the owner-uploaded Autosync data was passed through an additional filter using a given acoustic model. The procedure is extremely simple. The filtered owner-uploaded transcript is force-aligned to the audio. A simple confidence filter is then carried out as follows. In addition to the forced-alignment of the given text to the audio, a miniature trigram language model is generated using only the Autosync transcript. The audio portion is recognized using a decoder graph derived from the mini-LM, and the decoding result is then Edit-Distance-aligned to the owner-provided transcript. All matching words (with Edit-Distance of zero) are given a "confidence" of one; non-matching words are given a confidence of zero. The overall transcript filter is then implemented in terms of this binary confidence measure: If at least N consecutive words in the transcript have confidence of one, an "island of confidence" is deemed to have been detected. A specific example is as follows.

1. Video owner uploads word sequence "A B C D E F".
2. Autosync server decodes audio with mini-LM, which produces "A B F D E F"
3. A, B, D, E and F from owner matches the mini-LM decoded output, and so get confidence 1; C gets confidence 0.
4. "Islands of confidence": consecutive words with confidence 1
   (a) "A B" $\rightarrow$ Island of size 2
   (b) "D E F" $\rightarrow$ Island of size 3

More sophisticated filters using, e.g., lattice-based word posteriors [13] could certainly be used instead of the simple binary match/no-match word-level measure used here. Nonetheless, use of an aggressive choice of minimum island length was found to be effective at filtering out non-matched transcripts. The cost of this aggressive filtering is a high false rejection rate, but given the large amount of data available in the overall set of owner-uploaded videos, the approach is practical. Applying a minimum island length of $N = 50$ to a total set of 14,000 hours of Autosync video segments passing the non-acoustic filtering described previously resulted in "high confidence" transcripts for 1,450 hours of video segments. That dataset is referred to as "Autosync13" in the following. Preliminary analysis of the ROC curve corresponding to this filter using artificially corrupted known transcripts suggests that at $N = 50$, the false acceptance rate is below 10%. When this auto-sync procedure is evaluated on the `YtiDev11` test set of YouTube videos described below, the WER is 38.7%; 76% of these errors are deletions, which can be viewed as false rejects of words that should appear. This is a result of the high minimum island length. The work in [12] can transcribe more data by combining available "imperfect transcripts" with a traditional well-trained speech recognizer while decoding found audio; however, this "direct decoding algorithm" requires a good base recognizer to produce transcripts where none exist. Because of high initial YouTube automatic transcription error rates we chose to focus on aligning existing transcripts only.

3. TRAINING AND MODELING TECHNIQUES

In this work we use conventional fully-connected, feed-forward neural networks with sigmoid non-linearities and a softmax output layer [9]. The networks are trained with minibatch stochastic gradient descent and back-propagation. We experimented with networks with different numbers and sizes of hidden layers, as well as varying the size of the output layer. Throughout we use a fixed input window of 21 stacked frames of 40-dimensional mel spectrum log filterbank energies, 10 to the left and 10 to the right of the center frame to yield an 840 dimensional input vector, computed on 25ms windows with a 10ms step. On the basis of previous experimentation [14] we use an exponentially decaying learning rate schedule whereby the initial learning rate of 0.1 decays by a factor of 10 every 1.5 billion frames. The minibatch size is fixed at 200 and a constant momentum of 0.9 is used.

Our previous work on YouTube transcription showed little improvement, that is less than a 10% relative improvement, from using frame or sequence-based trained DNNs over GMM acoustic models [9]. While adaptation has been shown to be effective for GMM acoustic models, the improvements are small, e.g. less than 5% relative, for speaker adaptation of larger DNN networks in the tens of millions of parameters [15, 16]. In previous experimentation with VoiceSearch tasks we obtained reductions in word error rate by increasing the context dependent state inventory beyond the number of states chosen for a baseline GMM system. Consequently we were interested in increasing the number of states for improving YouTube transcription accuracy. Unfortunately, in the final layer the number of parameters increases in proportion to the number of output states, and since the number of states is large compared to the sizes of the hidden layers, the final layer contains most of the parameters of the network. Increasing the state inventory increases the number of parameters and decreases the speed of training and decoding of the network.

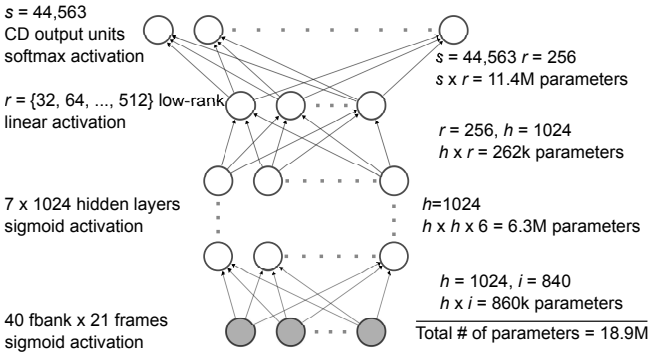

Fig. 2. DNN with low-rank approximation to final weight matrix.

Following Sainath *et al.* [17] we investigate a low-rank approximation to the final layer of the network, effectively replacing it by a linear layer with a small number of hidden units followed by a softmax layer. For a hidden layer of size h and a softmax layer with s state posterior outputs, using a rank r approximation (inserting a new linear layer with r linear hidden units) changes the number of parameters from $h \times s$ to $r \times (h+s)$. An example of this is shown in figure 2 where a low-rank approximation of the final weight matrix reduces the number of parameters from 53M to 18.9M with low-rank size of 256 nodes for a CD state inventory of 44,563 nodes (45k). With this approximation we are able to investigate much larger state inventories than the 2000, 6000 and 9000 units in [17]. The largest previous deep network state inventory known to us is 32,000 [18]. A hierarchical softmax [19] would also enable large state inventories with reduced numbers of parameters but complicates training and decoding. In this paper, we examine the use of a low-rank approximation for state inventories larger than what has been previously explored. When using a low-rank approximation of the final weight matrix, the linear low-rank layer is randomly initialized with a smaller variance of 0.5 and lower starting learning rate of 0.005.

4. EXPERIMENTS

Systems are trained on several data sets shown in Table 1. The `BnTrain97` is a publicly available training set. The set `BnGvYtn08` adds Google Video (now superseded and replaced by YouTube) and YouTube News data. These two sets are hand transcribed to provide high quality reference transcripts with manually obtained utterance boundaries. Both the Google Video and YouTube News videos were chosen from news channels, using news sources as a proxy for higher quality content. The `Autosync13` data is collected as described in section 2; in contrast to the previous data sets, this set is expected to be broader. The final training set, `AsBnGvYtn13`, is the union of these supervised and semi-supervised training sets. The neural networks are all trained on a single Nvidia GPU board, e.g. Tesla K20m or M2090, using the CUDAMat library [20].

The training sets are created by aligning the training data with a Broadcast News acoustic model with a context dependent state inventory produced using decision tree state tying [21] estimated on `BnTrain97` with a minimum number of observed frames per leaf of 3000 and contexts of 3. Standard phonetic questions are asked, yielding 6917 context dependent states states — this is the 7k inventory set. To create larger CD state inventories of 21k and 45k states, decision trees are estimated on the same data; for 21k states, the minimum observed frame count is reduced to 100, and for 45k states the minimum context is also reduced to 1. The data however is not realigned; the CD state targets are simply remapped from the 7k to 21k or 45k state inventories. The resulting increase in the number of CD states increases the overall size of the static decoding graph by less than 0.5%.

Testing is conducted on several data sets shown in Table 1. The test set `BnE97` is a standard, publicly available test set. The `Ytn08` test set is a collection of YouTube videos from news channels. The last test set `YtiDev11` is a sample of YouTube videos chosen based on those which have higher view counts; these videos tend to be more unpredictable with more variation due to spontaneous speech, noise, and music compared to the `Ytn08` test set. Unless noted, experiments are all conducted using a large search beam and number of active search paths to minimize search errors. The base language model is a Kneser-Ney smoothed trigram model with 15M n-grams and a vocabulary of 127122 words.

4.1. Autosync trained DNN

First, experiments were conducted to determine the quality and usefulness of semi-supervised training data, e.g. Autosync data, for acoustic model training with DNNs. Results are presented in Figure 3 for 7k CD state output targets only, but two different model topologies: 7x1024 hidden states, with 14M parameters, and 6x2048 hidden states with 37M parameters. Having more parameters shows

Train Set	Description	Shows	Hours	Frames	Words
BnTrain97	'96-97 Broadcast News (Hub4)	288	144	52M	1.7M
BnGvYtn08	BnTrain97 + '08 Google Video, '08 YouTube News	9.3k	764	275M	8.7M
Autosync13	Filtered YouTube Autosync user-captioned videos	79.3k	1016	366M	13.2M
AsBnGvYtn13	BnGvYtNews08 + Autosync13	88.6k	1781	641M	21.9M

Test Set	Description	Shows	Hours	Frames	Words
BnE97	1997 Broadcast News Eval (Hub4)	72	2.9	1.0M	33k
Ytn08	2008 YouTube News videos	240	11.1	4.0M	114k
YtiDev11	2011 YouTube view-count weighted videos	125	6.6	2.4M	68k

Table 1. Training and test sets.

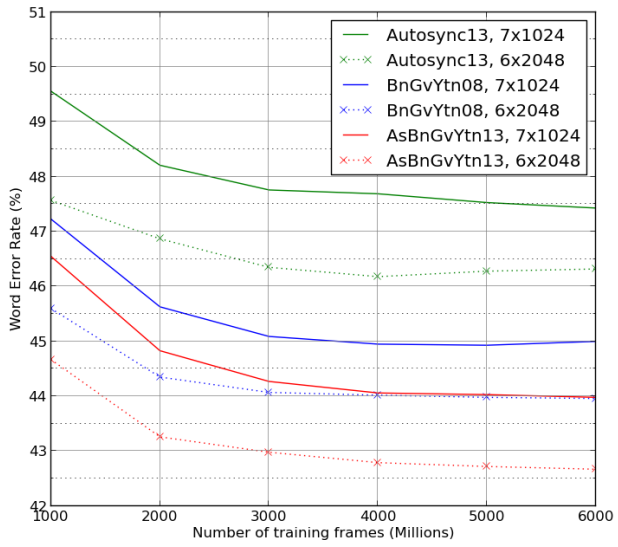

Fig. 3. Training on Autosync13 (366M frames semi-supervised), BnGvYtn08 (275M frames supervised), or AsBnGvYtn13 (641M frames mixed supervision) with 7k CD states. Test on YtiDev11.

a consistent gain of just over 1% absolute across the various training sets. Some evidence of over-training appears to occur after about 4-5 billion frames of training data are observed; this is over 14 epochs for the smaller BnGvYtn08 training set and 6-8 epochs for the larger, combined AsBnGvYtn13 training set. When only using the Autosync data itself, a relatively decent YouTube transcription system can be obtained converging to about a 47.5% error rate for the 7x1024 system and a 46.25% error rate for the 6x2048. This is still worse than the systems trained solely on supervised data, but by less than 5% relative. For both DNN sizes, adding the semi-supervised Autosync data also clearly helps by about 1% absolute for the 7x1024 DNN, and 1.25% for the 6x2048 DNN; however, this is less than a 3% relative improvement over not using the Autosync data.

The use of semi-supervised Autosync data for training showed small improvements on YtiDev11, however it is of interest whether the data improves performance for all data sets, since YouTube is quite diverse. In Table 2, some results on other test sets are shown. In comparison to the general YtiDev11 test set, BnE97 and Ytn08 are primarily news content and better matched to the supervised training set BnGvYtn08. Thus, it is not entirely unexpected to see that adding the Autosync13 data hurts performance

Size	Training set	BnE97	Ytn08	YtiDev11
7x1024	Autosync13	22.6	31.4	47.5
	BnGvYtn08	12.5	23.1	44.9
	AsBnGvYtn13	12.9	24.4	44.0
6x2048	Autosync13	24.3	30.6	46.3
	BnGvYtn08	12.1	21.4	44.0
	AsBnGvYtn13	12.3	23.0	42.7

Table 2. Comparison of DNNs trained on different training sets on various test sets. Upper results are 7x1024 hidden layer DNNs, the lower 6x2048, both 7k CD states and after 5B frames of training.

on these test sets compared to just using the smaller supervised training set. While on the YtiDev11 test set, the solely semi-supervised trained Autosync13 model is less than 10% relative worse, on the more mismatched news sets, the error rate can be double the error rates of supervised-trained models. These results show some domain specific effects of the data and perhaps lack of generalization and over-fitting to data.

4.2. Large CD state inventory

As discussed in Section 3, we explore using much larger numbers of context states than typically used and investigate approximating the resulting, large final weight matrix as the product of two low-rank matrices. Results comparing these approaches for 7k, 21k and 45k systems are shown in Figure 4. The results clearly show that using an increased number of output targets improves accuracy by about 0.5%-1% absolute; however this comes at a large computation cost for training and test. For example with a 7k state inventory, training occurs at 9.7kframes/sec on a K20m, but slows to 2.5kframes/sec with 45k states. Using the low-rank approximation with 512 nodes, the speed is improved by about 50% to 3.7kframes/sec. Moreover, the reduced number of parameters acts as a regularization that performs better at less than a 42.5% word error rate. As expected, for the 45k state inventory systems, using a low-rank approximation appears to help with speed and accuracy improvement of just under 1.0% absolute for 256 and 512 nodes. (Unfortunately, some of the largest systems have not trained to convergence after over 5B frames of training.) On the smaller 21k sized systems, the low-rank approximation is not as effective at improving over the 7k baseline systems by less than 0.5% absolute. When comparing the number of parameters, a plain 45k state system with 7x1024 hidden layers has about 52.8M parameters where 45.6M lie in the last weight matrix alone; this is reduced to 18.9M overall and 11.4M in the last layer by hav-

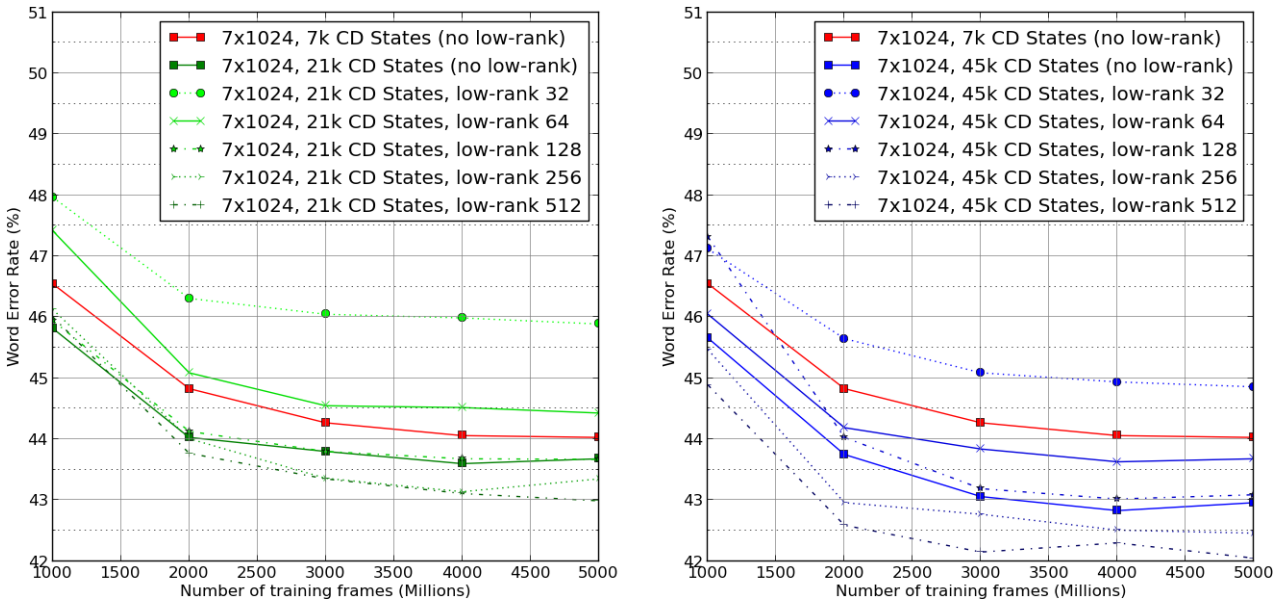

Fig. 4. Training on `AsBnGvYtn13` (641M frames). Test on `YtiDev11`.

ing a low-rank approximation using 256 nodes. This is less than half the number of parameters in the 6x2048/7k system (37M), yet gives an word error rate that is approximately the same at 42.5% versus 42.7%.

4.3. Comparison to other approaches

Some results on YouTube transcription have been reported by our group in the past [9, 22]. The different results are summarized in Table 3. It is useful to compare these previous experiments with the ones in this paper. Although the training set used is not exactly the same, the amount of data is similar with a comparable amount of training data and the same set of supervised training data used; only the Autosync data has changed. Results are reported on the same test set. The previous DNN training results [9] used SAT-CMLLR-MFCC features, a larger mini-batch size of 500 frames, and pre-trained each layer for 1 epoch as an RBM with a further 5 epochs of fine tuning of the entire network (fine-tuning over 2.7B frames). The first layer contained 2000 hidden, followed by three hidden layers of 1000 nodes each, and finally 18k CD state output nodes; this adds up to about 23M parameters. Adding sequence training, as implemented in [9], only gave a small improvement of 0.5% absolute which may be due to a missing heuristic [23] or limited gains from sequence training on a larger, semi-supervised training set. These numbers are slightly better than earlier reported figures of 49.4% for cross-entropy trained system and 48.8% for MMI sequence trained [22] due to further training a reduced learning rate plus sparsity.

System	%WER
MFCC GMM, 18k state, 450kcomps	52.3 [9]
MFCC DNN, pre-trained	47.6 [9]
+ MMI batch sequence training	47.1 [9]
Fbank DNN 7x1024, 7k state	44.0
Fbank DNN 6x2048, 7k state	42.7
Fbank DNN 7x1024, low-rank 256, 45k state	42.5
Fbank DNN 7x2048, low-rank 256, 45k state	40.9

Table 3. Comparison of previously reported results to the best results in this paper on `YtiDev11`.

In contrast, our baseline 7x1024 system with 7k output states is better than any of these previously reported results, albeit for a slightly different and larger training set size of 641M frames vs 520M frames. We suspect the improvements are due not to the different training set, but to the smaller batch size (200) providing more frequent updates for a given number of training epochs and a switch to a larger window of log filterbank features rather than LDA projected MFCCs that have been CMLLR adapted (9 frame window by 13 static MFCC coefficients, projected down to 39 dimensions via LDA and then modified by a speaker-specific affine transform). The increased training set size is due to more Autosync data and that was found to provide at most a 1% improvement compared to not using it. The pre-training and then fine-tuning for 4 epochs is equivalent to about 2B frames of training; as shown in Figure 3 a couple of the systems after 2B frames of training are below 45% WER on `YtiDev11`—even a system without the Autosync data. It appears that on datasets of this size, pre-training is unnecessary and having a wider and deeper network is more helpful. On this task, very simple DNN systems outperform a much more complex, previously reported, CMLLR-adapted MFCC, sequence trained, large language model DNN system. Combining the wide hidden layer topology of 2048 nodes with a low-rank approximation with high number of CD states in the final layer yielded the best result of 40.9% WER. Compared to the previous best acoustic modeling result of 47.1%, there is a 13% relative improvement and a 22% improvement over the GMM system with the language model held fixed. Using a much larger distributed language model for re-scoring [24] yielded a big improvement of 3.6% [22] absolute. It is expected that a larger language model and even larger CD state inventories trained on significantly more data would combine to yield even better results: this is the aim

of future work.

5. CONCLUSIONS

In this paper we presented significant improvements to speech recognition applied to YouTube videos using semi-supervised Autosync data along with larger neural network acoustic models. Many thousands of hours of Autosync data are available, however an important issue is removing low-quality captions. We do so using an "Island of Confidence" heuristic that enables us to produce about a thousand hours of high quality semi-supervised training data. While this gives about a 1% absolute improvement on our general YouTube test set, on a slightly different news domain it degrades accuracy slightly. Given the same language model, we show that by increasing the model size through increasing the width of hidden layers to 2048 nodes, deepening the network to 7 hidden layers, and using a low-rank approximation to the final weight matrix with 45k context dependent triphone states yields a relative gain of about 13% over a previous MMI sequence-trained, CMLLR-adapted MFCC DNN system, and more than 20% relative gain over an adapted MFCC GMM acoustic model baseline. It is expected by combining all these aspects—more semi-supervised data, larger language models and bigger neural network acoustic models—further significant gains can be achieved. Additional research will investigate this and examine how these systems can be trained effectively and efficiently.

ACKNOWLEDGMENTS

The authors would like to thank Chris Alberti for help with the Autosync servers and Georg Heigold for his help with the CD state mapping.

6. REFERENCES

[1] C. Alberti and M. Bacchiani, "Automatic captioning in YouTube," Dec. 2009, http://googleresearch.blogspot.com/2009/12/automatic-captioning-in-youtube.html.

[2] N. Ryant, M. Liberman, and J. Yuan, "Speech activity detection on YouTube using deep neural networks," in *Proc. Interspeech*, 2013.

[3] M. Cohen, H. Franco, N. Morgan, D. Rumelhart, and V. Abrash, "Hybrid neural network/hidden Markov model continuous-speech recognition," in *Proc. Eurospeech*, 1992.

[4] D. Rumelhart, G. Hinton, and R. Williams, "Learning representation by back-propagating errors," *Nature*, vol. 323, Oct. 1986.

[5] D. Yu, L. Deng, and G. Dahl, "Roles of pretraining and fine-tuning in context-dependent DNN-HMMs for real-world speech recognition," in *Proc. NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, Dec. 2010.

[6] A. Mohamed, G.E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. on Acoustics, Speech, and Language Processing*, 2012.

[7] T. Sainath, B. Kingsbury, B. Ramabhadran, P. Fousek, P. Novak, and A. Mohamed, "Making deep belief networks effective for large vocabulary continuous speech recognition," in *Proc. ASRU*, 2011.

[8] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Proc. Interspeech*, 2011.

[9] N. Jaitly, P. Nguyen, A.W. Senior, and V. Vanhoucke, "Application of pretrained deep neural networks to large vocabulary speech recognition," in *Proc. Interspeech*, 2012.

[10] P. J. Moreno and C. Alberti, "A factor automaton approach for the forced alignment of long speech recordings," in *Proc. ICASSP*, 2009.

[11] L. Lamel, J.-L. Gauvain, and G. Adda, "Lightly supervised and unsupervised acoustic model training," *Computer Speech and Language*, Jan. 2002.

[12] B. Lecouteux, G. Linarès, and S. Oger, "Integrating imperfect transcripts into speech recognition systems for building high-quality corpora," *Computer Speech and Language*, vol. 26, no. 2, Apr. 2012.

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

[14] A. Senior, G. Heigold, M. Ranzato, and K. Yang, "An empirical study of learning rates in deep neural networks for speech recognition," in *Proc. ICASSP*, 2013.

[15] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context dependent deep neural networks for conversational speech transcription," in *Proc. ASRU*, 2011.

[16] H. Liao, "Speaker adaptation of context dependent deep neural networks," in *Proc. ICASSP*, 2013.

[17] T. N. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, and B. Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets," in *Proc. ICASSP*, 2013.

[18] X. Chen, A. Eversole, G. Li, D. Yu, and F. Seide, "Pipelined back-propagation for context-dependent deep neural networks," in *Proc. Interspeech*, 2012.

[19] T. Mikolov, S. Kombrink, L. Burget, J. Černocký, and S. Khudanpur, "Extensions of recurrent neural network language model," in *Proc. ICASSP*, 2011.

[20] V. Mnih, "CUDAMat: a CUDA-based matrix class for python," Tech. Rep. TR 2009-004, Department of Computer Science, University of Toronto, 2009.

[21] S.J. Young, J.J. Odell, and P.C. Woodland, "Tree-based state tying for high accuracy acoustic modelling," in *Proc. ARPA Workshop on Human Language Technology*, 1994, pp. 307–312.

[22] C. Chelba, D. Bikel, M. Shugrina, P. Nguyen, and S. Kumar, "Large scale language modeling in automatic speech recognition," Tech. Rep., Google, 2012.

[23] H. Su, G. Li, D. Yu, and F. Seide, "Error back propagation for sequence training of context-dependent deep networks for conversational speech transcription," in *Proc. ICASSP*, 2013.

[24] T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean, "Large language models in machine translation," in *Proc. of Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2007.