Leveraging Large Amounts of Loosely Transcribed Corporate Videos for Acoustic Model Training

Matthias Paulik and Panchi Panchapagesan

Cisco Speech and Language Technology (C-SALT), Cisco Systems, Inc.
170 W Tasman Drive, San Jose, CA 95134, USA
mapaulik@cisco.com

***Abstract*—Lightly supervised acoustic model (AM) training has seen a tremendous amount of interest over the past decade. It promises significant cost-savings by relying on only small amounts of accurately transcribed speech and large amounts of imperfectly (loosely) transcribed speech. The latter can often times be acquired from existing sources, without additional cost. We identify corporate videos as one such source. After reviewing the state of the art in lightly supervised AM training, we describe our efforts on exploiting 977 hours of loosely transcribed corporate videos for AM training. We report strong reductions in word error rate of up to 19.4% over our baseline. We also report initial results for a simple, yet effective scheme to identify a subset of lightly supervised training labels that are more important to the training process.**

***Index Terms*—lightly supervised acoustic model training, automatic speech recognition, LVCSR**

I. Introduction

Video is expected to account for 90 percent of all internet traffic by 2013. With ever growing archives of video data, the need for solutions that make videos searchable and browsable becomes more and more urgent. This is especially true within corporate settings. The lack of high-quality automatic solutions for this task often times prompts a manual labeling of the most important corporate videos, despite the attached high costs. Over time, considerable amounts of labeled videos can accumulate, forming an ideal resource for lightly supervised acoustic model (AM) training. This paper gives an overview on the state-of-the-art in lightly supervised AM training and describes our efforts on AM training with 977 hours of loosely transcribed Cisco corporate videos. The main challenges faced include highly disfluent speech in the loosely transcribed corpus and a strong data mismatch to our baseline supervised training corpus.

The reminder of this paper is organized as follows. In section II we explain and review techniques used for lightly supervised AM training. Section III present our experimental setup. A detailed description of our experiments, along with their results, is given in section IV. We first validate and tune the used lightly supervised training techniques on our development set and then apply the techniques on the full training corpus of loosely transcribed videos. Finally, in section V, we conclude with a short summary and discussion of our results.

II. Lightly Supervised AM Training

Acoustic model training requires large amounts of highly accurate transcriptions that include speech disfluencies (e.g. fillers, word fragments), markers for non-speech events (e.g. noise, breath), and segment level timing information. Creating highly accurate transcriptions is costly and time-consuming. For this reason, lightly supervised acoustic model training has received a considerable amount of attention over the last decade. In lightly supervised AM training, accurate training labels are automatically obtained from imperfect, manual transcriptions, often referred to as *loose transcriptions*. The first publications on lightly supervised AM training [1], [2], [3], [4], [5], [6] use closed captions of broadcast news (BN) as loose transcriptions. More recent works on lightly supervised AM training consider parliamentary sessions, e.g. European Parliament Plenary Sessions [7] or Japanese Congress meetings [8], together with their proceedings/meeting minutes as loosely transcribed training corpora.

The general approach, first proposed by [1], is to decode the loosely transcribed corpus with an initial ASR system and then to select the parts for training where automatic and loose transcriptions agree. The basic idea is that the selected parts are very likely to be accurate transcriptions. In [2], [3] the approach of filtering automatic transcriptions of BN with closed-captions is compared to an approach where the initial ASR system is simply biased towards the closed-captions, by including them in the language model. The authors of [2], [3] report that, given their setup, comparable results can be achieved with both approaches and that a combination leads to additional improvements. This stands in contrast to results reported in [6], where a filtering of automatically transcribed BN data leads to a degradation in transcriptions performance of the final recognition system. The conflicting results can be explained by the fact that the authors of [2], [3] consider a scenario where only very limited amounts of supervised training labels are available for the initial AM, leading to a high word error rate of more than 30% for the unfiltered ASR hypotheses. The authors of [6], however, use a very strong initial, biased ASR system that provides hypotheses with a WER below 10%. In particular,

it should be noted that the unfiltered ASR hypotheses used in [6] for training have a lower WER than the closed-captions.

In summary, lightly supervised AM training relies on ASR hypotheses as training labels. The hypotheses can provide information that is typically missing from the loose transcriptions: non-speech event markers, fillers and timing information. As AM training is very sensitive to erroneous training labels, it is important to **[a]** automatically transcribe the training corpus as accurately as possible, and **[b]** exclude erroneous hypotheses from training. Point **[b]** gains more importance the more corrupted the ASR hypotheses are.

To achieve point **[a]**, researchers make use of ASR systems that are overfitted to the task of decoding the training corpus **correctly**. Biasing the LM with the loose transcriptions is an effective tool for this. As proposed in [5], small, biased language models specific to sub-sets of the training corpus should ideally be created. In the context of closed-captions, individual broadcast news shows can for example constitute appropriate sub-sets. Applying small, very specific language models not only improves transcription quality, but also speeds up recognition by constraining the search space. However, as noted in [5], it is important to not bias the ASR system too strongly towards the loose transcriptions. After all, the recognizer is supposed to confirm correct parts in the loose transcriptions, and not repeat any erroneous parts. Most commonly, multiple iterations of decoding the training corpus and re-training the AM with filtered ASR hypotheses are applied. The idea here is that a re-trained AM should yield more accurate transcriptions when being re-applied to the training corpus, in turn leading to more training data for the next training iteration.

Point **[b]**, removing erroneous hypotheses, is achieved by filtering based on the loose transcriptions. For this, the word alignment between ASR output and loose transcription is computed, in the same manner as it is done for word error rate computation. Typically, only those regions where automatic transcription and loose transcription fully agree[1] are retained for training, with a constraint on the minimum segment size. Filtering has repeatedly been shown to beneficial, even though it can result in excluding large amounts of training data. For example, in the context of closed-captions, numbers for the amount of training data excluded via filtering range between 20–50% [5], [3]. However, the amount of filtering needs to be carefully tuned, as its benefits strongly depend on the quality of the ASR hypotheses and the loose transcriptions. While it is very important to not train on erroneous hypotheses, it is noted in [4] that providing correct training labels for exactly those parts where the ASR system makes recognition errors may be particular valuable. To provide training labels for such regions, [4] proposes to filter the n-best hypotheses (presented in form of confusion networks) instead of just considering the 1-best hypotheses during filtering.

[1]Apart from non-speech event markers and fillers.

III. Experimental Setup

A. Speech Corpora

The baseline acoustic model training corpus comprises 54 hours of carefully transcribed speech data, mostly from studio quality Cisco Television broadcasts. This speech data can be characterized as high-quality, with almost exclusively clean speech and high signal-to-noise ratios. Furthermore, it should be noted that we have access to the uncompressed speech signal. This stands in contrast to our loosely transcribed training corpus, where we only have access to audio that suffers from lossy compression via Advanced Audio Coding, at various bit-rates. The videos found in this training corpus are mostly Cisco internal presentations with often times highly disfluent speech and noisy environments. In total, there are 1,350 videos amounting to 977h of audio.

For our experiments, we make use of one development set and one test set that we had previously separated from the loosely transcribed corpus. For both sets, high quality manual transcriptions are available, in addition to the loose transcriptions. We refer to these sets in the following as *cscoDev* and *cscoTest*. In this context, it should be noted that none of our experiments make use of the *cscoTest* loose transcriptions in any way. We also report word error rates on one additional test set, referred to as *nonCsco*. As the name suggests, the videos in this set are non-Cisco domain videos from a Cisco customer. The set mostly consist of road-show type product presentations with very noisy acoustic environments and speakers that are not included in any of our training sets. Table I lists the amount of speech included in our development and test sets.

TABLE I
Development and evaluation data: amount of speech

	cscoDev	**cscoTest**	**nonCsco**
speech	4.9h	8.2h	1.9h

B. Loose Transcriptions: A Closer Look

The loose transcriptions available for our 977 hours of Cisco corporate videos are formatted for an easy readability, for example by representing numbers as digits and by making use of common abbreviations. We apply some basic text normalization to transform the loose transcriptions more towards their spoken form. This text normalization is limited to the spelling out of numbers and the expansion of abbreviations. Number conversion is accomplished with the help of the 'token-to-word' rules offered by the Festival Speech Synthesis System [9]. For the expansion of abbreviations, we use our own set of simple rules. After text normalization, the loosely transcribed corpus comprises 9.8 million running words. All numbers reported in this paper are based on normalized loose transcriptions.

One central question when dealing with loose transcriptions is how close the transcriptions are to the true transcript. We can answer this question with the help of *cscoDev*. Since we have the accurate transcription as well as the loose transcription available for this set, we can simply compute the WER of loose transcription relative to accurate transcription. Word error rate computation for the purpose of evaluating ASR system output typically ignores non-speech events and speech disfluencies. However, in the context of AM training it is important that these tokens are accurately labeled and they should therefore be included when computing the WER between loose transcription and accurate transcription. The first data row of table II therefore lists the WER of the loose transcription when including non-speech events and speech disfluencies in WER computation. The WER is 24.3%. The table also lists the error rates achieved after successively removing noise markers, fillers and word fragments from the accurate transcription reference. Using a 'cleaned' reference, the WER is 12.9%. A significant amount of errors stem from fillers and word fragments, which demonstrates the highly disfluent nature of our lightly-supervised training corpus.

TABLE II
LOOSE TRANSCRIPTION QUALITY: WER ON CSCODEV

reference	WER
accurate transcription	24.3%
− noise markers	19.4%
− fillers	13.7%
− word fragments	12.9%

In order to compare the quality of our loose transcriptions to the quality of loose transcriptions used in related work, we report following numbers. The authors of [6] list a WER of 10.3% for closed-captions of broadcast news, however, without specifying if non-speech events and speech disfluencies were considered during WER computation. For meeting minutes of Japanese parliamentary meetings, [8] reports an average WER of 15.5%, with half of the errors stemming from fillers and without considering non-speech events like breath or laughter.

C. Training and Decoding Setup

Acoustic model training is performed with HTK [10]. We employ a simple training setup featuring maximum likelihood training of speaker independent cross-word tri-phone models. The acoustic models are trained on features derived by perceptual linear predictive (PLP) analysis [11].

Language model training is accomplished with the SRI LM toolkit [12]. In all of our experiments, we use 3-gram language models with modified Kneser-Ney discounting [13], [14]. A description of the biased language models used to decode the loosely transcribed corpus is given in section IV. Our decoding LM, used to measure the performance of the re-trained acoustic models on the test sets, is estimated on the loose transcriptions (excluding *cscoDev* and *cscoTest*), the supervised training transcriptions, selected Gigaword sentences and on data collected from the world wide web. In total, the decoding LM is trained on 560 million running words over a 58k vocabulary. The decoding dictionary has 64k entries. Table III lists the out-of-vocabulary (OOV) rates and LM perplexities (PPL) on the used development and test sets.

TABLE III
OUT-OF-VOCABULARY RATES AND LM PERPLEXITIES

Data Set	PPL	OOV
cscoDev	128	1.6%
cscoTest	133	1.5%
nonCsco	198	3.3%

We automatically obtain speech segments by Viterbi decoding of a mel-frequency cepstral coefficients (MFCC) based feature stream using 1-state HMMs modeling speech, silence, noises and music. These 1-state HMMs have 128 Gaussian components each, and are trained on 37.5 hours of the clean Cisco Television data. The speech and non-speech segments obtained from Viterbi decoding are smoothed by applying minimum duration constraints and are padded by 0.3 seconds of silence.

Decoding is performed with the Juicer [15] WFST decoder. We use static ASR cascades, compiled with the help of the openFST [16] libraries and scripts provided by the Transducersaurus [17] project. Our cascades include a silence class transducer with the same topology as described in [18], [19]. We model noise and hesitations within the language model, using only uni-gram entries for both. Decoding consists of one decoding pass with speaker independent models.

IV. EXPERIMENTS AND RESULTS

Our experiments follow the general recipe for lightly supervised acoustic model training, as described in detail in section II. In summary, we decode the loosely transcribed corpus with biased ASR systems, filter their first-best system output with the loose transcriptions and re-train the acoustic model based on the filtered hypotheses. We repeat the process once, using the re-trained acoustic model for decoding the loosely transcribed corpus once more at a lower WER. Before employing the whole training scheme on the 977 hours of loosely transcribed videos, we tune the parameters needed for biasing and filtering on *cscoDev*.

A. Experimental Considerations

To bias the initial ASR system towards the loose transcriptions, we make use of small, video-specific language models and one training dictionary. The training dictionary is created from the combined vocabulary of the initial 54 hours of training transcriptions and the loose transcriptions, including the loose transcriptions for *cscoDev*. The resulting dictionary has 61k entries over a 53k vocabulary. Slightly more than half of the words in the training vocabulary, 27k words, are not included in our manual created background pronunciation dictionary. The pronunciations for these words are created

automatically, with the help of a grapheme-to-phoneme (G2P) translation system. We train this G2P system on the letter and phone sequences found in the COMBILEX dictionary [20], using the GIZA++ toolkit [21]. The final translation from letter sequences to phone sequences is done with the help of the Moses decoder [22]. With the help of a held out test set, we estimate the phoneme error rate for the G2P tool to be 2.6%. A large amount of the training dictionary words for which pronunciations have to be created automatically are acronyms. For these acronyms, we offer two pronunciations in the training dictionary, one based on the isolated letters of the acronym and one based on the full letter sequence. The biased, video-specific language models are created in the following manner. For each video, we create a small 3-gram language model based on the loose transcription for this video. We then interpolate this small language model with a more generic training language model. The generic training language model is estimated on the combined corpus of training transcriptions and loose transcriptions. This corpus comprises 10.4 million running words. We use a fixed interpolation factor that was estimated on *cscoDev*, see section IV-B.

B. Tuning on cscoDev

As described in detail in the previous section, we compute video-specific language models by interpolating a generic LM with small LMs that are solely trained on the loose transcription of the respective video. For the individual videos in *cscoDev*, we are able to automatically determine the interpolation weight of the small LM that minimizes the perplexity of the interpolated model. Table IV lists the respective interpolation factors, perplexities and word error rates for each of the 8 videos included in *cscoDev*. For comparison, we also list the PPL and WER achieved when using the more generic training LM. The use of video-specific LMs results in strong word error rate and perplexity reductions. The average optimal interpolation factor is 0.25. We use this fixed value when we create the video-specific language models for the videos found in the 977 hour training corpus.

TABLE IV
PPL AND WER REDUCTION BY USING VIDEO-SPECIFIC LMS

video	training LM		biased LMs		best lambda
	PPL	WER	PPL	WER	
1	91	60.1%	38	46.1%	0.21
2	134	25.3%	46	16.5%	0.2
3	113	67.5%	46	50.9%	0.2
4	104	41.2%	46	30.1%	0.26
5	86	33.5%	37	23.0%	0.26
6	92	33.9%	46	26.3%	0.37
7	116	18.4%	41	12.9%	0.2
8	109	40.5%	44	27.0%	0.27

From the results in table IV we see that even with video-specific language models, the achieved word error rates are relatively high and they vary strongly between the different videos. The total WER on *cscoDev* is 32.3%, with significantly higher word error rates on some videos, e.g. 50.9% on video number three. A filtering of the hypotheses seems therefore necessary. The same argument holds for the loose transcriptions. The comparable WER for the loose transcriptions is only 12.9%, as estimated in section III-A. However, directly training with these loose transcriptions, by force aligning them, would not just suffer from 'regular' word errors, but also from the fact that they do not including noise markers and fillers. As a reminder, when including these tokens, the WER of the loose transcriptions is 24.3%. For filtering erroneous ASR hypotheses, we compute the word alignment between automatic and loose transcription using NIST's *sclite* WER scoring tool. We use the word alignment to identify regions where loose transcription and automatic transcription are identical, ignoring noise markers and fillers present in the ASR output. We discard regions that are less than one second long and contain less than three words. The effects of filtering on the total amount of extracted data are shown in table V.

TABLE V
EXTRACTING TRAINING DATA FROM CSCODEV

System	WER	Extracted Data
generic training LM	43.8%	128min (44%)
video-specific LMs	32.3%	172min (59%)
retrained AM_{172min}	28.2%	186min (63%)

The table shows the WER of the ASR hypotheses and the amount of speech data left after filtering. The amount of speech data is given in minutes and as a percentage of the total amount of speech included in the development set. For comparison, the table also list the results when only using the generic training LM. Applying video-specific LMs increases the amount of training data left after filtering significantly, from 128 minutes to 172 minutes. In other words, 44% and 59%, respectively, of the total amount of speech available in *cscoDev* can be 'recovered' for AM training. Adding these 172 minutes to the baseline training corpus and re-training the acoustic model lowers the WER on *cscoDev* from 32.3% to 28.2%, even though the total amount of AM training data is only increased by 5.4% relative, from 53.8 to 56.7 hours. This strong reduction in WER, despite the small additional amount of training data, speaks for the strong data mismatch between the supervised and lightly-supervised training corpora. With a decreased WER of the re-trained AM, the amount of extracted training data further increases. This indicates that introducing at least one iteration when extracting training data may be useful. The results also give a good indication for how much training speech we can expect to recover from our 977 hours of loosely transcribed videos. Using video-specific LMs in combination with the baseline AM, we expect to recover approximately 59% of the 977 hours for AM training. Introducing one iteration, by applying an already re-trained AM for training data extraction, we expect to recover 63%. In fact, we expect to recover more than 63%. After all, the first iteration should already yield more than 500 hours of training data, resulting in an AM that no longer suffers from a training corpus that mostly consists

of mismatched speech data.

C. Correcting the Baseline Model

While not the main focus of this paper, we want to make one observation that is related to the reasoning used by [4] to motivate their filtering of n-best ASR hypotheses with loose transcriptions. The reasoning (see also section II) is that it may be particularly valuable to provide correct training labels for exactly those parts where the ASR system makes recognition errors. In table V we showed that using the more generic training LM, we are able to extract 128 minutes of lightly-supervised training labels. Using the video-specific LMs, we extract a higher amount of lightly-supervised training labels, 172 minutes in total (44 minutes more). While the generic training LM already provides a light bias, by including the loose transcription of *cscoDev* in an unweighted fashion, we can argue that its performance is somewhat close to the performance of a large, well-trained in-domain decoding LM. On the other hand, the video-specific LMs are strongly biased. This strong bias allows us to create, presumably correct, hypotheses for speech on which the generic ASR system would fail. We therefore propose to take a closer look at the training data that can only be extracted by the video-biased ASR systems. To do so, we identify those parts in the 172 minutes of training labels that are not already covered by training labels obtained with the generic training LM system. After identifying these regions, we extend their right and left boundaries by one word, if the neighboring word was also included in the training labels of the generic training LM system. The reasoning for this is that an incorrectly modeled cross-word context may have been responsible for the word error in the more generic system. Extending the boundaries results in a total of 56 minutes of data. We then add those 56 minutes to the supervised training corpus and re-train the acoustic model. The supervised training corpus has significantly more data, 54 hours in total. The 56 minutes therefore can only have a very limited impact on the resulting AM. For this reason, we arbitrarily weight the contribution of the 56 minutes by a factor of three. We also randomly select[2] 56 minutes worth of labels from the training data extracted with the generic training LM system, again weighting their contribution by a factor of three during the subsequent AM training step. Table VI lists the word error rates achieved with the trained acoustic models on *cscoTest* .

TABLE VI
IDENTIFYING MORE IMPORTANT TRAINING LABELS: WER ON CSCOTEST

	baseline	**+56min**	**+56min random**
WER	43.9%	42.9%	43.3%

All re-trained acoustic models achieve a lower WER than the baseline AM, and the AM based on the 56 minutes of data that can only be recovered with the video-biased ASR system achieves the lowest WER. While the difference in WER, compared to the best performing random-selection AM, is only 0.4% absolute lower, it is statistically significant at a level of $p = 0.001$. This result indicates that the those training labels are more important to the training process. Consequently, in the context of adapting a larger baseline acoustic model with a comparably small amount of loosely transcribed data, it may be helpful to give a higher weight to training labels that can only be extracted with the (strongly) biased ASR system.

[2]We repeated the random selection process three times and only present the best results.

D. Final Results

We apply the described approach for lightly supervised AM training on the full 977 hours of loosely transcribed videos, using the parameters we found to work best on *cscoDev*. Table VII lists the amount of training data extracted from the corpus.

TABLE VII
AMOUNT OF EXTRACTED TRAINING DATA

Baseline AM	**Iteration-0 AM**
565h (58%)	678h (69%)

It should be noted that the predictions that were based on the results obtained on *cscoDev* are accurate. Applying the re-trained AM on the corpus recovers 69% of the available speech data for training, compared to 58% when only using the baseline AM. The additional training data results in a significant decrease in word error rate, as shown in table VIII. This is true not only for the matched data conditions, on *cscoDev* and *cscoTest*, but in particular also on the *nonCsco* test set. All listed WER reductions, with the exception of the WER reduction on *nonCsco* between iteration 0 and iteration 1, are statistically significant. Overall, the extracted training data reduces the word error rate on *cscoTest* by 19.4% relative, from 43.9% to 35.4% and on *nonCsco* by 13.8% relative, from 55.9% to 48.2%.

TABLE VIII
WORD ERROR RATES FOR THE RE-TRAINED AMS

	Baseline	**Iteration-0**	**Iteration-1**
cscoDev	48.0%	38.9%	37.1%
cscoTest	43.9%	36.5%	35.4%
nonCsco	55.9%	48.4%	48.2%

We also try to identify potentially more 'important' training labels, similar to the experiments described in section IV-C. As before, we transcribe the training corpus once with video-biased ASR systems and once with an ASR system that only uses the more generic training language model. After filtering, this results in 565 hours and 437 hours of training data, respectively. We then mark the transcription segments from the 565 hours of training data that are not already included in the 437 hours of training data and extend the marked segments by one word, where possible (see section IV-C for details). This results in 188 hours worth of marked training labels. We then repeat the re-training of the iteration-0 AM (54 hours of supervised training data plus 565 hours of lightly

supervised training data). However, we now weight the marked segments more strongly during training. Table IX compares the resulting acoustic model to an acoustic model where we randomly marked 188 hours of training labels. In both cases, we arbitrarily set the weight to 3.

TABLE IX
ITERATION-0 AM: WER WITH WEIGHTED TRAINING LABELS

Weighting →	**none**	**targeted**	**random**
cscoDev	38.9%	38.3%	38.7%
cscoTest	36.5%	35.9%	36.3%
nonCsco	48.4%	48.6%	48.4%

The results show that on *cscoDev* and *cscoTest*, the AM trained with the described targeted weighting of training labels outperforms both, the unweighted baseline AM and the randomly weighted AM. This gives further indication that the subset of lightly-supervised training labels that was selected in this manner is more important to the AM training process.

V. SUMMARY AND DISCUSSION

We gave an overview of the state of the art in lightly supervised acoustic model training and described in detail our efforts on applying this technology in the context of corporate videos. We also reported initial results for a simple, yet effective scheme to identify a subset of lightly-supervised training labels that are more important to the acoustic model training process. Future work should address the question on how to weight these training labels optimally and how the scheme performs in the context of multiple training iterations. We also plan to investigate the effects of discriminative lightly-supervised acoustic model training.

Despite the highly disfluent speech present in the 977 hours of loosely transcribed corporate videos used for training, we achieve significant word error rate reductions of up to 19.4% relative over our baseline. It should be noted that our decoding setup includes only one decoding pass based on speaker independent models. This, together with the strong acoustic channel mismatch between our supervised training corpus and the test sets certainly contributes to the strong performance improvements. However, especially the strong reduction in word error rate on the non-Cisco domain task demonstrate the importance of the described lightly-supervised training techniques in the context of corporate videos.

ACKNOWLEDGMENT

Special thanks go to Steven Wegmann, Ananth Sankar and Sachin Kajarekar for help with HTK related questions, valuable discussions and proof-reading.

REFERENCES

[1] P. J. Jang and A. G. Hauptmann, "Improving Acoustic Models with Captioned Multimedia Speech." in *Proc. of ICMCS*, vol. 2, June 1999, pp. 767–771.
[2] L. Lamel, J. luc Gauvain, and G. Adda, "Lightly supervised acoustic model training," in *Proc. ISCA ITRW ASR2000*, 2000, pp. 150–154.
[3] L. Lamel, J. Gauvain, and G. Adda, "Investigating Lightly Supervised Acoustic Model Training," in *Proc. of ICASSP*, Salt Lake City, USA, May 2001.
[4] L. Chen, L. Lamel, and J.-L. Gauvain, "Lightly supervised acoustic model training using consensus networks," in *Proc. ICASSP*, 2004.
[5] L. Nguyen and B. Xiang, "Light Supervision in Acoustic Model Training," in *Proc. ICASSP*, 2004.
[6] H. Chan and P. Woodland, "Improving Broadcast News Transcription by Lightly Supervised Discriminative Training," in *Proc. ICASSP*, Montreal, Canada, May 2004.
[7] M. Paulik and A. Waibel, "Lightly Supervised Acoustic Model Training on EPPS Recordings," in *Proc. Interspeech*, Brisbane, Australia, September 2008.
[8] T. Kawahara, "Automatic transcription of parliamentary meetings and classroom lectures - A sustainable approach and real system evaluations," in *Proc. Chinese Spoken Language Processing*, Tainan, Taiwan, November 2010.
[9] A. Black and P. Taylor, "The Festival Speech Synthesis System," University of Edinburgh, Scotland, Tech. Rep., 1997, http://www.cstr.ed.ac.uk/projects/festival.html.
[10] S. Young, G. Everman, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, "The HTK Book," Cambridge University, Tech. Rep., 2006.
[11] H. Hermansky, "Perceptual Linear Predictive (PLP) Analysis of Speech," *The Journal of Acoustical Society of America*, vol. 87(4), pp. 1738–1752, 1990.
[12] A. Stolcke, "SRILM - An extensible language modeling toolkit." in *Proc. of ICSLP*, Denver, USA, September 2002.
[13] R. Kneser and H. Ney, "Improved backing-off for n-gram language modeling." in *Proc. of ICASSP*, Detroit, USA, May 1995.
[14] S. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," Harvard University, Tech. Rep., 1998.
[15] D. Moore, J. Dines, M. M. Doss, O. Vepa, O. Cheng, and T. Hain, "Juicer: A Weighted Finite State Transducer Speech Decoder." in *Proc. of Interspeech*, Lisbon, Portugal, April 2005.
[16] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri, "OpenFst: A General and Efficient Weighted Finite-State Transducer Library," in *CIAA 2007*, ser. Lecture Notes in Computer Science, vol. 4783. Springer, 2007, pp. 11–23, `http://www.openfst.org`.
[17] Transducersaurus - Tools for generating WFST-based ASR cascades. [Online]. Available: http://code.google.com/p/transducersaurus
[18] C. Allauzen, M. Mohri, M. Riley, and B. Roar., "A Generalized Construction of Integrated Speech Recognition Transducers." in *Proc. of ICASSP*, Montreal, Canada, May 2004.
[19] J. R. Novak, P. Dixon, and S. Furui, "An Empirical Comparison of the T3, Juicer, HDecode and Sphinx3 Decoders." in *Proc. of Interspeech*, Makuhari, Japan, September 2010.
[20] K. Richmond, R. A. J. Clark, and S. Fitt, "Robust LTS rules with the Combilex speech technology lexicon," in *Proc. of Interspeech*, Brighton, UK, September 2009.
[21] F. Och and H. Ney, "A Systematic Comparison of Various Statistical Alignment Models," *Computational Linguistics*, vol. 29(1), pp. 19–51, 2003.
[22] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation." in *Proc. of ACL*, Prague, Czech Republic, June 2007.