Incremental Unsupervised Training for University Lecture Recognition

Michael Heck[1], *Sebastian Stüker*[1], *Sakriani Sakti*[2], *Alex Waibel*[1], *Satoshi Nakamura*[2]

[1]International Center for Advanced Communication Technologies (interACT),
Institute for Anthropomatics, Karlsruhe Institute of Technology, Karlsruhe, Germany
[2]Augmented Human Communication Laboratory,
Graduate School of Information Science, Nara Institute of Science and Technology, Nara, Japan
{heck|sebastian.stueker|alexander.waibel}@kit.edu, {ssakti|s-nakamura}@is.naist.jp

Abstract

In this paper we describe our work on unsupervised adaptation of the acoustic model of our simultaneous lecture translation system. We trained a speaker independent acoustic model, with which we produce automatic transcriptions of new lectures in order to improve the system for a specific lecturer. We compare our results against a model that was trained in a supervised way on an exact manual transcription.

We examine four different ways of processing the decoder outputs of the automatic transcription with respect to the treatment of pronunciation variants and noise words. We will show that, instead of fixating the latter informations in the transcriptions, it is of advantage to let the Viterbi algorithm during training decide which pronunciations to use and where to insert which noise words. Further, we utilize word level posterior probabilities obtained during decoding by weighting and thresholding the words of a transcription.

Index Terms: lecture translation, spoken language translation, simultaneous translation

1. Introduction

Lectures at universities around the world are often given in the language of the country or region that the respective university is located in. At the *Karlsruhe Institute of Technology* (KIT), for instance, most lectures are held in German. This is often a significant obstacle for students from abroad wishing to study at KIT, as they need to learn German first. In order to be able to truly follow the often complex academic lectures, the level of proficiency in German that the foreign students need to reach is quite high.

While, in principal, simultaneous translations by human interpreters might be a solution to bridge the language barrier in this case, in reality this approach is too expensive. Instead, technology in the form of *spoken language translation* (SLT) systems can provide a solution, making lectures available in many languages at affordable costs. Therefore, one of our current research focuses is the automatic translation of university lectures [1][2], and thus aiding foreign students, by bringing simultaneous speech translation technology into KIT's lecture halls.

The simultaneous lecture translation system that we use is a combination of an *automatic speech recognition* (ASR) and a *statistical machine translation* (SMT) system. For the performance of such a *spoken language translation* (SLT) system the *word error rate* of the ASR system is critical, as it has an approximately linear influence on the overall translation performance [3].

Automatic speech recognition for university lectures is rather challenging. In order to obtain the best possible ASR performance, the recognition system's models, including *acoustic model* (AM) and *language model*, need to be tailored as closely as possible to the lecturer's speech and the topic of the lecture.

In this paper we investigate the unsupervised adaptation of the acoustic model of our simultaneous lecture translation to specific speakers. We start with a speaker independent acoustic model that has only seen very few or no data for the respective lecturer to which we adapt. With this model we produce automatic transcriptions of new lectures from one lecturer which we then exploit in order to improve the system for this lecturer. We further examined the impact of various ways of treating pronunciation variants and noise models during model training, as the decoding results on the training data contain those informations besides the hypothesized string of words. However, we will show that it is not necessarily the best strategy to directly use these informations as provided by the recognizer, and rather let the Viterbi algorithm during training decide where to use which pronunciations and when to insert additional noise words.

Similar to [4] we intended to evaluate the possible improvements of a system by unsupervised acoustic model training in dependency of the amount of training data. We share the same basic conditions, that no closely related texts were available for any kind of supervision. Similar to [5, 6], we made use of state confidence scores on word level. As a pre-processing step to unsupervised training, automatic transcriptions were filtered by using word posterior confidence scores for thresholding. Our training conditions can be compared to [7] where new data for retraining comes from the same speaker, channel and related conversation topics. Following the implications of [8] we add low confidence score data to the training, but unlike in other work we apply word-based weighting in order to compensate for errors, as it was done by [9] for acoustic model adaptation. The assumption is that erroneous data is helpful to improve system generalization. Unlike other work, e.g. [10], we refrained from a lattice-based approach.

2. Data

The experiments in this paper were conducted with the help of the *KIT Lecture Corpus for Speech Translation* [11]. The corpus consists of recorded scientific lectures that were held at the Karlsruhe Institute of Technology (KIT). Currently the corpus mainly contains computer science lectures, and a small amount of lectures from other departments and ceremonial talks.

2.1. Training Data

The speaker-independent system that we used in our experiments was trained on about 94 hours of speech from the lecture corpus. Our experiments were constrained to two distinct speakers. As training data we had 7.4 hours for *speaker A* and 8.3 hours for *speaker B* respectively, which had not been used for training the speaker independent system (see also Section 3).

2.2. Test Data

For *speaker A* we took one, for *speaker B* two recordings — 0.5h and 0.6h overall length respectively — from the available data as our test material. These recordings come from separate lectures than the remaining training data, so that we can actually simulate the way the training data would be used during the real operation of the lecture translator.

3. Experimental Set-Up

In our experiments we simulate the way an ASR system would work when being used in our simultaneous lecture translation system as it is deployed in KIT's lecture halls.

When the system starts to translate the lecture series of a new lecturer, only a generic, mostly speaker independent acoustic model will be available. With every new lecture given, new audio recordings of the lecturer become available, but no manual transcripts. The system will thus only be able to exploit these audio recordings to incrementally transform the speaker-independent acoustic model that is available at the beginning into a speaker-dependent model that fits the specific lecturer.

3.1. Speaker-Independent System

The speaker independent system used in our experiments was taken from the inauguration of the lecture translation system at KIT on June 11th 2012 [12]. For the inauguration, first a speaker-independent acoustic model system was trained on all available training data from the KIT lectures corpus, and then adapted to the individual lecturers.

The ASR system's pre-processing uses the warped minimum variance distortionless response (MVDR) [13] with a model order of 22 without any filter-bank. Vocal tract length normalization (VTLN) [14] was applied in the warped frequency domain. The mean and variance of the cepstral coefficients were normalized on a per-utterance basis. The resulting 20 cepstral coefficients were combined with seven adjacent frames to a single 300 dimensional feature vector that was reduced to 40 dimensions using linear discriminant analysis (LDA).

The acoustic model is based on HMMs using context dependent generalized quinphones with three states per phoneme, and a left-to-right topology without skip states. It uses a total of 4,000 models that were trained using *incremental splitting of Gaussians* (MAS) training, followed by *semi-tied covariances* training [15] and 2 iterations of Viterbi training.

The 4-gram language model used in our experiments was trained on texts from various sources like webdumps, newspapers and acoustic transcripts. The, in total, 28 text corpora range in size from about 5 MByte to just over 6 GByte [12].

4. Unsupervised Training Experiments

In order to adapt the speaker independent acoustic models to our test speakers, we used unsupervised training. For this, the training data of the respective speaker was automatically transcribed. With the help of word lattices, every word in the transcription is annotated with its posterior probability as a measure of confidence. On the transcriptions obtained this way we then performed one iteration of Viterbi training, starting with the speaker independent acoustic model.

In our experiments, described below, we investigated different ways of treating pronunciation variants and noise models in training, as well as different ways of making use of the confidence annotations.

We were also interested in the way that increasing amounts of available training data affect the word error rate. We therefore divided our training data into five chunks (2.29h, 3.05h, 3.81h, 4.57h, 6.87h) for *speaker A* and six chunks (0.99h, 2.17h, 3.44h, 4.79h, 6.23h, 7.68h) for *speaker B*.

We measured the word error rate of the resulting acoustic models on the test set of our test speaker. We decoded the speaker specific test sets with an offline set-up in a similar way decoding is performed in the lecture translation system, i.e., without lattice rescoring, in real-time, and with incremental VTLN and feature space constrained MLLR [16].

4.1. Baseline

A lower limit for the performance of the speaker dependent models that were trained on the unsupervised data is given by the performance of the speaker independent model. It gives a word error rate of 19.7% on our test *speaker A*, and 34.8%

on *speaker B*.

For *speaker A* we were able to estimate how effective the unsupervised training is by comparing our results against a model that was trained in a supervised way on an exact manual transcription of the training data. We expect that this will give us an upper limit for the results obtained from unsupervised training. Just like it was done for the systems for the lecture translation inauguration (see Section 3) we applied one Viterbi training iteration for our test speaker, resulting in a speaker dependent model. It gives a word error rate of 17.3% on the test speaker's test set. For *speaker B*, no exact manual transcriptions were available, rendering these respective tests a case study for the real life application of our system.

4.2. Treatment of Pronunciation Variants and Noise Words

In our first set of experiments we examined how to treat pronunciation variants and noise models in training. The intention of these experiments was to elaborate, whether additional information carried by the transcriptions is beneficial for the training process.

While the output of the recognition run on the training data contains pronunciation variants and noise words, we will see that it is not necessarily the best procedure to use them as provided by the recognizer. Instead it turns out that it is of advantage to let the Viterbi algorithm during training decide which pronunciations to use for the words during training, as well as, where to insert which noise words. This is done by inserting pronunciation variants as alternative paths to the base form of the words; noise words are inserted as alternative paths between regular words.

JANUS allows modifications on the generated hypothesis during label writing. Within the process of writing labels, the decoder chooses the most probable variant of a recognized word and autonomously inserts optional words into the current hypothesis. We experimented with four different ways of processing the decoder outputs of the automatic transcription with respect to the treatment of pronunciation variants and noise words:

recognition The annotation of noise words and pronunciation variants are taken as is from the recognition output and is not altered by the Viterbi training.

baseAll Pronunciation variants in the recognizer output are mapped to their base form, and the pronunciation variants used during training are picked by the Viterbi alignment in training. Wherever a noised word was hypothesized, all other noised words are inserted as alternative paths, and the actual noise word used for training is again picked by the Viterbi alignment.

baseWords Only regular words are mapped to their baseform and their pronunciation variants are inserted as alternative paths. The hypothesized noise words are left as recognized.

filtered All regular words are mapped to their base form, their pronunciation variants are inserted as alternative paths; all recognized noise words are removed, and instead inserted as alternative paths between regular words.

Filtering	Example
filtered	*wenn wir hier*
baseAll	*\$ wenn wir \$ hier*
baseWords	*\$(noise) wenn wir \$(breath) hier*
recognition	*\$(noise) wenn(1) wir(6) \$(breath) hier*

Table 1: Different filtering methods for pre-processing. *filtered* corresponds to plain text, *baseAll* contains general noise tags, *baseWords* is enhanced by annotations of pronunciation variants, and *recognition* resembles the unprocessed decoder output.

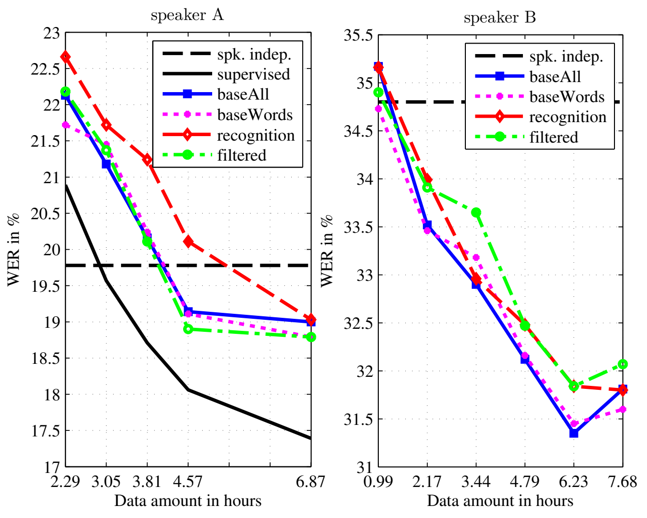

Figure 1: WER in % for the four different configurations for treating pronunciation variants and noise words with increasing amounts of training data.

Table 1 illustrates the different filtering methods that are applied to the automatic transcriptions before training. Figure 1 shows the resulting performance of the four configurations on the test speakers for increasing amounts of training data. The observations allow us to conclude that training on the exact transcriptions from the recognition run (*recognition*) are not the optimal choice. For speaker *A*, the improvement is significantly slower with increasing amounts of training data than for the other three methods. For speaker *B*, this type of transcriptions again does not lead to the optimal training performance. When using all available training material, differences between the filtering methods decrease. However, *recognition* type transcriptions can still not keep up with alternatively pre-processed annotations. Categories *baseWords* and *baseAll* perform about equally well, where the latter might be slightly more robust, as the performance curve as a function of the amount of training data tends to stay more stable in comparison to the one of *baseWords* and the other modalities' curves. It is interesting to see that configuration *filtered* seems to be beneficial for systems that already perform reasonably well, whereas for a weaker baseline performance other configurations are preferrable.

The baseline performance seems to have a noticeable impact on the effect of the adaptive training. For systems that already perform well, the improvements that can be expected tend to be lower than for systems that start with poorer recognition capabilities, according to the observations: For speaker *B*, speaker dependent models perform better than the speaker-independent models when at least 1h of training data is available, whereas for speaker *A* about four times as much data is necessary to see improvements in the same magnitude. The better the baseline performs, the more data is needed to observe first improvements. Ultimately and as expected, training on exact, i.e., manual transcriptions of the training data outperforms the unsupervised training, as can be seen for speaker *A*, where we had manual annotations at hand.

4.3. Confidence Weighting & Thresholding

The most common methods for processing unreliable, erroneous transcriptions in unsupervised acoustic model training are based on lattice confidence measures at word or state level [10]. In our experiments we utilized the word level posterior probabilities obtained during decoding. We utilized the confidences in three ways:

weighted Sets the gamma probabilities of the states of a word during Viterbi training to the posteriori probability of the respective word.

thresh Removes words with a confidence below a certain threshold from training.

weighted+thresh Combines both methods.

Given a Viterbi path through a built up HMM of a training utterance, a weighting factor *gamma* can be assigned to every frame prior to the update step for the model weights. If *gamma* is set to 0, parts of a path are effectively excluded from training. A $gamma \neq 0$ results in a weighted contribution of this particular frame to the training. Here, the *gamma* value corresponds to the confidence score $conf(w)$, e.g., the posterior probability of the word w to which a frame fr_i^w belongs, if weighting is applied. Thresholding with a factor *t* is performed by setting *gamma* to 0 for each fr_i^w with $conf(w) \leq t$.

Figure 2 shows the result of weighting with confidences and applying a threshold for our test speakers. Of these methods, the word-based weighting produces the better systems for both speakers: Weighting with the confidences gives the best performance, particularly when using all availabe data, whereas thresholding at least matches the performance of training on unfiltered data and at best gives slight advantages over not using confidences at all, in cases where a sufficient amount of training data is available. We randomly selected a threshold of 25% as lower limit for our experiments. Further experiments that are not represented in Figure 2 revealed, that the threshold should be used with care, as using too high a threshold discards too much data, and the performance suffers. Combining weighting and a threshold of 25% leads to the smoothest performance curve and has an effect on performance similar to weighting alone.

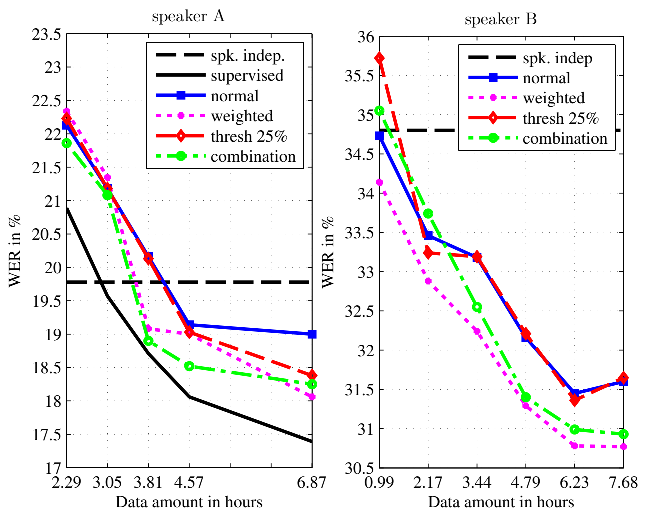

Figure 2: WER in % when weighting training data with confidences (*weighted*), excluding words from training by a posterior probability below a certain threshold (25%) (*thresh 25%*), and a combination of both (*combination*), compared to training without weighting and thresholding (*normal*).

5. Conclusion

In this paper we have described work on unsupervised adaptation of the acoustic model of our simultaneous lecture translation. We produced automatic transcriptions of new lectures with a speaker independently trained baseline system in order to improve the same for specific lecturers.

Evaluating four different ways of processing the decoder outputs led to the conclusion that it is of advantage to let the Viterbi algorithm during training decide which pronunciations to use and where to insert which noise words, instead of fixating the latter informations. The degree of detail for the transcriptions correlates with the baseline performance given a target speaker. *baseAll* and *baseWords* are beneficial when the baseline performance is lower, whereas *filtered* is better when the system already performs reasonably well, promoting the expectation that the Viterbi algorithm is able to make more accurate decisions with a better starting point, whereas additional information provided with the transcriptions helps when the baseline models show a lower performance. Speaker dependent models perform better than the speaker-independent models when at least 1h of training data is used. The Viterbi algorithm needs a certain amount of data so that training will succeed, where the amount required for performance gains correlates with the baseline performance.

Further, we utilized the word level posterior probabilities obtained during decoding by weighting and thresholding the words of a transcription. Combining word-based weighting and a threshold of 25% led to the smoothest performance curve as a function of the amount of training data. Our best systems in terms of WER reach an error rate of 18% and 30.7% for speakers *A* and *B* respectively, being trained on *baseAll* (*A*) and *baseWords* (*B*) processed transcriptions and *weighted* training. An additional *threshold* of 25% led to a competitive WER of 18.2% and 30.9% respectively. Obviously, weighting allows to cushion the influence of erroneously annotated training data, which is likely to have a lower confidence than potentially correct parts. The same explanation applies for the combination of weighting and thresholding, which leads to an even smoother convergence, whereas thresholding alone either excludes only few erroneous data, or even lots of correct data, depending on it's strictness. The winning techniques represent possible candidates for use in our simultaneous lecture translation systems as they combine fast convergence with good performance.

6. Acknowledgements

This work was supported in part by an interACT student exchange scholarship. '*Research Group 3-01*' received financial support by the '*Concept for the Future*' of Karlsruhe Institute of Technology within the framework of the German Excellence Initiative. The work leading to these results has received funding from the European Union under grant agreement no. 287658.

7. References

[1] C. Fügen, A. Waibel, and M. Kolss, "Simultaneous translation of lectures and speeches," *Machine Translation*, vol. 21, pp. 209–252, 2007.

[2] C. Fügen, "A system for simultaneous translation of lectures and speeches," Ph.D. dissertation, Universiät Karlsruhe (TH), November 2008.

[3] S. Stüker, M. Paulik, M. Kolss, C. Fügen, and A. Waibel, "Speech translation enhanced asr for european parliament speeches - the influence of asr performance on speech translation," in *Proceedings of the 2007 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4. Honolulu, HI, USA: IEEE, April 2007, pp. 1293–1296.

[4] L. Lamel, J. luc Gauvain, and G. Adda, "Unsupervised acoustic model training," in *Proceedings of the ICASSP 2002*, Orlando, Florida, USA, May 2002.

[5] C. Gollan, S. Hahn, R. Schlüter, and H. Ney, "An improved method for unsupervised training of lvcsr systems," in *Proceedings of the INTERSPEECH 2007*, Antwerp, Belgium, August 2007.

[6] T. Kemp and A. Waibel, "Unsupervised training of a speech recognizer using tv broadcasts," in *Proceedings of the EUROSPEECH 1999*, Budapest, Hungary, September 1999.

[7] G. Zavaliagkos, M.-H. S. abd Thomas Colthurst, and J. Billa, "Unsupervised acoustic model training," in *Proceedings of the ICSLP 1998*, Sydney, Australia, November 1998.

[8] H. Li, T. Zhang, and L. Ma, "Confirmation based self-learning algorithm in lvcsr's semi-supervised incremental learning," *Procedia Engineering*, vol. 29, pp. 754–759, 2012.

[9] C. Gollan and M. Bacchiani, "Unsupervised acoustic model training," in *Proceedings of the ICASSP 2008*, Las Vegas, NV, USA, March 2008.

[10] T. Fraga-Silva, J.-L. Gauvain, and L. Lamel, "Lattice-based unsupervised acoustic model training," in *Proceedings of the ICASSP 2011*, Prague, Czech Republic, May 2011.

[11] S. Stüker, F. Kraft, C. Mohr, T. Herrmann, E. Cho, and A. Waibel, "The kit lecture corpus for speech translation," in *LREC 2012*, Istanbul, Turkey, May 2012.

[12] E. Cho, C. Fügen, T. Hermann, K. Kilgour, M. Mediani, C. Mohr, J. Niehues, K. Rottmann, C. Saam, S. Stüker, and A. Waibel, "A real-world system for simultaneous translation of german lectures," interACT, Karlsruhe Institute of Technology, Tech. Rep., December 2012.

[13] M. Wölfel and J. McDonough, "Minimum variance distortionless response spectral estimation, review and refinements," *IEEE Signal Processing Magazine*, vol. 22, no. 5, pp. 117–126, September 2005.

[14] P. Zhan and M. Westphal, "Speaker normalization based on frequency warping," in *ICASSP*, Munich, Germany, April 1997.

[15] M. Gales, "Semi-tied covariance matrices for hidden markov models," Cambridge University, Engineering Department, Tech. Rep., February 1998.

[16] M. Gales, "Maximum likelihood linear transformations for hmm-based speech recognition," Cambridge University, Engineering Department, Tech. Rep., May 1997.