Semi-supervised speaker adaptation

Silke Goronzy, Krzysztof Marasek, Ralf Kompe

Sony International (Europe) GmbH, Advanced Technology Center Stuttgart (ATCS)
Home Network Company Europe, Man Machine Interfaces
TEL: +49-711-5858-456, FAX: +49-711-5858-199, E-mail: goronzy@sony.de

Abstract

We developed powerful unsupervised adaptation methods for speech recognition, i.e., the system improves its performance while the user uses it. No prior enrollment phase is necessary where the speaker has to read a given text. We tried to further improve the unsupervised adaptation by using confidence measures. These give an estimate of how likely the recognized words were correct. Adaptation then only is conducted on utterances where the recognition result was reliable. This avoids the adaptation of wrong models in case of misrecognitions. As a confidence measure we use a set of features that comprise features related to phoneme durations, acoustic scores etc. These feature are collected during recognition and from the recognition result and a neural net is used as a classifier.

1 Introduction

Most of state-of-the-art speech recognition systems use speaker adaptation techniques to improve their performance. While many applications use supervised, batch adaptation, where the user has to read a predefined text prior to using the system, we worked on unsupervised, online adaptation. Here, no enrollment phase is necessary, but instead the system is adapting to the user while he is using the system. So he is not realizing that the adaptation is running. In online adaptation the utterance is recognized as usual, adaptation is conducted and the modified models are used for recognizing the next utterance and so on, so that a steady performance improvement can be observed. The disadvantage of this online, unsupervised approach is that the text that was spoken is not known, but the recognizer output has to be used instead. If utterances are misrecognized, the wrong models will be adapted, which may lead to a performance decrease if it happens repeatedly. This is especially a problem in noisy environments, where the initial SI recognition rates are normally lower than in clean environments. As a measure of how reliable the recognition result was we use confidence measures (CMs). CMs use knowledge, which could not directly be used during recognition, because the computational cost would be too high. So features are extracted during search and from the N-best output of the recognizer and are used to compare the N-best recognition results. In our semi-supervised adaptation approach we then only use those words for adaptation that were classified correctly by the CM. We thus avoid to adapt to misrecognized words and use only those that were recognized correctly with high confidence. The features that are used to formulated the CM are related to phone durations, speaking rate and the acoustic score. As a classifier we use a neural net (NN).

2 Unsupervised Speaker Adaptation

Our previous work, cf. [1] has shown that it is beneficial to combine Maximum Likelihood Linear Regression (MLLR) and Maximum A Posterior (MAP) adaptation.

2.1 MLLR

In MLLR adaptation a transformation matrix A is estimated from the adaptation data so as to maximize its likelihood. The matrix is then used to update the means of the HMM parameters, specifically the means μ of the Gaussian densities:

$$\mu_i = A * \mu_{i-1} \tag{1}$$

i denotes the current utterance. In our approach we use one single transformation matrix to update all means of the system, thus achieving a very fast but broad speaker and channel adaptation. We extended the original MLLR approach cf. [2] by introducing a dynamic weighting scheme for updating the means, that allows fast adaptation on a very limited amount of adaptation data, cf. [3].

2.2 MAP

MAP achieves a finer adaptation compared to MLLR since it updates each of the means separately. The update is performed as follows:

$$\mu_i^{(n)} = \frac{\tau_i^{(n-1)}\mu_i^{(n-1)} + c_i\overline{x_i}}{\tau_i^{(n-1)} + c_i} \tag{2}$$

τ is a weighting factor, c_i counts how often Gaussian i has been observed and $\overline{x_i}$ is the average of the observed feature vectors. Again we extended the original approach, cf. [4]. Though being much more accurate and achieving near speaker dependent (SD) performance if a lot of adaptation is available, this approach is quite slow.

In order to exploit the capability of MLLR to quickly adapt on a very limited amount of adaptation data and the capability of MAP to reach SD performance on the long run, we combined these two methods, such that for the first utterances MLLR is conducted. MLLR is switched off and the MLLR-adapted models are used as prior information for the further MAP adaptation.

3 Confidence Measures

3.1 Phone-Duration-Based Features

It can be observed that when a word is misrecognized, there is a severe mismatch between the segmentation (and thus the phoneme durations) found by the recognizer and the durations that can be found in the training data. This motivated the use of some new features that are related to phoneme durations. During the training the distributions of durations of all phones are determined, based on a forced alignment of the training data. The durations are additionally smoothed and are later compared to the phoneme durations that were determined by the alignment of the recognizer during testing. Since it is well known that the speaking rate strongly influences the phoneme durations, we estimated the speaking rate and normalized the found durations accordingly. The speaking rate was estimated as follows, cf. [5]:

$$\alpha = \frac{1}{N} \sum_{i=1}^{N} \frac{d_i}{\overline{x_p}} , \qquad (3)$$

where N denotes the number of phones in the observed utterance and/or in the past few utterances, d_i is the duration of the *i-th* phone segment (recognized as phone p) in the utterance and $\overline{x_p}$ is the mean length of the corresponding phone p learned during training.

Some of the features were multiply used, e.g. not normalized, normalized by number of frames, by acoustic score of the best hypothesis or by the speaking rate.

The feature set comprises the following features:

1. *n_toolong01, n_toolong05*: Number of phones in the best hypothesis that are longer than the 0.01 and 0.05 percentile, respectively, compared to the training data

2. *n_tooshort01, n_tooshort05*: See above for too short durations

3. *sequence*: Number of sequences of phone pairs within one utterance where the first phoneme was too long and the second one too short (or vice versa) (using the 0.01 percentiles).

4. *avg_tempo*: The average speaking rate

5. *stdev_tempo*: The standard deviation of the speaking rate (w.r.t. to average speaking rate of last *n* utterances)

6. *diff_tempo*: The absolute difference between the average and the actual speaking rate.

7. *tempo*: Current speaking rate

To show the relation between the features chosen and the correct/misrecognized classification we show some box-and-whiskers plots. Box-and-whiskers plots are a way to look at the overall shape of the data. The central box shows the data between the 'hinges' (roughly quartiles), with the median presented by a line. 'Whiskers' go out to the extremes of the data, and very extreme points are shown by themselves, cf. [6]. We show the plots of the features *n_toolong05, n_tooshort05 and avg_tempo* in Figures 1, 2 and 3, respectively. Although several outliers are present we can see a good distinction capability between the correct/misrecognized classes of the corresponding feature.

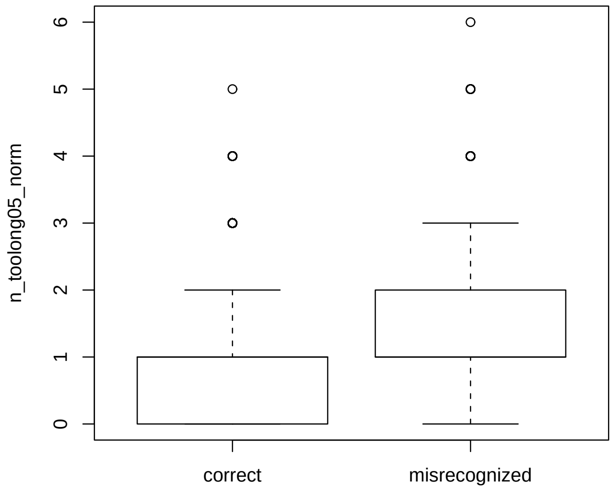

Figure 1: box-and-whiskers plot for feature n_too_long05 normalized by the number of frames

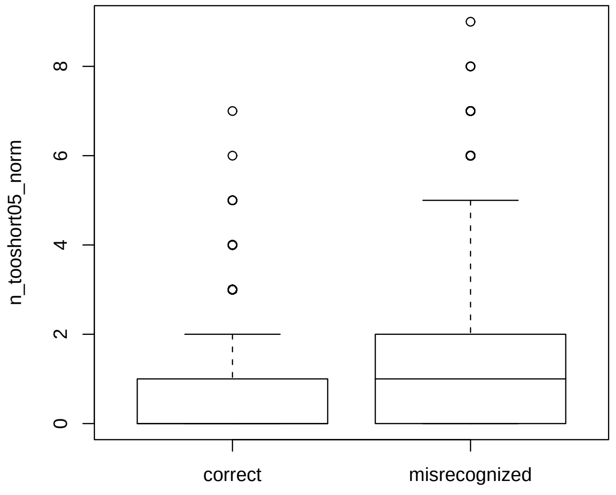

Figure 2: box-and-whiskers plot for feature n_too_short05 normalized by the number of frames

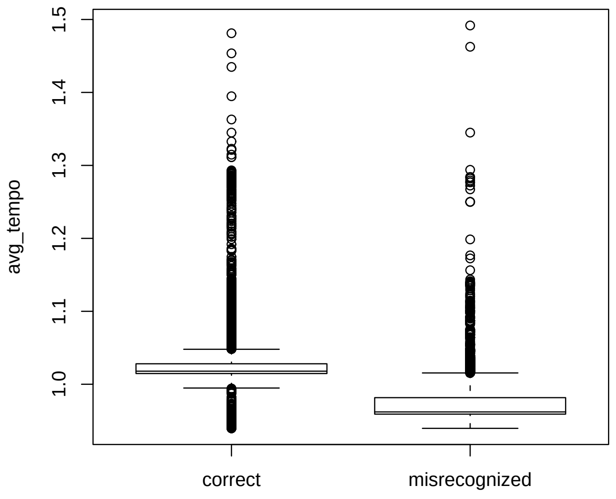

Figure 3: box-and-whiskers plot for feature avg_tempo

3.2 Additional Features

In addition to the duration-based features described above we used some more features that have proven to be useful for CMs in the past. These consist of the following:

1. *n_frames*: Total number of frames of the utterance
2. *n_phones*: Total number of phones of the utterance
3. *n_frames_nosil*: Total number of frames (without silence)
4. *first_score*: Acoustic score of the best hypothesis
5. *first_second*: Difference in acoustic scores between the first and second-best hypothesis
6. *avg*: Average acoustic score for the N-best hypotheses
7. *first_avg*: Difference between *first_score* and the average score
8. *first_last*: Difference between the first and last-best hypothesis
9. *first_beambest*: For each frame and all active states the distances between the Gaussians and the feature vectors are computed. The best possible distance, i.e., the minimal one in the beam, is compared to the distance found for the state belonging to the best state sequence.
10. *first_beambest_zeros*: The number of frames for which the score difference (see *first_beambest*) is zero
11. *first_beambest_largest*: The largest continuous difference between the first-best and the best state sequence in beam
12. *best*: The best possible score in the beam
13. *first_best*: See *first_beambest*, taking into account the transition probabilities.
14. *worst_phonescore*: The worst phone score in the best hypothesis
15. *avg_phonescore*: The average phone score in the best hypothesis
16. *stdev_phonescore*: The change of score within one phone
17. *worst_phone_best*: The difference between the best possible and worst phoneme score in the best hypothesis
18. *worst_frame_score*: Worst frame score for all frames in the best hypothesis
19. *best_in_beam*: The sum of the differences between the best frame scores for the hypotheses in the beam and of the best hypothesis
20. *snr*: signal-to-noise ratio

We again exemplarily show a box-and-whiskers plot of one of the features, the normalized *worst_phonescore* in Figure 4. Again a good separation between the correct/misrecognized classes can be observed.

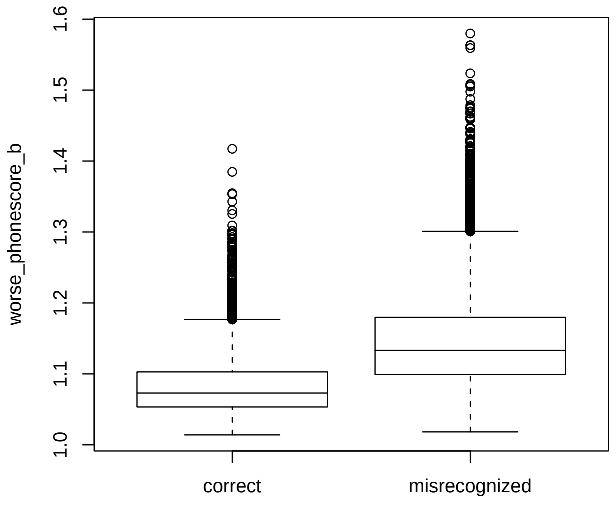

Figure 4: box-and-whiskers plot for feature worst_phonescore

	# patterns	corr	wrong	OOV
train	37718	18859	3022	15837
eval	250	1795	330	2125
test	16186	7071	1022	8093

Table 1: Number of patterns used for training, evaluation and testing of the NN classifier

4 Classification

For testing the 46 derived features we constructed a feature vector for each utterance of the command&control test task and used a NN [1] to classify the recognition result as either correct or wrong. For the NN we used a feed forward net, that consisted of one hidden layer only. We used 46 input nodes, 8 hidden nodes and 2 or 3 output nodes, respectively. The training data for the NN comprised clean speech only and also included a large amount of OOV words. The detailed statistics concerning the patterns used for training, evaluation and testing can be found in Table 1. One pattern corresponds to the features extracted from one utterance. The data for training the NN was obtained using our standard recognizer, that will be described in more detail in section 5.1. We used different data for training the monophone models for the recognizer than we used for training the NN. Since we use some features related to the acoustic score, this was necessary to avoid a possible influence of the training data on the acoustic score if the same data was used for training the monophone models and determining the reference durations. The NN training data was labelled as being correctly recognized or not. So the target output for the 2 output nodes of the NN were either '1 0' or '0 1', '1 0' means the utterance was correctly recognized and '0 1', it was misrecognized.

In a second set of experiments we use a NN with 3 output nodes. The first two output nodes have the same meaning as before, the third output node is to indicate, whether a misrecognized word was an OOV word or not ('1' means OOV). So possible outputs (in the training data) are '0 1 1' or '0 1 0' in case of a misrecognition. For correctly recognized words only '1 0 0' is possible. During testing the NN

[1]we used SNNS to build our nets, [7]

	CER	C_a	F_a	C_r	F_r
baseline (SI)	-	43.7	56.3	-	-
2 out	9.67	38.45	4.62	51.55	5.05
3 out	10.29	37.9	4.62	51.62	5.67

Table 2: Classification results of a NN with 46 input nodes, one hidden layer with 8 nodes and 2 and 3 output nodes, respectively

outputs values between 0 and 1. The final decision threshold for the values between 0 and 1 is then simply 0.5. This means that if the first output is greater than 0.5 and the second is smaller than 0.5, the utterance is considered as being correctly recognized. If both values are greater or smaller then this threshold, the utterance cannot be classified at all. This happened in 0.3% of the cases only. Correspondingly, if the third output is greater than 0.5, the word is classified as being OOV.

5 CM Experiments and Results

5.1 Experimental Setup

Training and testing were conducted using a German command&control isolated word data base recorded in our sound treated studio. It consists mainly of isolated words and short phrases. The vocabulary size was 375. The speech was sampled at 16 kHz and coded into 25 ms frames with a frame shift of 10 ms. Each speech frame was represented by a 38-component vector consisting of 12 MFCC coefficients (no energy) and their first and second time derivatives. The first and second time derivatives of the energy are also included. We trained 3-state, monophone HMM models with 1 Gaussian per state using 34281 utterances from 80 speakers. The choice of such a simple model was due to the memory and speed requirements of a command&control application. The corpora we used for testing were a German address corpus with approximately 23 utterances per speaker and a command&control corpus with approximately 234 utterances per speaker, so around 260 utterances per speaker in total. We then added the same number of OOV utterances for each speaker, resulting in a total number of 540 utterances per speaker. The test set consisted of 35 speakers.

5.2 NN results

When using the full set of features, we achieved the results that are listed in Table 2. The performance of our CM was measured in terms of classification error rate (CER), which is the number of misclassified patterns divided by the number of total patterns. The first row shows the results for the baseline speaker independent (SI) system. Also we listed the correct and false alarm rates (C_a and F_a, respectively) and correct and false rejection rates (Cr and Fr, respectively). The C_a rate directly corresponds to the recognition rate of the SI system. The relatively low initial recognition rates of 43.7% can be explained by the fact that we included 50% of OOV utterances into the test set.

False alarms are those utterances that have been classified as being correctly recognized although they were misrecognized, so this corresponds to the WER of the SI system. Both NNs correctly reject (C_r) more than 50% of the utterances (remember that the more than 50% OOV were

	corr	wrong
3 out	88.6	11.4

Table 3: OOV classification results (in %) for the NN with 3 outputs, determined on the C_r-cases only

	CER	C_a	F_a	C_r	F_r
baseline (SI)	-	43.7	56.3	-	-
ac feat only	9.03	39.48	5.15	50.8	3.88
ac + spk rate	8.95	37.39	3.73	51.19	5.21
dur feat only	16.4	36.09	8.85	47.47	7.57
all	10.79	37.5	4.6	51.72	6.19

Table 4: Classification results (%) for different subsets of features

included in the test set). Although this is at the cost of rejecting 5.1% of the correctly recognized ones (F_r). 4.6% of the utterances were accepted although they were misrecognized (F_a). Applying this CM to the SI system would reduce the WER by more than 90% (from 56.3 to 4.6%). All utterances that were classified as unreliably recognized would be rejected and the user could be re-prompted for these utterances.

The results for the NN with 3 output are slightly worse. However, the 3-output net provides us with further information. This information is shown in Table 3. For all cases, in which an utterance was classified as misrecognized, we tried to judge, whether they were OOV utterances or not. In 88.6% of these (C_r-)cases of the baseline system the NN classified OOV words correctly.

In dialogue systems it could be beneficial for the course of the dialogue, to not just classify a word or sentence as being misrecognized, but to also know if it was an OOV word or not. This knowledge can greatly influence the following dialogue steps.

If we look at the large number of OOV patterns in the test set, we see that 88.8% of the 'misrecognized' patterns were OOV words. So if we simply classify all misrecognized words as OOV, we would be wrong in 11.2% of the cases on this test set, which is almost the same result delivered by the NN. So in this case our NN does not seem to be better than guessing. But since we do not detect 100% of the misrecognized words correctly, this cannot be directly compared. Furthermore when testing the CM in an online demonstration system, we saw that the OOV rejection works reasonably.

To assess the contribution of the prosodic features, we also trained NNs with 13, 31 and 35 input features, respectively. The 35 features comprised all acoustic-score based features and the features related to the speaking rate. The 31 features included the acoustic score-related features only. The 13 features were those related to the phoneme durations and speaking rates. The results are given in Table 4. As can be seen there the CER for the acoustic purely score-related features are the lowest followed by all features, the acoustic and tempo features and the purely duration features. Combining the duration-based features with the acoustic score-based ones unexpectedly does not yield any improvements. It seems that all information captured in the phonemes durations is also kept in the acoustic score related features. However, taking the duration-based features alone still shows acceptable performance. One major disadvantage of the acoustic score-based features is that they strongly depend on the topology of the recognizer, the front-end, etc. So whenever there is a change in one of the above mentioned parameters, a new training of the NN would become necessary. On the contrary the duration-based features are independent of these parameters. So the use of these features would make the CM independent of the recognizer, which is a big advantage if it is to be used for different applications or on different platforms.

6 Semi-supervised Speaker Adaptation

Using misrecognized utterances for adaptation results in a modification of the wrong models. Using such utterances for MLLR adaptation does not seem to be a big problem, because all models are always updated using a global transformation matrix. But especially for MAP adaptation, where only the parameters of those HMMs that were observed in the adaptation data are updated, misrecognitions can have an adverse effect on adaptation performance. In such a case, the wrong models will be adapted, leading to a possible decrease in recognition rates. Ideally we would thus only use those utterances for adaptation, that were recognized correctly. The approach we are proposing uses the CM, that was introduced in section 3 to judge how reliable a word was recognized. The word is recognized as usual and after recognition the CM is applied. If it was a reliable word, it will be used for adaptation, otherwise no adaptation for this utterance will be conducted. We call this method semi-supervised adaptation.

7 Experiments and Results

For our adaptation experiments we used 6 speakers that were recorded in a clean studio environment. The settings and the preprocessing were the as described in section 5.1. For each of these speakers we have approximately 2000 utterances that were split into 10 sets of 200 utterances each. Set 10 was used for testing always. We tested the SI models, the unsupervised adaptation in which each utterance, no matter if it was misrecognized or not, was used for adaptation and semi-supervised adaptation. We tested after different numbers of utterances ranging from 5 to 2000. Table 5 exemplarily shows the results in word error rate (WER) for the results after 15 and 600 utterances. The results for other numbers of utterances are not explicitly shown, since no significant changes were observed. It can be seen, that our unsupervised adaptation approach outperforms the SI models already for very small amounts of adaptation data. When more utterances are available, the results can be further improved and the initial WER can be reduced by 40.3%. When the semi-supervised adaptation is used, no improvements compared to the unsupervised approach can be observed. We added the results for a simulated 'perfect CM', in which we took for the adaptation only those utterances that were correctly recognized using the SI models. This is comparable to supervised adaptation were it is known, what was actually said. The testing set of course remained unchanged and comprised also misrecognized utterances. Also the results for speaker dependent models are shown. They achieve the lowest error rates.

Even using this perfect CM for supervision does not yield any improvements as can be seen in Table 5. We thus conclude, that since the number of erroneous utterances that were used for unsupervised adaptation was quite small (the initial WERs are around 13%), this does not have an adverse effect on performance. Or at least the positive effect of using only correctly recognized utterances is nullified by the reduced number of utterances (resulting from the rejection of unreliable utterances). On the other hand the results show, that our online and unsupervised adaptation approach is quite powerful and robust against misrecognitions.

	15	600
SI	22.8	13.4
unsupervised	21.9	8
semi-supervised	22.8	8.1
supervised	-	8.1
SD	10.5	3.4

Table 5: WER Results for the baseline SI system, unsupervised, semi-supervised and supervised adaptation after 15 and 600 utterances

8 Conclusion

An approach for semi-supervised speaker adaptation was presented. As a means of supervision we used CMs, so that only utterances that were reliably recognized were used for adaptation. Our CM consists of a NN classificator that uses a set of features for decision making. These features were mainly related to phoneme durations, speaking rate and acoustic score. Even though we achieve classification error rates of around 10% using our CM, the results when applying this CM to speaker adaptation can not be improved compared to the unsupervised adaptation. While the unsupervised approach reduced the WER by 40.3%, the semi-supervised approach did not yield any further improvements. However even supervised adaptation could not improve the results. This lets us conclude that for clean speech an such low initial low error rate as were considered here, the few misrecognitions do not really harm the unsupervised adaptation. This demonstrates the robustness of our online, unsupervised adaptation approach. Tests on noisy data, which usually exhibits higher initial error rates are necessary to assess the influence of higher WERs on speaker adaptation and thus the effect of applying a CM before conducting adaptation.

References

[1] S. Goronzy and R. Kompe. A Combined MAP + MLLR approach for speaker adaptation. In *Proceedings of the Sony Research Forum 99*, volume 1, pages 9–14, 1999.

[2] C. J. Leggetter and P. C. Woodland. MLLR for Speaker Adaptation of CDHMMs. *Computer Speech and Language*, 9:171–185, 1995.

[3] S. Goronzy and R. Kompe. A MAP-like weighting scheme for MLLR speaker adaptation. In *6th European Conference on Speech Communication and Technology*, volume 1, pages 5–8. European Speech Communication Association (ESCA), 1999.

[4] Q. Huo and C.-H. Lee. On-line Adaptive Learning of the CHMM Model Based on Approximate Recursive Bayes Estimate. *Transactions on Speech, Audio Processing*, 5(2):161–172, March 1997.

[5] Ralf Kompe. *Prosody in Speech Understanding Systems*. Springer Verlag, 1997.

[6] Venables and Ripley. *Modern Applied Statistics with S-Plus*. Springer Verlag, 1997.

[7] Institute of Parallel and Distributed High Performance Systems, University of Tübingen, Department of Computer Architecture. *SNNS, Stuttgart Neural Network Simulator, User Manual v4.2*, 1998. http://www-ra.informatik.uni-tuebingen.de/SNNS.