Interspeech 2018
2-6 September 2018, Hyderabad

Domain-Adversarial Training for Session Independent EMG-based Speech Recognition

Michael Wand[1], *Tanja Schultz*[2], *Jürgen Schmidhuber*[1]

[1]Istituto Dalle Molle di studi sull'Intelligenza Artificiale (IDSIA),
USI & SUPSI, Manno-Lugano, Switzerland
[2]University of Bremen, Bremen, Germany
{michael,juergen}@idsia.ch, tanja.schultz@uni-bremen.de

Abstract

We present our research on continuous speech recognition based on Surface Electromyography (EMG), where speech information is captured by electrodes attached to the speaker's face. This method allows speech processing without requiring that an acoustic signal is present; however, reattachment of the EMG electrodes causes subtle changes in the recorded signal, which degrades the recognition accuracy and thus poses a major challenge for practical application of the system. Based on the growing body of recent work in domain-adversarial training of neural networks, we present a system which adapts the neural network frontend of our recognizer to data from a new recording session, without requiring supervised enrollment.

Index Terms: Silent Speech interface, Neural Networks, EMG-based Speech Recognition, Domain Adaptation

1. Introduction

The field of *biosignal-based* speech recognition, i.e. speech recognition without making use of the acoustic signal, has seen a major surge of interest during the past decade, propelled by innovative recording technologies, increasing computational power, and the needs of an ageing population and an ever-mobile society: the latter may profit from the possibility to communicate silently and confidentially in public places, the former can potentially use speech capturing devices based on biosignal technology to overcome speaking disabilities [1].

In this paper, we consider a speech recognizer based on *surface electromyography* (EMG): Small electrical currents which emerge as a byproduct of human muscular activity are captured by electrodes attached to the subject's face, thus allowing to communicate even when no acoustic signal is available or can be captured. This device is lightweight and mobile, does not require intrusive wiring, and allows continuous speech recognition [2] as well as real-time speech resynthesis [3]. In this paper we consider only EMG-based speech *recognition*.

The myoelectric signal varies when the recording electrodes are removed and reattached, even when the speaker is the same: i.e. the signal varies across *sessions*. Creating a truly session independent EMG-based speech recognizer has been a long-standing goal [4, 5]. In this paper we use state-of-the-art neural network methods to achieve improved session independency using unsupervised adaptation: Based on our previous work [6], where a Deep Neural Network Frontend is combined with HMM-based sequence modeling, we augment the frontend part with *Domain-Adversarial training* [7] to adapt the system to data of a *target* session without requiring any supervised training data from the target session at all.

2. Related Work

Biosignal-based speech recognition is an active field of research, with very diverse modalities and methods under investigation [1, 8, 9]. Currently, the most promising concepts are *(Permanent) Magnetic Articulography* (PMA) [10], where small magnets glued to or even implanted into the articulators yield a high-quality representation of the underlying speech; visual methods [11], which may be augmented by ultrasound imaging [12]; and the electromyographic approach [13, 14, 3], which is less intrusive than PMA and allows very lightweight signal capturing devices. A major line of research is real-time speech reconstruction [10, 3, 15], which plays a major role for practical communication scenarios, but also for user feedback.

Major landmarks of EMG-based speech processing include continuous speech recognition with phone [2] and *Bundled Phonetic Feature* (BDPF) [16] models, vocabulary-free direct speech synthesis from the speech signal [17, 18, 3], and in the context of this paper, supervised [4] and unsupervised [5] session adaptation using Gaussian mixture models.

(Deep) artificial neural networks (DNN), first used in speech recognition in the 1990's [19], have by now become a de facto standard. The first systems used *hybrid* approaches: DNNs are used as a frontend for computing local probabilities, which are subsequently fed into an HMM-style search algorithm [20]. More recent systems model the entire processing pipeline from (usually spectral) input features to phone-level or even word-level output by recurrent neural networks [21].

Domain adaptation / transfer learning in neural networks has been investigated in a variety of contexts, in particular in image recognition. The usual assumption is that labeled data from the *source* domain and unlabeled data from the *target* domain are to be used to train a classifier which performs well on the target domain. A common approach, which is also used in this paper, is to encourage similar representations of source data and target data at some intermediate layer (see section 4.3 for details on our specific implementation), this similarity can be measured and enforced directly [22, 23], or indirectly with an *adversarial* network [7]. Further methods include the training of even more complex network architectures, attempting to factorize the information contained in the input data in different ways [24, 25, 26].

3. Data Corpus and Feature Extraction

We use the EMG-UKA Corpus [27], which is currently the largest available data corpus of myoelectric recordings of speech. It consists of a total of 7:32 hours of EMG and acoustic speech recordings in three speaking modes (audible, whispered, and silent), of which we only use the audible part (this allows us

10.21437/Interspeech.2018-2318

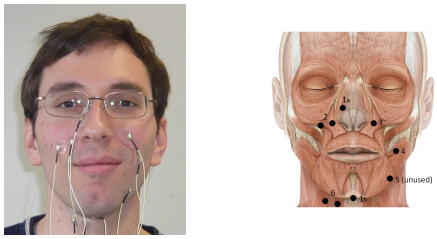

Figure 1: *Electrode positioning (from [27]) with chart of the underlying muscles (muscle chart adapted from [28])*

to concentrate on session discrepancies, see ref. [29] for a treatise on speaking mode differences). Figure 1 shows the recording setup [30]: A total of 6 EMG channels is recorded, covering the most important facial muscles; as in previous studies, channel 5 is not used due to artifacts. EMG data is recorded at 600Hz sampling rate. Audio data is simultaneously recorded with a close-talking microphone and used to create the phone-level alignments which are part of the corpus distribution, otherwise acoustic data is *not* used in this study.

From the total of 61 sessions, we use 48 sessions by the two speakers who recorded a large number of sessions: Precisely, these are sessions 1 – 32 by speaker 2, and sessions 1 – 16 by speaker 8. Each session consists of 50 sentences, 10 of which are used as *test* set, the remaining 40 sentences per session are used for *training* and unsupervised adaptation. The test sentences are identical across sessions.

For experiments on session independency, we subdivide the set of 48 sessions as follows: We create six *blocks* of eight consecutive sessions (four blocks for speaker 2, two blocks for speaker 8). All parameter tuning is performed on blocks 1 and 3 of speaker 2 and block 1 of speaker 8 (the *development* dataset), the remaining data is set aside for the final *evaluation* of our experiments, see section 5.3. Note that the subdivision into training and test data is done *within* each session, whereas the held-out evaluation dataset consists of *entire sessions*. All systems are speaker-dependent. Table 1 gives the statistics of the dataset, which in total contains more than 2 hours of EMG recordings.

Single experiments are run by block, as follows: One session is designated as *target* session for unsupervised adaptation, and one session is set aside as *extra* session for future experiments. The remaining six sessions form the *source* data on which supervised training is performed. Thus, each system receives around 17 minutes of supervised training data. In total, eight experiments with different session subdivision are run on each block, so that each session is taken as target session for exactly one experiment.

4. Methods

4.1. EMG Features

Time-domain EMG features [31, 2] are derived as follows [6]: The input EMG data is windowed with a window length of 27ms and a window shift of 10ms, for each window, a low-frequency (LF) and a high-frequency (HF) part are extracted using a weighted moving average filter, and finally five features (LF power, LF absolute mean, HF power, HF absolute mean, and HF zero-crossing rate) are computed. Thus we compute 25 features from the five input EMG channels. The features are normalized *independently* for each session (which greatly improves the accuracy of the session-independent systems). Finally, a Linear Discriminant Analysis (LDA) transformation is computed on a context window of 11 frames. The LDA target classes are taken from the subphone alignments contained in the corpus distribution, where each phone is subdivided into three substates (begin, middle, and end), yielding a total of 3×45 subphone classes plus a silence class. The LDA cutoff dimension varies, see below.

Table 1: *Statistics of the data corpus*

Set	Number of Speakers	Number of Sessions	Avg session length	Total length
Dev	2	24	2:53	1:09:17
Eval	2	24	2:41	1:04:30

4.2. Baseline System

Our speech recognizer is a *hybrid* system, where an HMM backend performs a time-synchronous beam search over state probabilities generated by a frontend which performs recognition at the frame level, without taking temporal information into account. The two frontends employed in this study are a classical Gaussian Mixture Model (GMM) and a (significantly more performant) Deep Neural Network (DNN) as introduced in prior work [6]. We always train at the frame level (with 11 context frames, see section 3), since high-quality frame-level alignments are available, there is no need to perform realignment during training. The GMM frontend and the search are implemented with the software package BioKIT [32], for the DNN training and evaluation, we use Tensorflow [33].

GMM frontend The GMM frontend is based on *Bundled Phonetic Features* (BDPF) [16], which are an efficient method to create data-adapted context-dependent models for small data sets where classical context-dependent modeling cannot be applied. For BDPF modeling, a number of phonetic decision trees [34] are created, whose roots correspond to (binary) phonetic features, for example the place or manner of articulation. Multiple BDPF trees are used to compensate for the fact that a single tree may not fully discriminate all available phones, we use eight BDPF trees whose roots are the most common phonetic features [14]. Details about the generation of these BDPF trees are described in ref. [16], see ref. [6] for the exact training protocol. Each BDPF tree is trained separately, during HMM-based decoding, the state probability scores of the trees are averaged.

In order to allow a fair comparison between the frontends, we use the same BDPF trees for both the GMM and the DNN frontend [6] (note, though, that they are far more important for the former than for the latter). We also remark that the GMM-based criterion for the tree computation can be replaced by a neural network based criterion [35].

DNN frontend Separate DNNs are trained for each BDPF tree, each of which is set up as follows. We use three hidden layers with 600 neurons, each of which is followed by a tanh nonlinearity and Dropout with 50% dropout probability (see section 5.1 for remarks on the optimal network architecture). The final DNN output layer corresponds to the final states of the respective BDPF tree. The DNN weights are initialized using a Gaussian distribution with a standard deviation of 0.1 and trained with an ADAM optimizer [36] for a maximum of 500 epochs, using a minibatch size of 20 sentences, a multiclass cross-entropy loss function, and early stopping based on the accuracy on the test data of the *source* sessions (since we assume that target labels are not available in practical scenarios).

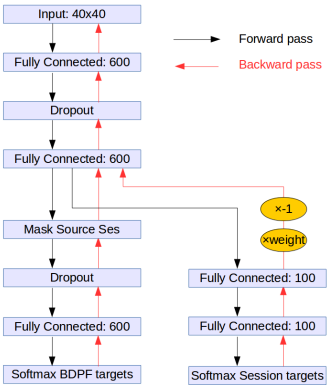

Figure 2: *Optimal network topology for adversarial training, with a common part (top), BDPF state classifier (bottom left, only on source sessions), and session classifier (bottom right). Note that the gradient of the sessions classifier is inverted, and that the contribution of the adversarial part is configurable.*

HMM backend As described above, the HMM backend is not used or tuned during training. For decoding, it is used with a trigram Broadcast News language model (evaluation set perplexity = 24.24) [37]; as in previous experiments, we limit the decoding vocabulary to the 108 words in the test set.

4.3. Domain-Adversarial Training

In order to achieve improved session independency, we follow the *Domain-Adversarial Training* approach by Ganin and Lemptsky [7]. This is a variant of multitask training, where two loss functions are simultaneously optimized: The standard classifier loss (i.e. multi-class cross entropy) is optimized only on data from the source domain, since only for this data training targets are available. Each source training data batch is augmented by an equal amount of randomly chosen data from the target session (note that there is much more source data than target data), and at an intermediate layer—in this study, the second feedforward layer—we attach a further network which performs framewise *session* classification. This is a variation of the original setup, where the secondary classifier only attempts to distinguish the source and target domains; for us it yielded slightly better results. The secondary network follows a standard pattern; it is trained jointly with the main classifier, with a configurable weight.

So far, this describes a joint classifier for two different tasks (session classification and BDPF state classification). The secondary network, however, is made *adversarial* by a simple twist: The backpropagated gradient from the secondary network is *inverted* where it is fed into the main branch. This causes the lower (joint) part of the network to perform gradient *ascent* on the session classification task, thus it learns to confuse sessions instead of recognizing them. Figure 2 shows a graphical overview of the system: The joint part is at the top, at the bottom are word classifier (left) and speaker classifier (right).

5. Experiments and Results

Here we display and describe the results of our experiments. We always report Word Error Rates (WER) on the *target* sessions, for which no supervised training is performed. Results are averaged over all sessions of the development dataset.

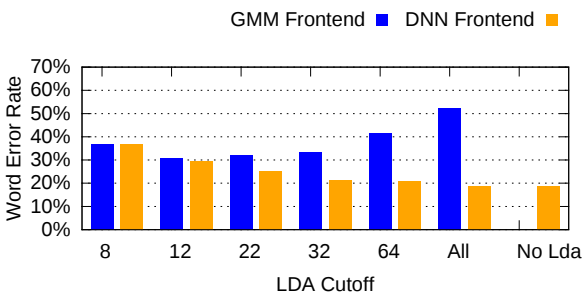

Figure 3: *Comparison of target session Word Error Rates with GMM and DNN frontend, averaged over the development dataset, for different LDA cutoff dimensions. GMM training without LDA dimensionality reduction does not converge.*

5.1. Optimal Baseline System

The first set of experiments deals with establishing our baseline architecture, which might be different from [6] due to the larger amount of training data. We first train a series of experiments on the development dataset (see section 3) to validate the result from [6] that the LDA cutoff for the DNN frontend can be chosen much higher than for the GMM frontend.

The experimental protocol is as follows: We first train GMM systems with varying LDA cutoff (otherwise, there are no further major parameters which need to be tuned). All systems are trained on the *source* sessions and tested on the *target* session. The results can be seen in figure 3, which shows the behavior reported in [6]: Below an LDA cutoff of 12, the WER rises quickly; when the LDA cutoff is raised, one likewise sees a slow degradation of the system. For the corpus used here, we observe optimal results with an LDA cutoff of 12, so we use this as our GMM baseline.

We now train DNN systems, using BDPF trees from the GMM baseline system and the network architecture described in section 4.2. Compared to the architecture used in [6], we have introduced Dropout and consequently increased the number of neurons per layer. From figure 3 it can be seen that an optimal DNN system is obtained with no LDA preprocessing at all, with a WER of 18.8%. The best GMM system achieves 30.6% WER.

We also experimented with network architecture variations. The network setup from [6], which does not use Dropout, performs slightly worse than the architecture described in section 4.2. By increasing or decreasing the number of neurons per layer, or by adding another hidden layer, we likewise obtain slightly increasing WERs; this may not necessarily be significant, but it indicates that our settings are reasonable. A smaller number of layers causes substantial accuracy degradation.

5.2. Adversarial training results

We now apply domain-adversarial training as follows. The adversarial network is set up as described in section 4.3; after initial experiments we chose to use two hidden layers with tanh nonlinearity and 100 neurons each, followed by a standard softmax layer. Dropout did not yield any improvement. The weight of the adversarial part is set to 2.5 after 5 epochs, and to 5.0 after 10 epochs. This follows the suggestion to slowly "activate" adversarial training [7].

Figure 4 displays the results of this experiment, subdivided into the three sessions blocks of the development set: We see that in all cases, the average Word Error Rate on the target

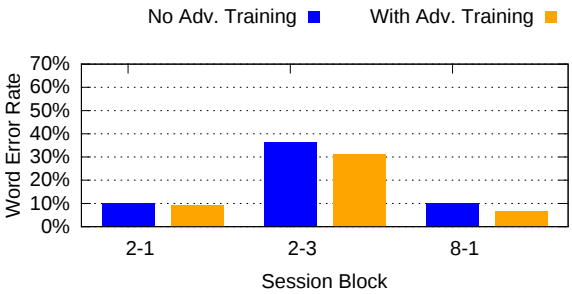

Figure 4: *Target session Word Error Rates with Adversarial Training, averaged over the three session blocks of the development dataset (see section 3 for details)*

.

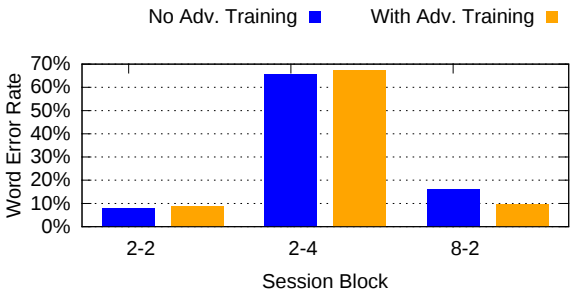

Figure 5: *Target session Word Error Rates with Adversarial Training, averaged over the three session blocks of the evaluation dataset*

session decreases substantially. The average baseline WER (18.8%) is reduced to 15.6% by adversarial training; this is an improvement of 17.0% relative.

The WER reduction is reflected in an improved accuracy of the underlying neural network frontend. Without adversarial training, the average accuracy of the BDPF state recognizer on the target session is 16.4%, averaged over all BDPF trees—a decent result given that we have a recognition task with typically 400 - 500 classes. When adversarial training is applied, the average accuracy rises to 21.9%, which is an improvement of more than 30% relative. The DNN frontend accuracy and the WER are closely related, for the systems without adversarial training, they correlate with a factor of -0.87, with adversarial training, the correlation is lower, at -0.63.

We finally consider the neural network layer to which the adversarial network is attached. At that layer, the objective of the adversarial network is to make the representations of data from the different training sessions more confusable, to improve the target classification accuracy. This can be evaluated with a standard method: We take a session combination (from block 3 of speaker 2) for which adversarial training has a strong effect (WER 36% $\rightarrow$ 25%). For each of the underlying BDPF classifier networks, we take the input data, compute its hidden representation, and perform *session classification* on this data. For this purpose we use a standard SVM with an RBF kernel; the hidden layer data is randomly split into a training set and a test set. We observe a clear trend: Without adversarial training, the SVM accuracy on seven sessions, averaged over the BDPF state classifiers, is 53%, with adversarial training this number drops to 44%. Still, the data representations remain far more different than for the tasks on which domain-adversarial training works best (consider for example figure 3 in [7]).

Table 2: *Target session Word Error Rates on the held-out evaluation dataset.*

	Dataset	
System	**Development**	**Evaluation**
GMM frontend	30.6%	41.4%
DNN frontend	18.8%	29.9%
DNN + adv. training	15.6%	28.5%

5.3. Evaluation

Finally, we verify our result on the held-back evaluation dataset, using the best three systems which we have determined so far. Table 2 summarizes the resulting Word Error Rates.

The characteristics of the evaluation dataset turn out to be different from the development part: In particular, the baseline WER for both the GMM frontend and the DNN frontend is substantially higher than for the development part. Yet, it is known [14] that such variations are well within the normal range.

The effect of domain-adversarial training is lower than for the development dataset, it amounts to 4.7% relative improvement. This result is also reflected in a very small accuracy improvement of the DNN frontend: The average accuracy rises from 17.2% to 17.4%. From the corresponding figure 5, we finally note that there is a strong discrepancy between the two speakers: On the evaluation sessions of speaker 8, adversarial training substantially improves the WER (from 15.9% to 9.7%), which is in line with the expectations from the development dataset. On the sessions of speaker 2, many of which show a generally lower baseline accuracy, we frequently see no improvement at all. This is in line with the results on the development dataset (see figure 4), where the greatest relative improvement of more than 33% is also seen for speaker 8.

6. Conclusion

In this study we have shown that *domain-adversarial training* is applicable to a state-of-the-art session-independent EMG-based speech recognition system with a hybrid DNN + HMM architecture. Testing on a *target* session from which no transcribed training data is used, we observed a relative Word Error Rate reduction of 17.0% on the development dataset and 4.7% on the evaluation dataset. This improvement is reflected in the accuracies of the underlying DNN frontends. However, it is also observed that for different training session combinations, the effect of the method varies drastically. The underlying reasons which make certain systems more or less amenable to domain-adversarial training are not yet known; further investigations are required to shed light on this discrepancy.

7. Acknowledgements

The first author was supported by the H2020 project INPUT – Intuitive Natural Prosthesis UTilization (grant #687795). This work used computational resources from the Swiss National Supercomputing Centre (CSCS) under project ID d74.

8. References

[1] T. Schultz, M. Wand, T. Hueber, D. J. Krusienski, C. Herff, and J. S. Brumberg, "Biosignal-Based Spoken Communication: A Survey," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 12, pp. 2257 – 2271, 2017.

[2] S.-C. Jou, T. Schultz, M. Walliczek, F. Kraft, and A. Waibel, "Towards Continuous Speech Recognition using Surface Electromyography," in *Proc. Interspeech*, 2006, pp. 573 – 576.

[3] L. Diener, C. Herff, M. Janke, and T. Schultz, "An Initial Investigation into the Real-Time Conversion of Facial Surface EMG Signals to Audible Speech," in *Proc. EMBC*, 2016.

[4] M. Wand and T. Schultz, "Session-independent EMG-based Speech Recognition," in *Proc. Biosignals*, 2011, pp. 295 – 300.

[5] ——, "Towards Real-life Application of EMG-based Speech Recognition by using Unsupervised Adaptation," in *Proc. Interspeech*, 2014, pp. 1189 – 1193.

[6] M. Wand and J. Schmidhuber, "Deep Neural Network Frontend for Continuous EMG-Based Speech Recognition," in *Proc. Interspeech*, 2016, pp. 3032 – 3036.

[7] Y. Ganin and V. Lempitsky, "Unsupervised Domain Adaptation by Backpropagation," in *Proc. ICML*, 2015, pp. 1180 – 1189.

[8] J. Freitas, A. Teixeira, M. S. Días, and S. Silva, *An Introduction to Silent Speech Interfaces*. SpringerBriefs in Speech technology, 2017.

[9] B. Denby, T. Schultz, K. Honda, T. Hueber, and J. Gilbert, "Silent Speech Interfaces," *Speech Communication*, vol. 52, no. 4, pp. 270 – 287, 2010.

[10] J. A. Gonzalez, L. A. Cheah, A. M. Gomez, P. D. Green, J. M. Gilbert, S. R. Ell, R. K. Moore, and E. Holdsworth, "Direct Speech Reconstruction From Articulatory Sensor Data by Machine Learning," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 12, pp. 2362 – 2374, 2017.

[11] J. S. Chung and A. Zisserman, "Lip Reading in the Wild," in *Proc. ACCV*, 2016.

[12] T. Hueber and G. Bailly, "Statistical Conversion of Silent Articulation into Audible Speech using Full-covariance HMM," *Computer Speech and Language*, vol. 36, pp. 274 – 293, 2016.

[13] G. S. Meltzner, J. T. Heaton, Y. Deng, G. D. Luca, S. H. Roy, and J. C. Kline, "Silent Speech Recognition as an Alternative Communication Device for Persons With Laryngectomy," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 12, pp. 2386 – 2398, 2017.

[14] M. Wand, "Advancing Electromyographic Continuous Speech Recognition: Signal Preprocessing and Modeling," Dissertation, Karlsruhe Institute of Technology, 2014.

[15] F. Bocquelet, T. Hueber, L. Girin, C. Savariaux, and B. Yvert, "Real-Time Control of an Articulatory-Based Speech Synthesizer for Brain Computer Interfaces," *PLOS Computational Biology*, vol. 12, no. 11, pp. 1–28, 11 2016.

[16] T. Schultz and M. Wand, "Modeling Coarticulation in Large Vocabulary EMG-based Speech Recognition," *Speech Communication*, vol. 52, no. 4, pp. 341 – 353, 2010.

[17] A. Toth, M. Wand, and T. Schultz, "Synthesizing Speech from Electromyography using Voice Transformation Techniques," in *Proc. Interspeech*, 2009, pp. 652 – 655.

[18] K.-S. Lee, "Prediction of Acoustic Feature Parameters using Myoelectric Signals," *IEEE Transactions on Biomedical Engineering*, vol. 57, pp. 1587 – 1595, 2010.

[19] H. Bourlard and N. Morgan, *Connectionist Speech Recognition. A Hybrid Approach*. Kluwer Academic Publishers, 1994.

[20] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep Neural Networks for Acoustic Modeling in Speech Recognition," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82 – 97, 2012.

[21] D. Amodei et al., "Deep Speech 2: End-to-end Speech Recognition in English and Mandarin," in *Proc. ICML*, 2016.

[22] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep Domain Confusion: Maximizing for Domain Invariance," *CoRR*, 2014.

[23] E. Ustinova and V. Lempitsky, "Learning Deep Embeddings with Histogram Loss," in *Proc. NIPS*, 2016.

[24] K. Bousmalis, N. Silberman, G. Trigeorgis, D. Krishnan, and D. Erhan, "Domain Separation Networks," arXiv: 1608.06019, 2016.

[25] K. Ridgeway and M. C. Mozer, "Learning Deep Disentangled Embeddings with the F-Statistic Loss," arXiv:1802.05312, 2018.

[26] J. Schmidhuber, "Learning Factorial Codes by Predictability Minimization," *Neural Computation*, vol. 4, no. 6, pp. 863 – 879, 1992.

[27] M. Wand, M. Janke, and T. Schultz, "The EMG-UKA Corpus for Electromyographic Speech Processing," in *Proc. Interspeech*, 2014, pp. 1593 – 1597.

[28] M. Schünke, E. Schulte, and U. Schumacher, *Prometheus - Lernatlas der Anatomie*. Stuttgart, New York: Thieme Verlag, 2006, vol. 3: Kopf und Neuroanatomie.

[29] M. Wand, M. Janke, and T. Schultz, "Tackling Speaking Mode Varieties in EMG-based Speech Recognition," *IEEE Transaction on Biomedical Engineering*, vol. 61, no. 10, pp. 2515 – 2526, 2014.

[30] L. Maier-Hein, F. Metze, T. Schultz, and A. Waibel, "Session Independent Non-Audible Speech Recognition Using Surface Electromyography," in *Proc. ASRU*, 2005, pp. 331 – 336.

[31] B. Hudgins, P. Parker, and R. Scott, "A New Strategy for Multifunction Myoelectric Control," *IEEE Transactions on Biomedical Engineering*, vol. 40, pp. 82 – 94, 1993.

[32] D. Telaar, M. Wand, D. Gehrig, F. Putze, C. Amma, D. Heger, N. T. Vu, M. Erhardt, T. Schlippe, M. Janke, C. Herff, and T. Schultz, "BioKIT - Real-time Decoder for Biosignal Processing," in *Proc. Interspeech*, 2014.

[33] M. Abadi et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," 2015, software available from tensorflow.org.

[34] L. R. Bahl, P. V. de Souza, P. S. Gopalakrishnan, D. Nahamoo, and M. A. Picheny, "Decision Trees for Phonological Rules in Continuous Speech," in *Proc. ICASSP*, 1991, pp. 185 – 188.

[35] L. Zhu, K. Kilgour, S. Stüker, and A. Waibel, "Gaussian Free Cluster Tree Construction Using Deep Neural Network," in *Proc. Interspeech*, 2015.

[36] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proc. ICLR*, 2015.

[37] H. Yu and A. Waibel, "Streamlining the Front End of a Speech Recognizer," in *Proc. ICSLP*, 2000.