Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

COMPARING SPEAKER INDEPENDENT AND SPEAKER ADAPTED CLASSIFICATION FOR WORD PROMINENCE DETECTION

Andrea Schnall, Martin Ernst Heckmann

2016

Preprint:

This is an accepted article published in 2016 IEEE Workshop on Spoken Language Technology. The final authenticated version is available online at: https://doi.org/10.1109/SLT.2016.7846271

COMPARING SPEAKER INDEPENDENT AND SPEAKER ADAPTED CLASSIFICATION FOR WORD PROMINENCE DETECTION

Andrea Schnall
aschnall@rmr.tu-darmstadt.de
TU Darmstadt - Control Methods and Robotics Lab, TU Darmstadt
Landgraf-Georg Str. 4, 64283 Darmstadt, Germany

Martin Heckmann
martin.heckmann@honda-ri.de
Honda Research Institute Europe GmbH
D-63073 Offenbach am Main, Germany

Abstract

Prosodic cues are an important part of human communication. One of these cues is the word prominence which is used to e.g. highlight important information. Since individual speakers use different ways of expressing prominence, it is not easily extracted and incorporated in a dialog system. As a consequence, up to date prominence only plays a marginal role in human-machine communication. In this paper we compare DNNs and SVMs trained speaker independently with the results of classication with SVM using a speaker adaptation method. Our recently developed adaptation method is based on the radial basis function of the SVM with a Gaussian regularization, which is derived from fMLLR. With this adaptation, we can reduce notably the problem of speaker variations. We present detailed evaluation of the methods and discuss advantages and shortcomings of the proposed approaches for word prominence detection.
Index Terms - word prominence detection, speaker adaptation, SVM, deep neural network

Preprint version. Paper published in: 2016 IEEE Spoken Language Technology Workshop (SLT)

https://doi.org/10.1109/SLT.2016.7846271

©2016 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works

COMPARING SPEAKER INDEPENDENT AND SPEAKER ADAPTED CLASSIFICATION FOR WORD PROMINENCE DETECTION

Andrea Schnall[1]*, Martin Heckmann*[2]

[1] TU Darmstadt - Control Methods and Robotics
, 64295 Darmstadt, Germany
[2] Honda Research Institute Europe GmbH
Carl-Legien-Str. 30, 63073 Offenbach/Main, Germany
Email: aschnall@rmr.tu-darmstadt.de, martin.heckmann@honda-ri.de

ABSTRACT

Prosodic cues are an important part of human communication. One of these cues is the word prominence which is used to e.g. highlight important information. Since individual speakers use different ways of expressing prominence, it is not easily extracted and incorporated in a dialog system. As a consequence, up to date prominence only plays a marginal role in human-machine communication. In this paper we compare DNNs and SVMs trained speaker independently with the results of classification with SVM using a speaker adaptation method. Our recently developed adaptation method is based on the radial basis function of the SVM with a Gaussian regularization, which is derived from fMLLR. With this adaptation, we can reduce notably the problem of speaker variations. We present detailed evaluation of the methods and discuss advantages and shortcomings of the proposed approaches for word prominence detection.

Index Terms— word prominence detection, speaker adaptation, SVM, deep neural network

1. INTRODUCTION

Speech recognition systems and human-machine interfaces have gained much attention in the last years. Still most of them do not include prosodic cues [1, 2]. Since prosody is an important part of human communication, a speech recognition system could be improved by it and be more intuitive to use. One of those important prosodic cues is prominence. It helps us e.g. to mark important information. The difficulty in prominence detection lies in the wide range of possible ways, used by different speakers, to express it. Therefore, good results can be obtained if a classification model is trained and used by the same user, but if the model is trained speaker independent, performance decreases significantly.

The problem of speaker variations in speech processing is normally solved by using speaker adaptation techniques. Commonly used adaptation techniques are maximum a posteriori (MAP) [3], maximum likelihood linear regression (MLLR) [4] and feature space MLLR (FMLLR). All these methods have in common that they are developed for GMM-HMM based systems, which makes sense in speech recognition where they are most often used. In problem cases like the prominence detection, which is a two-class detection problem with highly unbalanced classes, a discriminative classifier like the support vector machines (SVM), leads to better results than a generative classifier. Since the fMLLR can be used to transform the feature space instead of model parameters it can be in general used together with other classifiers. But our experiments [5] showed that the gain from this combination is very low. Therefore, we developed a method which directly adapts the features of a new speaker to the radial basis function (RBF) kernel of an SVM. We could show that with this adaptation method the results of a speaker independent system can be improved.

Compared to our work there has been not much work yet on speaker adaptation for the detection and classification of prosodic cues. In [6] a system is proposed which adapts to a drivers voice for emotion recognition in the automotive environment. The adaptation is accomplished in the acoustic space by mean and standard deviation normalization. In [7] a combination of MAP and Gaussian mixture models (GMM) was used for an unsupervised adaptation to a new unlabeled data set. But since that, to our knowledge, not much research has been done and there is no other work which uses speaker adaptation methods like fMLLR for detection of word prominence and especially not for support vector machines. Compared to the work of [7] we investigate an adaptation that is independent of the classifier. The work of [8] used deep neural networks architecture for classifying lexical stress patterns, which is related to the prominence detection, showing promising results for using more hidden layers than in the traditional Multi Layer Perceptron (MLP). Consequently we present in this paper an evaluation of speaker independent trained models with a deep neural network architecture compared to an SVM classification with and without using speaker adaptation.

In 2 and 3 the used database and features for prominence detection are introduced. In the following section we present the setup of the used classifiers and we shortly review our SVM based algorithms including the extension of the algorithm to enforce sparseness. In 5 we present the experiments and results. We will conclude with a discussion of our findings in section 6 and a subsequent conclusion in our final section.

2. DATA SET

The audio-visual data was recorded as a Wizard-of-Oz experiment during a small game where the subjects moved tiles to uncover a cartoon [9]. The subjects had to give spoken advice, in a simple grammar, to a computer, such as: "place green in B one". Some of the words were misunderstood by the machine; this was verbally and visually shown. Then the subjects had to correct the sentence by repeating it and using prosodic cues to emphasize the misunderstood word as they would do with a human listener. However, correcting cue phrases such as "no I said" were not allowed. This procedure should lead to a rather natural use of prosody, creating a narrow focus condition (in contrast to the broad focus condition of the original utterance) with the corrected words marked as highly prominent.

In total 16 subjects, male and female, speaking either British or American English as native language or being either bilingual in British English/German or American English/German were recorded. Over 2500 utterances with 4 to 5 words each where used for evaluation. For the current experiments we used only the audio recordings. The speech was recorded with a distant microphone at 48 kHz and later downsampled to 16 kHz. A speech recognition system trained on the Grid Corpus [10] was used for forced alignment.

Three human annotators rated the recorded data with a 4 level scale (0-3) of prominence for each word. We calculated the inter-annotator agreement with Fleiss' kappa κ. While doing so we binarized the annotations, only differentiating between prominent and non-prominent. A word was annotated as prominent if the mean rating of all annotators was above 1.5. During the annotation we saw that on the one hand, not all speakers consistently used prominence to highlight the corrected word and, on the other hand, that for some speakers the annotators were not able to come to a consensus on the prominence of the words. Therefore, from the original 16 speakers we retained 8 which showed an agreement measured with Fleiss' κ of more than 0.55 between all annotators ($0.4< \kappa <0.6$ is usually considered as moderate agreement).

3. FEATURES

For the detection of word prominence we employ the prosodic features which have been previously proposed to correlate with word prominence. The beginning and end of the word is taken from the forced alignment and used to calculate the duration of a word as well as the gap length before and after the word.

The loudness is calculated by first filtering the signal with a 12th order IIR filter following the ideas outlined in [11], followed by calculating the instantaneous energy, smoothing with a low pass filter with a cut-off frequency of 10 Hz, and finally converting the results to dB. We expected the loudness to better capture the perceptual correlates of prominence than the energy.

As described in [12], we extract the fundamental frequency f_0 (following [13]), interpolate values in the unvoiced regions via cubic splines and convert the results to semitones. To detect voicing we use an extension of the algorithm described in [14]. Another feature we use is the spectral emphasis i.e. the difference between the overall intensity and the intensity in a dynamically low-pass-filtered signal with a cut-off frequency of 1.5 f_0 [15].

From the fundamental frequency, energy and loudness, we extract functionals as described in [13] for a better prosodic analysis. We extract the mean, max, min, spread (max-min) and variance along the word. Prior to the calculation of the functionals we normalize the prosodic features by their utterance mean and calculated their first and second derivative. To model the contour we calculate the DCT features from the features and retain the first 10 coefficients.

Marking the focus of a word in an utterance rendering it prominent also has an influence on the neighboring words. Modifying the articulatory parameters to raise the prominence of a segment of an utterance (hyperarticulating) is usually accompanied by a reduction of these parameters (hypoarticulation) for the neighboring segments [16, 17]. It has previously been shown that taking this context information into account is very effective for the detection of word prominence and pitch accents [18, 19, 20]. Therefore, we take not only the functionals of the current word itself for classification but also those of the previous and the following word (see [18] for details). Considering all audio features with its functionals and context features we obtain a feature vector with a dimension of 159.

4. CLASSIFICATION METHODS

4.1. SVM

For the experiments with the SVM we used the libsvm implementation [21]. A grid search on each training data set was performed to determine the penalty parameter of the error term, C, and the variance scaling factor γ of the basis function.

In order to use the adaptation we determined mean and variance of the training data, scaled them to zero mean and unit variance and then used the same scaling factors for the test and the adaptation data. We experimented with full covariance matrices [22], but it is a well-known problem that the

estimation of the covariance matrix with a small amount of adaptation data is difficult; especially, with less samples per class than dimensions [23]. So on the one hand, we did a feature reduction with a PCA transformation taking only the first 79 dimensions, because discarding the 80 higher dimensions did not change the performance of the SVM. On the other hand we assumed a diagonal covariance matrix for the training and the adaptation data.
For speaker dependent classification the data is divided in a training set, containing 75% of the data, and a test set, containing the remaining 25% of the data. A 30 fold cross validation in which the data set is always split in a way that the same number of elements is taken from both classes was run. For speaker independent classification the model is trained on 7 speakers and tested on the last (leave-one-speaker-out classification). For testing the adaptation we did a 10 fold cross validation, where we split the data of the new speaker in adaptation and testing data, taking the adaptation data to calculate the transformation matrix and the test data for evaluating the performance of the adaptation. For handling the imbalanced data set (approx. one times to nine), we weight the classes corresponding to their occurrence.

4.2. Adaptation

In [5] we presented a new adaptation method, namely the Gaussian regularized feature-space SVM adaptation, which is primarily designed for SVMs with radial basis function kernels. As basis for adaptation it uses the radial basis function of the SVM and for better generalization, a Gaussian and a sparseness regularization [24]. The method is derive from the fMLLR [4].
The feature-space adaptation method calculates a transformation matrix $\mathbf{W} = [\mathbf{A} \;\; \mathbf{b}]$, where $\mathbf{A}$ is an affine, square transformation matrix and $\mathbf{b}$ is a bias vector, to fit new data to a trained model:

$$\hat{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b} = \mathbf{W}\xi, \tag{1}$$

where $\xi = [1 \;\; \mathbf{x}^T]^T$ is the input vector extended with an extra element equal to unity.
To calculate $\mathbf{W}$, we optimize a function consisting of the main term $\mathcal{Q}_{\text{SVM}}$, which adapts the data to the radial basis function of the SVM and two regularization terms $\mathcal{Q}_{\text{G}}$ and the newly proposed term $\mathcal{Q}_{\text{S}}$, weighted by λ_G and λ_S:

$$\mathcal{Q}_C = \mathcal{Q}_{\text{SVM}} + \lambda_G \mathcal{Q}_{\text{G}} + \lambda_S \mathcal{Q}_{\text{S}}, \tag{2}$$

with

$$\mathcal{Q}_{\text{SVM}} = \sum_{t,n}^{T,N} y_t \alpha_n e^{(-\gamma_{\text{SVM}} \|\mathbf{W}\xi_t - \mathbf{s}_n\|^2)},$$

$$\mathcal{Q}_{\text{G}} = \sum_{t,m}^{T,M} \gamma_m(t)[\log \mathcal{N}(\mathbf{W}\xi_t, \mu_m, \mathbf{\Sigma}_m) + \log(\mathbf{A})]$$

$$\mathcal{Q}_S = \sum_{i,j}^{I,J} \|w_{i,j}\|.$$

The optimization term of the single fSVMA, $\mathcal{Q}_{\text{SVM}}$, makes use of the radial basis function with the parameters of the trained model, the weights α_i, the class affiliation y_i, the scaling factor γ_{SVM} and the support vectors $\mathbf{s}_n$. It calculates a transformation that minimizes the mismatch between the new data and the model,with N being the number of support vectors and T the number of samples. The first extension term (GrfSVMA), the Gaussian regularization, uses the fMLLR to regularize the SVM adaptation in order to get a tradeoff between the fitting of the adaptation data to the SVM boundary and the more general fitting to the assumed Gaussian distribution, with μ and $\mathbf{\Sigma}$ being the mean and variance, i.e. the sufficient statistics when assuming a normal distribution, of the training data and the posterior probability γ of sample $\mathbf{x}_t$ being in Gaussian m. M is the number of mixtures and T the number of samples.
To estimate a model with a lot of free parameters and not much available data is difficult and will often tend to overfit. In classification tasks an often used technique to control the overfitting is regularization. Therefore, usually a penalty term is added to the error function in order to discourage the coefficients from reaching e.g. large values, to keep the model from getting too complex or to restrict it to bounds or smoothness [25]. In our case we want to prevent a bad estimate of the high amount of parameters if not enough adaptation data is available. Since the diagonal is the important part in this transformation matrix the preferable choice is a sparse matrix where especially the small off-diagonal elements are suppressed, therefor we choose the L_1-norm.
For the calculation of the adaptation matrix we use the row-by-row updates for adaptation matrices [26]. We use the supervised version for adaptation, taking a small labeled subset of the test data for adaptation. While doing so, we use the same amount of prominent and non-prominent samples to calculate the transformation matrix.
The function is optimized using gradient descent. We iteratively adapt the rows $\mathbf{w}_i$ of $\mathbf{W}$.

4.3. Deep neural network architecture

Following the work in [8], we tested for our neural network structure 1 to 5 hidden layers with 25 to 400 hidden units per

	GMM	SVM	DNN
speaker dependent	17.1%	11.5%	12.1%
speaker independent	22.0%	16.4%	16.7%

Table 1. Speaker dependent vs. speaker independent classification with SVM and DNN for prominence detection: unweighted error rate average over all speakers.

hidden layer. We implemented the DNN with Tensorflow [27] using a mini-batch gradient descent with an Adam optimization and a batch size of 100. As for the SVM, the features were scaled to zero mean and unit variance and we then used the same scaling factors for the test data.
For this setting we used the full feature dimension and no PCA dimension reduction. We did the same experiments with the reduced dimension, as for SVM, but the performance was slightly worse, so we have chosen the full dimension. To avoid a preference of the dominant class we use an oversampling strategy to balance both classes.

5. EXPERIMENTS

Because the two classes, prominent and non-prominent, are highly imbalanced - approx. one times to nine - the accuracy is not a good measure to compare the results. A preference of non-prominent words would always lead to very good accuracy but would not provide an objective evaluation. Consequently, the unweighted error rate is used instead:

$$\text{unweighted error} \quad = \quad 1 - \left(\frac{tp}{tp+fn} + \frac{tn}{tn+fp} \right) * 0.5,$$

with tp: true positive, fp: false positive and tn: true negative.
We first compare in Tab.1 speaker dependent with speaker independent classification using either GMM, SVM or DNN. For all classifiers, the speaker dependent trained models works better than the speaker independent trained models. The GMM has the highest unweighted error rate, where the speaker dependent classification with 17.1% is already worse than the speaker independent classification with the other two classifiers. The difference between SVM and DNN is small with DNN having a 0.3% higher error rate for speaker independent classification and a 0.6% higher error rate for speaker dependent classification. We see the best performance using SVMs with 16.4% respectivly 11.5% error rate. The performance difference between speaker independently and speaker dependently trained models of up to 4.9% shows that there is potential for using a speaker personalization like a feature adaptation for the classification.
When investigating the performance of the DNN we varied the number of hidden layers and units per layer between 1 - 5 and 25-400, respectively. The corresponding error rates for speaker dependent training can be seen in Fig.1. As a reference the results for SVM classification is also depicted. The

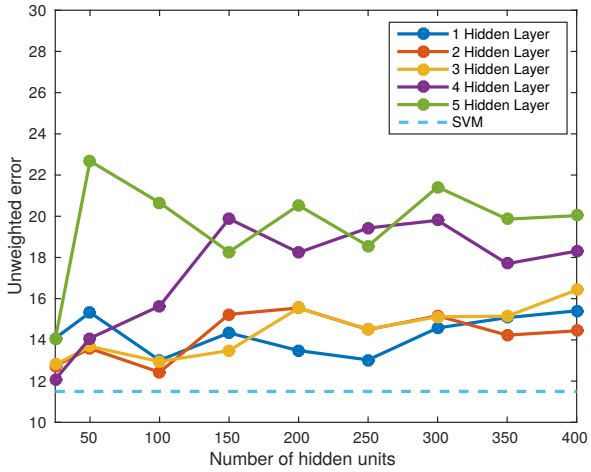


Fig. 1. The average unweighted error over all speakers for speaker dependent classification for prominence detection, using different amount of hidden layers and units.

	DNN	GMM		SVM	
Without Adaptation	16.7%	22.0%		16.4 %	
		10	40	10	40
fMLLR		22.2%	20.4%	18.1%	16.0%
fSVMA				36.4%	27.5%
GrfSVMA				16.9%	14.8%
sparse GrfSVMA				15.1%	14.3%

Table 2. Error for speaker independent classification and speaker adapted classification, adapted data with fMLLR , fSVMA (λ_G:0, λ_S:0), GrfSVMA (λ_G:3, λ_S:0/ λ_G:1, λ_S:0) and sparse GrfSVMA (λ_G:3, λ_S:7/ λ_G:1, λ_S:5). Adaptation was done with either 10 or 40 data points per class.

smallest error rate is reached for 4 Layers and 25 Neurons with 12.1%. Fig. 2 depicts the results of the corresponding experiment for the speaker independent classification task. Here the maximum value is reached for 3 Layers and 25 Neurons with 16.7%.

Next we investigated how an adaptation to the speaker can help to improve performance. We deployed standard fMLLR with GMMs and SMVs as well as our novel approach as described in 4.2 where we adapt to the SVM decision boundary and use Gaussianity and sparseness for regularization. Tab. 2 compares the results of the different methods with a small amount of labeled training data (10 data points per class) and a larger amount of adaption data (40 data points per class). For this experiments we have chosen as weighting parameters $\lambda_G = 3$ and $\lambda_S = 7$ for 10 data points and $\lambda_G = 1$ and $\lambda_S = 5$ for 40 data points as this gave the best results. The detailed parameter optimization can be found in [24].

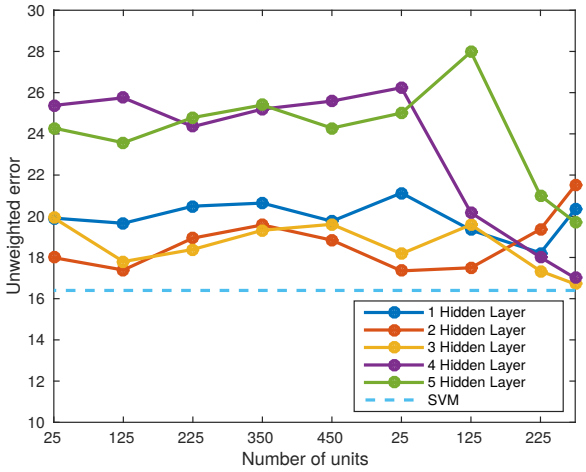


Fig. 2. The average unweighted error over all speakers for speaker independent classification for prominence detection, using different amount of hidden layers and units.

6. DISCUSSION

Generally we can say, that for this problem and feature choice, DNNs perform quite similar to SVMs. Adding more hidden layers showed to be beneficial for the speaker dependent case as well as for the speaker independent case, where the optimum is reached for 3 hidden layers, respectively for 4 hidden layers. As for the number of neurons, a smaller number was better for our problem case. This might be due to the rather small database we use for training. Taking more than 4 hidden layers and a higher amount of neurons led to an increase of the error.

Overall the SVM performed a little better than the DNN. Applying our recently proposed speaker adaptation method, the performance improved notably and the error is much smaller than for the conventional methods. Combining both regularization terms, Gaussiantiy and sparseness, led to the best performance. When only few adaptation data is available (10 data points) the additional sparseness term is most beneficial.

7. CONLUSION

Our experiments showed that with DNNs with more hidden layers and some parameter tuning, a similar performance than with SVM can be reached. Still, for problems where not that much training data is available, like in our case, methods like SVM can be a better choice. Performance might improve for DNN, when using different features and more data.

But best performance can be reached when using additioinal adaptation of a new speaker to the trained model. In future work we will investigate the benefit of adaptation to a DNN model.

8. ACKNOWLEDGMENT

We thank Heiko Wersing, Lydia Fischer and Nikola Aulig for fruitful discussions. Many thanks to Merikan Koyun for the help in the data preparation.

9. REFERENCES

[1] A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. Van, and E. Meteer, "Dialogue act modeling for automatic tagging and recognition of conversational speech," *Computational Linguistics*, vol. 26, pp. 339–373, 2000.

[2] J. Hirschberg, D. Litman, and M. Swerts, "Characterizing and predicting corrections in spoken dialogue systems," *Comput. Linguist*, vol. 32, pp. 417–438, 2006.

[3] J. Gauvain and C. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *IEEE Transactions on Speech and Audio Processing*, vol. 2, pp. 291–298, 1994.

[4] M. Gales, "Maximum likelihood linear transformations for hmm-based speech recognition," *Computer Speech and Language*, vol. 12, 1998.

[5] A. Schnall and M. Heckmann, "Speaker adaptation for word prominence detection with support vector machines," 2016, Speech Prosody, Boston.

[6] B. Schuller, "Speaker, noise, and acoustic space adaptation for emotion recognition in the automotive enviroment," *ITG-Fachtagung Sprachkommunication*, 2008.

[7] S. Ananthakrishnan and S. Narayanan, "Unsupervised adaptation of categorical prosody models for prosody labeling and speech recognition.," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 17, pp. 138–149, 2009.

[8] M. Shahin, B. Ahmed, and K. Ballard, "Classification of lexical stress patterns using deep neural network architecture," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*, 2014.

[9] M. Heckmann, "Audio-visual evaluation and detection of word prominence in a human-machine-interaction scenario," *INTERSPEECH, Portland, OR*, 2012.

[10] M. Cooke, J. Barker, S. Cunningham, and X. Shao, "An audio-visual corpus for speech perception and automatic speech recognition," *The Journal of the Acoustical Society of America*, vol. 120, no. 5, pp. 2421–2424, 2006.

[11] Replaygain, "1.0 specification," .

[12] M. Heckmann, "Steps towards more natural human-machine interaction via audio-visual word prominence detection," in *2nd Workshop on Multimodal Analyses enabling Artificial Agents in Human-Machine Interaction, Singapore*, 2014.

[13] M. Heckmann, F. Joublin, and C. Goerick, "Combining rate and place information for robust pitch extraction," in *INTERSPEECH, Antwerp*, 2007.

[14] T. Kristjansson, S. Deligne, and P. Olsen, "Voicing features for robust speech detection," in *INTERSPEECH*, 2005, vol. 2, p. 3.

[15] Mattias Heldner, "On the reliability of overall intensity and spectral emphasis as acoustic correlates of focal accents in swedish," *Journal of Phonetics*, vol. 31, no. 1, pp. 39 – 62, 2003.

[16] Y. Xu and C. Xu, "Phonetic realization of focus in english declarative intonation," *Journal of Phonetics 33*, pp. 159–197, 2005.

[17] M. Dohen and H. Loevenbruck, "Interaction of audition and vision for the perception of prosodic contrastive focus," *Language and speech*, vol. 52, 2009.

[18] A. Schnall and M. Heckmann, "Integrating sequence information in the audio-visual detection of word prominence in a human-machine interaction scenario," in *INTERSPEECH, Singapour*, 2014.

[19] G. Levow, "Context in multi-lingual tone and pitch accent recognition," *INTERSPEECH*, pp. 1809–1812, 2005.

[20] A. Rosenberg and J. Hirschberg, "Detecting pitch accents at the word, syllable and vowel level," *HLT-NAACL*, 2009.

[21] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27;1–27;27, 2011, Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[22] D. Povey and G. Saon, "Feature and model space speaker adaptation with full covariance gaussians.," in *INTERSPEECH*. 2006, ISCA.

[23] Joseph P. Hoffbeck and David A. Landgrebe, "Covariance matrix estimation and classification with limited training data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, pp. 763–767, 1996.

[24] A. Schnall and M. Heckmann, "Balancing gaussianity and sparseness in feature-space speaker adaptation for word prominence detection," 2016, ITG Conference on Speech Communication, Paderborn.

[25] Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[26] K. Sim and M. Gales, "Adaptation of precision matrix models on large vocabulary continuous speech recognition," *ICASSP*, 2005.

[27] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, Software available from tensorflow.org.