INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Estimating the Sincerity of Apologies in Speech by DNN Rank Learning and Prosodic Analysis

Gábor Gosztolya [1,2], *Tamás Grósz* [1], *György Szaszák* [3], *László Tóth* [2]

[1]Institute of Informatics, University of Szeged, Hungary
[2]MTA-SZTE Research Group on Artificial Intelligence, Szeged, Hungary
[3]Department of Telecommunications and Media Informatics,
Budapest University of Technology and Economics, Budapest, Hungary
{ ggabor, groszt, tothl } @ inf.u-szeged.hu, szaszak@tmit.bme.hu

Abstract

In the Sincerity Sub-Challenge of the Interspeech ComParE 2016 Challenge, the task is to estimate user-annotated sincerity scores for speech samples. We interpret this challenge as a rank-learning regression task, since the evaluation metric (Spearman's correlation) is calculated from the rank of the instances. As a first approach, Deep Neural Networks are used by introducing a novel error criterion which maximizes the correlation metric directly. We obtained the best performance by combining the proposed error function with the conventional MSE error. This approach yielded results that outperform the baseline on the Challenge test set. Furthermore, we introduce a compact prosodic feature set based on a dynamic representation of F0, energy and sound duration. We extract syllable-based prosodic features which are used as the basis of another machine learning step. We show that a small set of prosodic features is capable of yielding a result very close to the baseline one and that by combining the predictions yielded by DNN and the prosodic feature set, further improvement can be reached, significantly outperforming the baseline SVR on the Challenge test set.

Index Terms: computational paralinguistics, Deep Neural Networks, prosodic features

1. Introduction

The Interspeech 2016 Computational Paralinguistics Challenge (ComParE) deals with states of speakers as manifested in their speech signal's acoustic properties. Although most paralinguistic tasks are classification ones, there are regression tasks in this area as well such as estimating the alcohol intoxication level of the speaker [1, 2], the neurological state of Parkinson patients according to the Unified Parkinson's Disease Rating Scale [3, 4, 5] and the intensity of conflict present [6, 7, 8]. ComParE 2016 [9] also includes such a regression task: in the Sincerity Sub-Challenge the sincerity level of apologies has to be predicted. Following the Challenge guidelines, we will omit the detailed description of the task and the dataset; however, an important property of this database is that the speakers uttered one of six pre-defined sentences. Although in some cases the speakers misread the text or read two consecutive apologies, we still make use the fact that most utterances had a fixed structure.

Since the evaluation metric is Spearman's correlation, the task can be handled as a rank learning task, as the goal is to predict the annotated *order* of the speech files. To this end, in our first approach we modify Deep Neural Networks to optimize for rank learning instead of minimizing the standard Mean Squared Error (MSE) function.

Nevertheless, in our opinion sincerity evaluation and emotion recognition are similar tasks in the sense that sincerity is supposed to be rated higher if the semantic content and the emotions reflected by the speech are coherent. In the case of apologies, a coherent emotion means feeling really sorry, not just saying it. Given this coreference of sincerity and emotions, we can exploit results of speech emotion recognition research. Speech prosody is known to be essential in emotion detection tasks and it has also been linked to the perception of sincerity [10, 11] or the listener's impression of possible lying [12]. In his thesis [11], Saowanee mentions high pitch accent and low boundary tone as prosodic markers of sincerity, whereas double pitch accents and high boundary tones are associated with ostensible apologies. In Mexican Spanish Rao et al. found that sarcasm results in a lower speech rate (and hence higher syllable length) and lower mean F0 [10].

The standard approach in emotion recognition tasks is to obtain abundant sentence level statistics on features (such as means, maxima, minima and ranges) and then use some feature selection method (i.e. LDA or PCA) to reduce feature set dimensionality [13]. Slightly modifying this framework, in the current study we propose a dynamic feature representation approach from syllable to syllable (vowel), which is supposed to be closer to human perception. Besides the raw prosodic features like F0, energy or duration we compute derivatives based on small, medium and large contexts to capture both short and long term dynamics and sample these signals at the position of the vowels.

Lastly, we experiment with feature selection for DNN training, and also combine our two approaches outlined above with a simple and robust procedure. By using these techniques we were able to significantly outperform the baseline Support-Vector Regression on the unpublished test set.

2. Learn to rank with DNNs

The Sincerity Sub-Challenge of ComParE 2016 [9] might be viewed as a rank learning task, as the goal is to predict the annotated *order* of the speech files. The evaluation metric of the sub-challenge is Spearman's correlation, which only depends on the ordering produced by our method and the gold standard ranks of the examples. A standard regression training optimizes the Mean Squared Error (MSE), which may not be the optimal solution, as it is known that a good regression model may produce a poor ranking performance [14, 15].

When one uses Deep Neural Networks (DNNs) as the machine learning method, a feasible solution to the problem is to modify the *learn-rate scheduler* [16]. That is, an important part of the DNN training procedure is to use an optimal learning rate schedule. It is a common solution to adjust the learn rate based on the actual DNN prediction performance (see e.g. [17, 18]); if we measure this performance by employing Spearman's correlation as the validation metric, we might get an improvement in the final ranking scores.

In this approach, however, we change only the criterion used for model validation, while the training criterion is unaffected. That is, the parameters of the DNN are still trained by minimizing the MSE criterion. The derivative of this error function is

$$\delta_{MSE}(i) = target_i - output_i, \tag{1}$$

where $target_i$ is the annotated regression target and $output_i$ is the output produced for the ith training example. To achieve the best rank-predicting model, however, we suggest changing the error function of the DNN so that it maximizes the correct correlation metric. Spearman's correlation can be calculated using the formula

$$\rho = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)}, \tag{2}$$

where n is the number of examples and d_i is the difference between the annotated and predicted rank belonging to the ith example. To maximize Eq. (2) for a given set of examples, we need to minimize $\sum_{i=1}^{n} d_i^2$ (as n is independent of i), which may be done in one of two ways. The first option is the standard approach of training our DNN to predict the annotated scores, while the second focuses on learning only the correct ordering instead of the scores themselves.

Evidently, the two tasks are different by nature. By focusing on learning the scores, we can treat each example independently; however, to learn an order we need to examine *pairs* of examples. In this case the learning task can be formalized as a classification of object pairs into two categories: correctly ranked and incorrectly ranked [19]. The drawbacks of this approach are that the number of training pairs could become rather high ($\binom{n}{2}$), and the evaluation of the model is not an easy task. During the evaluation, we get pairwise preferences that we need to aggregate in order to get the final ranks of the examples.

Here, we propose a simple method which uses only the correct ordering provided by the annotators and the actual outputs of the DNN to maximize Eq. (2). Notice that this approach means that we *do not use* the manually annotated training targets, as these are not required for the correct order of examples. Let $rank_{ref}^i$ and $rank_{out}^i$ be the expected and the estimated rank of the ith example, respectively. Then we can calculate the error for DNN training solely from its current output values. The basic idea is that we know the correct rank for each example, and we can also determine which example has that rank in the ordering based on the DNN outputs. That is, for the ith example, we have to find the index j for which $rank_{out}^j = rank_{ref}^i$ holds. If the DNN had omitted $output_j$ for the ith example, then $rank_{out}^i$ would be just equal to $rank_{ref}^i$, meaning that it would have been correctly ranked. Therefore we should just use $output_j$ as the training target for example i. This leads to the simple error function

$$\delta_{Spearman}(i,j) = output_j - output_i. \tag{3}$$

Of course, if we would simply replace the standard MSE error function by the one in Eq. (3) and then train randomly initialized DNNs, we would probably get quite bad models. The reason for this is plain: as a randomly initialized network omits only small random values as outputs, the error signal is quite small as well, hence the training could quickly converge to a suboptimal model. The fact that each DNN model trained would use its own scale for predictions, even when trained on the same examples, means it would also lead to issues which are difficult to handle. For instance, in a cross-validation setup, the predictions of the folds would not be comparable; or it would become quite complicated to train several models on the same task and average out their predictions.

To alleviate this problem, one can initialize the DNNs with the standard MSE regression training and use the $\delta_{Spearman}$ error function afterwards to fine-tune the weights. The main drawback of this method is that it is hard to find the optimal meta-parameters such as the number of training iterations.

Another, quite simple technique for circumventing the above-mentioned issue, which can be expected to be faster as well, is to combine the two training targets and optimize the regression-based and rank-based objectives *simultaneously*. We propose to simply use the weighted sum of the two objectives. That is,

$$\delta_{comb}(i,j) = (1-\lambda)\delta_{MSE}(i) + \lambda\delta_{Spearman}(i,j). \tag{4}$$

Using this error function we can ensure that the magnitude of the outputs remains similar to those of the annotated scores and the actual ranks of the examples are also taken into account. However, the hyper-parameter λ has to be set.

2.1. Results Obtained In the Sincerity Sub-Challenge

We evaluated this training strategy on the Challenge Dataset. For comparison, we also tested the simpler approach of validating with Spearman's correlation in the learn rate scheduler. We always used a Deep Neural Network [20, 21] with three hidden layers, each consisting of 100 neurons that apply the rectifier activation function [22], while in the output layer we used the linear activation function. The input feature vectors were standardized. We trained 10 DNNs for each fold in the speaker-wise CV setup, and averaged out their outputs. The results can be seen in Table 1 below.

	CV		Test	
Training strategy	Pe.	Sp.	Pe.	Sp.
MSE error (Eq. (1))	0.472	0.486	—	—
MSE error + corr. valid.	0.512	0.513	—	—
Combined error (Eq. (4))	0.519	0.520	0.625	0.609
ComParE baseline [9]	0.477	0.474	—	0.602

Table 1: *Pearson's and Spearman's correlation scores got by applying the DNN training strategies we tested.*

Examining the values one can see that even with the standard training strategy, by utilizing the MSE error function we slightly outperformed the baseline. However, by modifying the learn rate scheduler we got better results, which, in our opinion, justifies our efforts of rank learning with DNNs. Lastly, by using the combined error function of Eq. (4) with $\lambda = 0.2$, we got even better correlation scores; this strategy outperformed baseline SVR even on the test set, although by only a slight amount.

3. Utilizing Prosodic Features

Given the results of Saowanee [11] and Rao et al. [10] we decided to derive prosody related features. Our focus of interest covered F0, mean energy and duration and their temporal courses. F0 was extracted using the Kaldi toolkit's `compute-kaldi-pitch-feats` tool [23]; signal energy was obtained with a 250 ms Hamming window. Both feature streams had a 10 ms frame rate initially. The F0 stream was continuous and defined overall (interpolated for unvoiced frames).

In order to capture the dynamic characteristics of both F0 and energy, which we consider relevant in the current task, first and second order deltas were computed as well. This was done based on 3 different intervals (signal contexts) in order to reflect short, mid and long term trends. The delta calculation was based on a regression formula. For frame i and stream $s(n)$, we computed the delta d_i as follows:

$$d_i = \frac{\sum_{j=1}^{W}(s_{i-j} - s_{i+j})}{2\sum_{j=1}^{W} j^2} \quad (5)$$

The higher value was set to W (half window) the longer the context becomes. We used the values of 5, 10 and 25 for W for short, mid and long context, respectively. A short context is expected to capture sudden changes (such as accents), while longer context represents a general trend, i.e. intonation.

For reasons explained in Section 1, each type of sentences (out of the 6) were handled separately and time-aligned with a speech recognizer. Based on this, syllabification (the identification of vowels) was carried out. For each vowel, a 14 dimensional feature vector was composed (raw F0 and energy plus their first and second order deltas with W = 5, 10 and 25), produced by sampling the corresponding feature streams at vowel nuclei (in the middle of the sound). Although this resulted in a variable length vector sequence for the different sentences, sentences with the same content remained pairwise comparable.

Regarding feature normalization and further averaging (mean filtering) our experience tells us that these are really not helpful in the signal processing steps shown so far. For the sake of completeness, however, we present Pearson correlations of raw and normalized (z-scores for energy and log F0) and/or averaged features (7 points mean filter) vs sincerity scores for the train set shown in Table 2. (We omitted the normalized feature vectors, as we standardize the feature sets before employing machine learning, so normalization alone would have no effect.) Our hypothesis was that in the background of this counterproductivity, advantages resulting from the normalization ("level" equalization and variability restriction) are canceled out by the information loss caused by the same normalization (i.e loudness or mean F0 can be an important factor in sincerity evaluation).

Additional meaningful information can be encoded in the speech tempo (e.g. an apology uttered in a monotonic speech style would be judged as not so sincere). As we had only six sentence types, this could be expressed simply as the length of each phoneme; hence we also calculated these features.

3.1. Determining the Sentence Uttered

To utilize the feature vectors described above, first we have to determine the type of sentence for each utterance. For this, we trained an acoustic Deep Rectifier Neural Network [22] with 5 hidden layers and 1000 neurons in each layer. The DNN was trained on the fairly large TEDLIUM speech corpora [24] (following the Kaldi recipe [23]). We used our custom DNN implementation for GPU, which achieved outstanding results on several datasets (e.g. [25, 4]). We used the 39-sized MFCC+Δ+$\Delta\Delta$ feature set [26] with a context-independent tristate phoneme representation.

Next, based on the frame-level DNN outputs, we force-aligned the six sentences for each utterance by dynamic programming. We treated silence as an optional phoneme, and chose the sentence which had the highest overall probability. Besides being able to identify the set of utterances belonging to each sentence type (with an accuracy of over 98% on the training set), with this method we also got the time-alignment, which we were utilized in the prosodic feature extraction step.

3.2. Results

Since each sentence had a different number of phonemes (leading to a different number of features extracted), we had to split both the training and the test sets according to the sentence uttered. This meant that we had training sets of about 100 examples each (as the whole training set consisted of 655 utterances), on which we also had to perform speaker-wise cross-validation. As neural networks are known to behave very unreliably with such tiny training sets, we chose to apply Support-Vector Regression (SVR, [27]) in this sub-task.

The feature sets therefore varied in size from 28 to 392, and from 6 to 92, intonation-based and phoneme length-based feature sets, respectively. Having six sentences and 22 speakers, we trained 132 SVR models overall for each complexity parameter, and we tested the values for C in the range $10^{\{-5,\ldots,1\}}$. In the last step we merged the predictions of all the 132 SVRs into one vector and calculated the two types of correlation scores based on these values.

Feature set	Correlation	
	Pearson	Spearman
Intonation (raw)	0.441	0.436
Intonation (mean)	0.435	0.432
Intonation (normalized + mean)	0.439	0.435
Phoneme lengths	0.447	0.440
Intonation (raw) + phoneme lengths	0.473	0.463
ComParE baseline [9]	0.477	0.474

Table 2: *Pearson's and Spearman's correlation scores got by using the prosodic feature sets in the speaker-wise CV setup.*

Table 2 lists the correlation values we obtained via these strategies. It can be seen that surprisingly high scores could be achieved by utilizing these simple feature sets: by using the two feature sets independently we achieved correlation values that fell quite close to the baseline one, despite the compactness of the feature sets (especially the one consisting of the phoneme lengths). When we combined the two prosodic feature sets, we got a Pearson's correlation score almost identical to that of the baseline, and Spearman's correlation was lower by only 0.01. In our view, these results justify our approach of applying prosodic feature extraction.

4. Further Applied Techniques

4.1. Feature Selection for DNN

The 6373-item standard feature set extracted by the Challenge organizers is naturally full of redundant and irrelevant features. Although current state-of-the-art machine learning methods are able to make reliable predictions in this extremely high-dimensional space, it was shown that they can be assisted by applying feature selection in paralinguistic tasks as well [28, 29, 8]. Therefore we decided to carry out some kind of feature selection beforehand. But as feature selection is not the main focus of this study, we opted for a quite simple method, hoping that it would be sufficiently robust.

Our feature selection approach was based on the assumption that features which correlate well with our target score could be of help for any machine learning algorithm. To this end, we calculated the Pearson's correlation coefficient with the target score for all the standard 6373 attributes, and sorted the attributes according to the absolute value of this coefficient. Then we performed simple nu-SVR regression (using the LibSVM library [30]) utilizing the first n most correlated features, and a step size of 25. We found the optimal number of features to be 400; and by using just these attributes, we improved the Pearson's and Spearman's correlation scores in the speaker-wise CV setup from 0.477 to 0.554 and from 0.474 to 0.555, respectively (see Table 3 below).

	CV		Test	
Error function	Pe.	Sp.	Pe.	Sp.
MSE	0.567	0.563	—	—
MSE + corr. valid.	0.569	0.567	—	—
Combined error	0.582	0.580	0.612	0.601
ComParE baseline [9]	0.477	0.474	—	0.602

Table 3: *Pearson's and Spearman's correlation scores got on the selected feature set.*

Unfortunately, this approach was not justified on the test set: our correlation scores got by feature selection were slightly below the baseline scores there. This may be due to overfitting, since the correlation scores for the feature selection process were calculated on the whole training set instead of just the training folds of the speaker-wise CV.

4.2. Combining the Predictions

Since the standard, 6373-long feature set and the prosodic features led to similarly good results, while the features were fundamentally different, it was reasonable to combine the two approaches in some way. Perhaps the simplest approach is to unite the two kinds of features into one feature vector for each example, although this – due to the different number of prosodic features for each sentence type – again leads to a split of both sets, according to the sentence type. Unfortunately, our tests revealed that this is not really viable for this particular task, as the correlation scores obtained this way were quite low.

Furthermore, uniting the feature sets would mean that we have to apply SVR as the regressor method, due to the tiny size of training sets belonging to each sentence type. Still, for the standard feature set consisting of 6373 attributes, we got quite good results with Deep Neural Networks using the combined error function of Eq. (4). A straightforward solution, however, is to combine the *predictions* of the two approaches.

As the number of submissions was limited, we wanted to avoid using methods with several meta-parameters, as these may turn out to be less robust than expected. Therefore we opted for a simple combination technique. First, to avoid the complications of the two types of predictions having different scales, we converted them so as to have the same standard deviation. (This step affects neither correlation metric.) Next, we calculated the combined predictions using the weighted sum of the predictions of the two approaches. That is, we had

$$Pred_{final} = wPred_{DNN} + (1-w)Pred_{Prosody}. \quad (6)$$

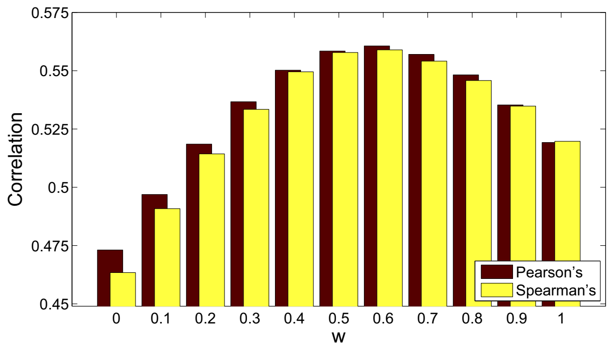

Figure 1: *Pearson's and Spearman's correlation scores obtained by combining the predictions of the DNN using the standard feature set and of the SVR using prosodic features.*

The weight w was set in the speaker-wise CV setup using a grid search; we explored the interval $[0, 1]$ with a step size of 0.1. Figure 1 shows the correlation scores we got in speaker-wise cross-validation as a function of weight w.

		CV		Test	
Feature set	Method	Pe.	Sp.	Pe.	Sp.
Standard	DNN	0.519	0.520	0.625	0.609
Prosodic	SVR	0.473	0.463	—	—
Combination, $w = 0.6$		0.561	0.559	0.636	0.626
ComParE baseline [9]		0.477	0.474	—	0.602

Table 4: *Pearson's and Spearman's correlation scores achieved by using our two approaches in the speaker-wise CV setup and on the test set.*

Table 4 above shows the results we got by applying the two approaches and their combination. It can be seen that linear combination clearly improved the correlation scores in every case; of course, as DNNs were more accurate in the first place, cases with $w \geq 0.5$ produced better results. In the end the weight $w = 0.6$ proved to be optimal, therefore we used this value for our prediction on the test set. On the test set, using the techniques described above, we managed to achieve a Spearman's correlation score of 0.626, which is significantly over the baseline one of 0.602.

5. Conclusions

We approached the Sincerity Sub-Challenge of ComParE 2016 from two different directions. Firstly we exploited the fact that this is a rank learning task; hence we modified our DNN to optimize for directly the order of the training instances. Secondly, we exploited the fact that, apart from a small number of misread recordings, the utterances contained one of the six pre-defined sentences, which allowed us to extract prosodic features such as F0, energy and phoneme duration. Using these simple prosodic properties alone as features in a further machine learning step, we practically matched the baseline SVR scores. Finally, we combined our two approaches by using a simple weighted sum, which gave us a Spearman's correlation score of 0.626 on the test set, thus we significantly outperformed the Challenge baseline of 0.602.

6. References

[1] B. Schuller, S. Steidl, A. Batliner, F. Schiel, and J. Krajewski, "The INTERSPEECH 2011 speaker state challenge," in *Proceedings of Interspeech*, Florence, Italy, Sep 2011.

[2] D. Bone, M. P. Black, M. Li, A. Metallinou, S. Lee, and S. S. Narayanan, "Intoxicated speech detection by fusion of speaker normalized hierarchical features and GMM supervectors," in *Proceedings of Interspeech*, Florence, Italy, Sep 2011, pp. 3217–3220.

[3] B. Schuller, S. Steidl, A. Batliner, S. Hantke, F. Hnig, J. R. Orozco-Arroyave, E. Nth, Y. Zhang, and F. Weninger, "The INTERSPEECH 2015 computational paralinguistics challenge: Nativeness, Parkinson's & eating condition," in *Proceedings of Interspeech*, 2015.

[4] T. Grósz, R. Busa-Fekete, G. Gosztolya, and L. Tóth, "Assessing the degree of nativeness and Parkinson's condition using Gaussian Processes and Deep Rectifier Neural Networks," in *Proceedings of Interspeech*, Dresden, Germany, Sep 2015, pp. 1339–1343.

[5] D. Sztahó, G. Kiss, and K. Vicsi, "Estimating the severity of Parkinson's disease from speech using linear regression and database partitioning," in *Proceedings of Interspeech*, Dresden, Germany, 2015, pp. 498–502.

[6] S. Kim, F. Valente, M. Filippone, and A. Vinciarelli, "Predicting continuous conflict perception with Bayesian Gaussian Processes," *IEEE Transactions on Affective Computing*, vol. 5, no. 2, pp. 187–200, 2014.

[7] B. Schuller, S. Steidl, A. Batliner, A. Vinciarelli, K. Scherer, F. Ringeval, M. Chetouani, F. Weninger, F. Eyben, E. Marchi, H. Salamin, A. Polychroniou, F. Valente, and S. Kim, "The Interspeech 2013 Computational Paralinguistics Challenge: Social signals, Conflict, Emotion, Autism," in *Proceedings of Interspeech*, Lyon, France, 2013.

[8] G. Gosztolya, "Conflict intensity estimation from speech using greedy forward-backward feature selection," in *Proceedings of Interspeech*, Dresden, Germany, Sep 2015, pp. 1339–1343.

[9] B. Schuller, S. Steidl, A. Batliner, J. Hirschberg, J. K. Burgoon, A. Baird, A. Elkins, Y. Zhang, E. Coutinho, and K. Evanini, "The Interspeech 2016 computational paralinguistics challenge: Deception, sincerity & native language," in *Proceedings of Interspeech*, San Francisco, USA, 2016.

[10] R. Rao, "Prosodic consequences of sarcasm versus sincerity in Mexican Spanish," *Concentric: Studies in Linguistics*, vol. 39, no. 2, pp. 33–59, 2013.

[11] S. T. Alexander, "Sincerity, intonation, and apologies: A case study of Thai EFL and ESL learners," Ph.D. dissertation, Indiana University, 2011.

[12] S. Rigoulota, K. Fisha, and M. D. Pella, "Neural correlates of inferring speaker sincerity from white lies: An event-related potential source localization study," *Brain Research*, vol. 1565, no. 27, pp. 48–62, 2014.

[13] C. Busso, S. Lee, and S. Narayanan, "Analysis of emotionally salient aspects of fundamental frequency for emotion detection," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 17, no. 4, pp. 582–596, 2009.

[14] D. Sculley, "Combined regression and ranking," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2010, pp. 979–988.

[15] R. Busa-Fekete, B. Kégl, T. Éltető, and G. Szarvas, "Tune and mix: learning to rank using ensembles of calibrated multi-class classifiers," *Machine Learning*, vol. 93, no. 2–3, pp. 261–292, 2013.

[16] C. Zhang and P. C. Woodland, "A general artificial neural network extension for HTK," in *Proceedings of Interspeech*, Dresden, Germany, Sep 2015, pp. 3581–3585.

[17] D. Yu and L. Deng, *Chapter 4: Automatic Speech Recognition: A Deep Learning Approach.* Boston, MA: Springer, 2015, ch. Deep Neural Networks, pp. 57–78.

[18] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning*, vol. 12, pp. 2121–2159, 2011.

[19] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li, "Learning to rank: From pairwise approach to listwise approach," in *Proceedings of the 24th International Conference on Machine Learning*, 2007, pp. 129–136.

[20] D. Yu and L. Deng, *Automatic Speech Recognition: A Deep Learning Approach.* Springer Publishing Company, 2014.

[21] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[22] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier networks," in *Proceedings of AISTATS*, 2011, pp. 315–323.

[23] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlí ček, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Veselý, "The Kaldi speech recognition toolkit," in *Proceedings of ASRU*, Hilton Waikoloa Village, Big Island, Hawaii, US, Nov 2011, pp. 1339–1343.

[24] A. Rousseau, P. Deléglise, and Y. Esteve, "TED-LIUM: an Automatic Speech Recognition dedicated corpus," in *Proceedings of LREC*, 2012, pp. 125–129.

[25] L. Tóth, "Phone recognition with hierarchical Convolutional Deep Maxout Networks," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2015, no. 25, pp. 1–13, 2015.

[26] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition.* Prentice Hall, 1993.

[27] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[28] O. Räsänen and J. Pohjalainen, "Random subset feature selection in automatic recognition of developmental disorders, affective states, and level of conflict from speech," in *Proceedings of Interspeech*, Lyon, France, Sep 2013, pp. 210–214.

[29] H. Kaya, F. Eyben, A. A. Salah, and B. Schuller, "CCA based feature selection with application to continuous depression recognition from acoustic speech features," in *Proceedings of ICASSP*, Florence, Italy, 2014, pp. 3757–3761.

[30] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 1–27, 2011.