



On Evaluation Metrics for Social Signal Detection

Gábor Gosztolya

MTA-SZTE Research Group on Artificial Intelligence, Szeged, Hungary

ggabor@inf.u-szeged.hu

Abstract

Social signal detection is a task in speech technology which has recently become more popular. In the Interspeech 2013 ComParE Challenge one of the tasks was social signal detection, and since then, new results have been published on the dataset. These studies all used the Area Under Curve (AUC) metric to evaluate the performance; here we argue that this metric is not really suitable for social signals detection. Besides raising some serious theoretical objections, we will also demonstrate this unsuitability experimentally: we will show that applying a very simple smoothing function on the output of the frame-level scores of state-of-the-art classifiers can significantly improve the AUC scores, but perform poorly when employed in a Hidden Markov Model. As the latter is more like real-world applications, we suggest relying on utterance-level evaluation metrics in the future.

Index Terms: Social signals, speech technology, AdaBoost.MH, deep neural networks, exponential smoothing

1. Introduction

In speech technology an emerging area is paralinguistic phenomenon detection, which seeks to detect non-linguistic events (laughter, conflict etc.) in speech. Being a relatively new area, there is still not a wide consensus about which evaluation metrics should be used. In contrast to ASR, where edit distance-based Word Error Rate (WER) (and sometimes Phoneme Error Rate, PER) is the de facto standard, paralinguistic research uses a wide variety of metrics, sometimes even on the same corpus.

One task belonging to this area is the detection of social signals, from which, perhaps laughter and filler events (vocalizations like “eh”, “er” etc.) are the most important. Many experiments were performed with the goal of detecting laughter (e.g. [1, 2, 3, 4]), and the detection of filler events has also become popular (e.g. [5]). To highlight the importance of social signal detection, in 2013 one of the tasks of the INTERSPEECH ComParE Challenge [6] focused on detecting these events.

Perhaps the most interesting finding of the Challenge was that although classification and evaluation were performed at the frame level, it is worth making use of the contextual information and adjusting the frame-level scores based on the local neighborhood. Gupta [7] applied probabilistic time series smoothing; later, Brueckner [8] trained a second neural network on the output of the first, frame-level one to smooth the resulting scores, using a surprisingly wide sliding window; in a later study, Brueckner [9] used a deep bidirectional recurrent neural network, which gathers information about the earlier frames in its Long Short-Term Memory block.

This publication is supported by the European Union and co-funded by the European Social Fund. Project title: Telemedicine-oriented research activities in the fields of mathematics, informatics and medical sciences. Project number: TÁMOP-4.2.2.A-11/1/KONV-2012-0073

All these studies used the evaluation methodology defined in the Challenge, which was based of the Area Under the Curve (AUC) metric, calculated on the frames of the utterances. As in this actual task there were two phenomena to detect (laughter and filler events), the AUC scores for the two events were averaged to give the final ranking of the classifier, resulting in the Unweighted Average Area Under the Curve (UAAUC) score.

Clearly, there are a number of reasons for using this metric. AUC is often used to measure the reliability of a two-class classifier, and its three-class extension in this actual task is quite straightforward. However, in our opinion, there are a number of reasons why this metric does not really suit this particular task.

One of them is that when we apply this metric, we assume that the class labels of the frames are 100% accurate (so we have a perfect manual annotation). In practice, however, it is impossible to objectively position the boundaries of any linguistic event (in our case, laughter and filler events) within a 10ms precision, which is the typical frame-shift: due to the continuous movement of the vocal chords and the mouth, usually there is no clear-cut boundary between two consecutive events.

This also implies that the frames cannot be treated as independent and equally important examples: while two annotators would clearly agree in the inner frames of an occurrence, they would probably disagree on a number of frames near the borders. This implies that we should use a metric where the frames at the center of a phenomenon are more important than those at the sides. Furthermore, the fact that the more successful approaches for this task all used some kind of averaging over time also contradicts the assumption of frame independence, on which UAAUC is based on.

Another reason is that of user expectations. This task is a typical information retrieval (IR) one: we wish to find real occurrences of specific events from a huge number of occurrence candidates. If our aim is to locate the filler events or laughter inside an utterance, we clearly do not mind if the boundaries of the found and the hand-labeled occurrences differ a couple of frames (especially since we cannot expect a perfect match anyway). Frame-level AUC, however, just does not accord with this expectation.

Of course, all these theoretical objections do not mean that UAAUC does not correlate well with some other metric that better fits the above expectations. To this end, in this study we will also show experimentally why we cannot rely on UAAUC in this task. For this purpose, we will use our approach submitted to the Challenge [10], where we utilized the AdaBoost.MH [11] method trained on frame-level. Then we apply an extremely simple smoothing algorithm on the top of the output likelihoods of AdaBoost.MH, and show that this extension improves the results of our classifier by a surprisingly large amount in terms of class-wise averaged AUC. (In fact, we outperform all previous results except heavyweight Recurrent Neural Networks.) However, when we evaluate the performance of the classifier

methods by utilizing a Hidden Markov Model to detect event *occurrences*, and use occurrence-level precision, recall and F-measure scores as evaluation metrics, we find that smoothing actually makes the results attained *worse*. (We also repeat this experiment with the baseline SVM classifier and with Deep Neural Networks, which can also be regarded as a state-of-the-art classifier; the results will be quite similar in both cases.) This, in our opinion, supports the view that the UAAUC metric is unable to identify the real nature of this problem, and the accuracy of social signal detection methods can be more reliably expressed in terms of utterance-level IR metrics.

Note that we do not question the applicability of frame-wise AUC in a classification challenge as in ComParE 2013. For such tasks we need easy-to-calculate and platform-independent metrics, which happens to be quite different from needing to configure a Hidden Markov Model, deal with the special cases of matching overlapping occurrences of different events, etc., just to rank a classification method. However, in other scientific studies, in our opinion, applying a HMM and measuring utterance-level scores should be standard practice, similarly to the case of phoneme classification.

1.1. The SSPNet Vocalization Corpus

The SSPNet Vocalization Corpus [5] consists of 2763 short audio clips extracted from telephone conversations from 120 speakers, containing 2988 laughter and 1158 filler events. We used the feature set supplied for the Challenge, which was extracted with the openSMILE tool [12]. It consisted of the frame-wise 39 MFCC + Δ + $\Delta\Delta$ feature set along with voicing probability, HNR, F0 and zero-crossing rate, and their derivatives. To these 47 features their mean and standard derivative in a 9-frame long neighbourhood were added, resulting in a total of 141 features [6]. Each frame was labeled as one of three classes, namely “laughter”, “filler” or “garbage”.

2. The Classification Methods Used

2.1. AdaBoost.MH

AdaBoost.MH [11] is an efficient meta-learner algorithm, which seeks to build a strong *final classifier* from a linear combination of simple, scalar *base classifiers*. For more complex problems, the state-of-the-art performance of AdaBoost.MH is usually achieved using *decision trees* as base learners, parametrized by their number of leaves.

We employed an open source implementation (the *multi-boost* tool [13]), and followed a multi-armed bandit (MAB) setup [14], which can speed up training significantly. In it, for each boosting iteration step, the optimal base learner is found using only a small subset of features, and the usefulness of these subsets are learned from the accuracy of these basic classifiers [15]. We used our model trained for the Challenge: we sampled the overrepresented “garbage” class, and included the feature vectors of 8 neighbouring frames on each side. We used 8-leaved decision trees as base learners, and trained our model for 100,000 iterations. For details, see [10].

2.2. Deep Rectifier Neural Networks

Deep neural networks differ from conventional ones in that they consist of several hidden layers. This deep structure can provide significant improvements in speech recognition results compared to techniques used previously [16], but the conventional backpropagation algorithm encounters difficulties when

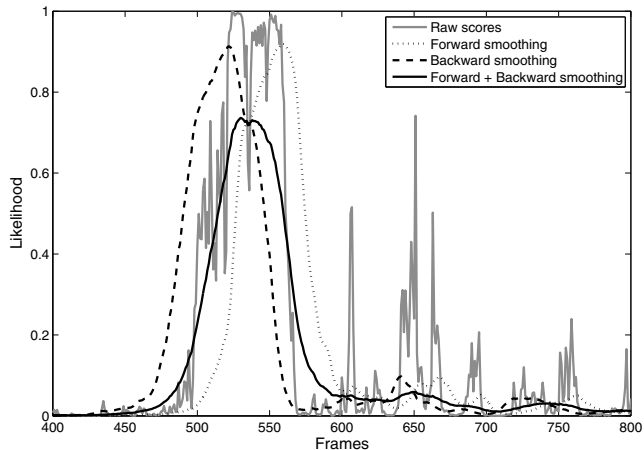


Figure 1: The effect of exponential smoothing on the raw likelihood scores.

training such networks [17]. For this, one possible solution is deep rectifier neural networks (DNNs) [18].

In deep rectifier neural networks, rectified linear units are employed as hidden neurons, which apply the rectifier activation function $\max(0, x)$ instead of the usual sigmoid one [18]. The main advantage of deep rectifier nets is that they can be trained with the standard backpropagation algorithm, without any tedious pre-training (e.g. [19, 20]). We used our custom implementation, originally developed for phoneme classification. On the TIMIT database, frequently used as a reference dataset for phoneme recognition, we achieved the best accuracy known to us with a phonetic error rate of 16.7% [21, 22].

For the actual task, we trained our model on 31 consecutive neighbouring frame vectors. (Due to shorter execution times, we were able to do more experiments with neural networks than with AdaBoost.MH.) After preliminary tests, we used five rectified hidden layers, each consisting of 256 neurons, while we had neurons using the softmax function in the output layer.

2.3. Results Using UAAUC

Table 1 shows the values we got for the two classification methods in terms of the Area Under the Curve metric. Both methods significantly outperformed baseline SVM; AdaBoost.MH performed slightly better than neural networks, which is somewhat surprising as DNN is considered to be the state-of-the-art in frame-level phoneme classification. DNN performed better for filler events, whereas AdaBoost.MH produced a better AUC score for laughter.

3. Posterior Smoothing

We trained our classifiers by treating the frames as individual examples. However, it is easy to see that in practice this is not the case, as usually a transition occurs from one class to another over time. Successful approaches for this task [7, 8, 9] all applied some kind of smoothing along time, which suggests that this is a good way of improving UAAUC scores. Next we will extend our frame-level classifiers by applying the quite basic Simple Exponential Smoothing (SES) [23] method. For the input value series x_1, \dots, x_T it calculates its result s_1, \dots, s_T as $s_1 = x_1$ and

$$s_i = \alpha x_i + (1 - \alpha) s_{i-1} \quad (1)$$

Method	Metric	Dev.	Test
SVM (baseline)	AUC (Laughter)	86.2	82.8
	AUC (Filler)	89.0	83.7
	UAAUC	87.6	83.3
DNN	AUC (Laughter)	92.9	91.3
	AUC (Filler)	95.5	87.9
	UAAUC	94.2	89.6
AdaBoost	AUC (Laughter)	94.0	91.9
	AUC (Filler)	94.9	87.9
	UAAUC	94.5	89.9

Table 1: AUC scores got by using the two classifier methods.

for all $2 \leq i \leq T$, where $0 < \alpha \leq 1$ is a real-valued parameter which controls the level of smoothing. As this method considers only previous observations, we also smoothed the series in descending order; that is, $b_T = x_T$, and

$$b_i = \alpha x_i + (1 - \alpha)b_{i+1} \quad (2)$$

for all $1 \leq i \leq T - 1$, using the same α value as before. The final value was calculated as the mean of the two values, i.e. $x'_i = (s_i + b_i)/2$. (See Fig. 1.) Although exponential smoothing causes a lag in the signal (which can be clearly seen in Fig. 1), we found no improvement in the performance scores when this delay was corrected in time. Here α had been chosen as the value which gave the best UAAUC value on the development set, independently for each method used.

3.1. Results Using UAAUC

Table 2 shows the resulting scores we got by smoothing. It may seem surprising that with such a simple method we could markedly improve the performance: the UAAUC scores are significantly better than their non-smoothed counterparts. The score of 92.8 for AdaBoost.MH was higher than the challenge-winner 91.4 [7] and the 92.4 belonging to the neural network smoothing approach [8], although it lags behind the 94.0 score achieved using heavyweight Recurrent Neural Networks. However, considering the simplicity of our smoothing technique, these scores appear unusually high, and it is hard to see why the smoothed scores are so much better than their raw counterparts. As such a significant improvement in the accuracy scores should be human-interpretable, in our opinion this fact alone could mean that frame-level UAAUC is not the most suitable metric for social signal detection.

Optimal α s (the weight of the actual likelihood) were quite small (in the range 0.02 – 0.05), indicating that for a good UAAUC score, the tendency of the scores is much more important than the raw likelihood for the given frame. Of course, the quality of scores used as the input for smoothing is also very important, as the UAAUC values varied greatly depending on the classification method used.

4. Turning to Event Occurrence Detection

Up to now, we have treated the task as a frame-level classification one. In practice, however, frame-level classification task evaluations are quite rare, frames being an intermediate step required only for technical reasons. In speech recognition the performance is typically evaluated at the sentence level, by applying a Hidden Markov Model (HMM) [24], which turns the frame-level likelihood scores into utterance-level phoneme sequences, which are then rated by the accuracy metrics.

Method	Metric	Dev.	Test
SVM (baseline)	AUC (Laughter)	92.9	88.6
	AUC (Filler)	89.7	84.4
	UAAUC	91.3	86.4
DNN	AUC (Laughter)	97.1	94.9
	AUC (Filler)	96.5	89.4
	UAAUC	96.8	92.1
AdaBoost	AUC (Laughter)	97.9	95.4
	AUC (Filler)	96.7	90.2
	UAAUC	97.3	92.8

Table 2: AUC scores got by using exponential smoothing.

The reason for this is partly that it is impossible to objectively position phoneme boundaries within a 10ms precision, which is the typical frame-shift: due to the continuous movement of the vocal chords and the mouth, usually there is no clear-cut boundary between two consecutive phonemes. Even two human annotators would not agree on the exact phoneme boundaries, but this requirement of precision is avoided if we rate the performance at the utterance-level, where slight differences in time-alignment are tolerated. AUC is slightly better than frame-level classification accuracy: we do not expect clear-cut changes at the “objective” boundaries of the events we are looking for, but prefer a slight transition instead. But treating each frame independently is still a weakness of this approach.

Another reason to change the evaluation metric is the nature of the problem: in the social signal detection task we basically have some occurrences of certain events (now laughter and filler events), and we would like to measure how accurately a system can detect these occurrences. This is clearly the domain of information retrieval, which has straightforward and widely-used metrics: *precision* measures how big the ratio of the hypotheses is for real occurrences, whereas *recall* tells us how many of the real occurrences were found. To balance these two values, they are usually aggregated together by *F-measure* (or *F₁-score*).

It is quite obvious that during this evaluation “occurrence” refers to a time interval and not to one frame or a set of individual frames, which approach clearly better satisfies user expectations. In the task of Spoken Term Detection (STD [25], sometimes also referred to as Keyword Spotting, KWS [26]), where we look for occurrences of given words in speech, it is also standard practice not to expect frame-level precision. Overall, in our opinion there are several reasons for dealing with whole occurrences instead of individual frames; and when we do so, straightforward metrics are precision, recall and *F₁*.

As we have two phenomena to detect (laughter and filler events), we should aggregate the corresponding metric values into one final score. For this, instead of *micro-averaging*, which weights the values obtained for the two events by their number of occurrences, we opted for *macro-averaging*, where the combined precision and recall scores are calculated by taking the unweighted mean of the corresponding scores (then the *F₁*-score is calculated as the harmonic mean of these combined metrics). This means that we consider the two kinds of events equally important, which also matches the set-up of the Challenge [6].

We treated a hypothesis matching an occurrence if they both referred to the same social signal (laughter or filler event), and their time intervals intersected. We adopted this approach as it is standard practice in spoken term detection [27, 28]; also, this way we can eliminate the requirement of frame-precision boundary matching we found so ill-founded previously. Manual

Method	Task	Development set			Test set		
		Prec.	Recall	F_1	Prec.	Recall	F_1
SVM (baseline)	Laughter	50.3%	74.7%	60.1%	42.3%	64.4%	51.6%
	Filler	60.3%	68.0%	65.6%	51.8%	60.3%	55.7%
	Averaged	56.8%	71.3%	63.2%	47.4%	62.3%	53.8%
SVM + smoothing	Laughter	59.5%	69.8%	64.2%	49.9%	57.0%	53.2%
	Filler	64.2%	51.6%	57.2%	51.3%	46.1%	48.6%
	Averaged	61.8%	60.7%	61.3%	50.6%	51.6%	51.1%
DNN	Laughter	68.3%	76.4%	72.1%	58.3%	72.9%	64.8%
	Filler	83.3%	61.7%	70.9%	71.4%	60.8%	65.7%
	Averaged	75.8%	69.1%	72.3%	64.9%	66.9%	65.8%
DNN + smoothing	Laughter	89.3%	48.4%	62.3%	84.3%	35.9%	50.4%
	Filler	96.9%	11.1%	20.0%	85.6%	11.5%	20.3%
	Averaged	93.1%	29.8%	45.2%	84.9%	23.7%	37.1%
AdaBoost	Laughter	74.8%	80.4%	77.5%	58.4%	74.7%	65.5%
	Filler	81.8%	76.8%	79.2%	65.2%	71.1%	68.0%
	Averaged	78.3%	78.6%	78.5%	61.8%	72.9%	66.9%
AdaBoost + smoothing	Laughter	84.7%	73.8%	78.9%	75.0%	58.1%	65.5%
	Filler	84.0%	63.3%	72.2%	69.2%	42.4%	52.6%
	Averaged	84.4%	68.6%	75.6%	72.1%	50.2%	59.2%

Table 3: Precision, recall and F_1 scores obtained on the development and test sets for the different classifier methods.

annotation also tends to be ambiguous at this level of precision, and it is also prone to subjective factors, like whether adjacent or intermediate silent pauses should be part of a filler event.

4.1. Employing a Hidden Markov Model

To calculate these metrics we first have to convert the frame-level likelihoods produced by our classifier method into utterance-level time-aligned event occurrences. We made the straightforward choice of using a HMM; for the sake of simplicity we used uniform state transitional probability values, but employed an insertion penalty to fine-tune the performance. We would like to stress here that this time we wanted to use exactly the same likelihoods as we did in sections 2 and 3.

4.2. Results

Table 3 shows the scores we got for precision, recall and F_1 for both the development and the test sets for each classifier technique tested. As can be seen, both classifier methods (DNN and AdaBoost.MH) attained better scores than the baseline SVM; AdaBoost.MH was a little better than DNN.

The use of simple exponential smoothing dramatically improved UAAUC scores, but it reduced the F_1 values quite significantly in each case. The reason is probably that AUC is an intra-class metric: in it, only the likelihoods of the target class (i.e. filler events or laughter) are considered, and the goal is to produce higher posterior scores for frames belonging to the given class than for the others. This has two important properties in our case: firstly, AUC is scale-independent, so it does not matter if all the likelihoods are quite low as long as they are even lower for frames containing speech or silence; and secondly, these likelihoods are *never compared to the likelihoods of other classes for the same frame*. When we use a HMM, however, this latter property suddenly becomes important, which is completely ignored by AUC. This hypothesis is supported by the fairly high precision and low recall scores: these reflect that the smoothed likelihoods for the laughter and filler classes were generally low, being higher than those of the “garbage” class only at the clean-cut occurrences, while missing the rest.

Note that we do not suggest that systems presented in earlier studies are inadequate for this task (clearly, Recurrent Neural Networks are one of the most powerful tools in speech technology). We just say that their potential cannot really be measured by the UAAUC metric, hence they should be re-evaluated using utterance-level metrics like precision, recall and F-measure.

5. Conclusions

Automatic social signal detection is an emerging task in speech technology. The 2013 Interspeech ComParE Challenge contained a social signal detection task, on which further experiments were later published, all using the original set-up of the Challenge in terms of training-development-test set divisions, feature set and evaluation metric. In this study we argued against applying the Unweighted Average of Area Under the Curve (UAAUC) metric for a number of reasons, primarily because of its frame-based nature.

To demonstrate the unsuitability of UAAUC, we also carried out an experiment. We used two state-of-the-art frame-level classification methods and applied a very simple smoothing technique on their output. This smoothing improved the UAAUC scores by a surprisingly large amount. Then we converted the frame-level posterior scores into utterance-level, time-aligned occurrence hypotheses of laughter and filler events by employing a HMM; these utterance-level occurrences could then be rated via the standard information retrieval metrics of precision, recall and F-measure. We found that although exponential smoothing markedly improved the methods in terms of UAAUC, it actually made the occurrence-level F_1 scores worse. As we consider the latter a more appropriate evaluation methodology, in our opinion this proves that UAAUC is ill-suited to measure the efficiency of social signal detection methods (although it was good for a classification challenge which ComParE 2013 was). We do not mean that methods for this task appearing in previous publications do not perform well; we suggest, however, that they should be re-evaluated by using occurrence-level IR metrics; and researchers should consider using this kind of evaluation in the future.

6. References

- [1] M. T. Knox and N. Mirghafori, "Automatic laughter detection using neural networks." in *Proceedings of Interspeech*, 2007, pp. 2973–2976.
- [2] Y.-X. Li and Q.-H. He, "Detecting laughter in spontaneous speech by constructing laughter bouts," *International Journal of Speech Technology*, vol. 14, no. 3, pp. 211–225, 2011.
- [3] H. Tanaka and N. Campbell, "Classification of social laughter in natural conversational speech," *Computer Speech & Language*, vol. 28, no. 1, pp. 314–325, 2014.
- [4] T. Neuberger, A. Beke, and M. Gósy, "Acoustic analysis and automatic detection of laughter in Hungarian spontaneous speech," in *Proceedings of ISSP*, 2014, pp. 281–284.
- [5] H. Salamin, A. Polychroniou, and A. Vinciarelli, "Automatic detection of laughter and fillers in spontaneous mobile phone conversations," in *Proceedings of SMC*, 2013, pp. 4282–4287.
- [6] 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*, 2013.
- [7] R. Gupta, K. Audhkhasi, S. Lee, and S. S. Narayanan, "Speech paralinguistic event detection using probabilistic time-series smoothing and masking," in *Proceedings of InterSpeech*, 2013, pp. 173–177.
- [8] R. Brueckner and B. Schuller, "Hierarchical neural networks and enhanced class posteriors for social signal classification," in *Proceedings of ASRU*, 2013, pp. 362–367.
- [9] R. Brueckner and B. Schuller, "Social signal classification using deep BLSTM recurrent neural networks," in *Proceedings of ICASSP*, 2014, pp. 4856–4860.
- [10] G. Gosztolya, R. Busa-Fekete, and L. Tóth, "Detecting autism, emotions and social signals using AdaBoost," in *Proceedings of Interspeech*, Lyon, France, Aug 2013, pp. 220–224.
- [11] R. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning*, vol. 37, no. 3, pp. 297–336, 1999.
- [12] F. Eyben, M. Wöllmer, and B. Schuller, "Opensmile: The Munich versatile and fast open-source audio feature extractor," in *Proceedings of ACM Multimedia*, 2010, pp. 1459–1462.
- [13] D. Benbouzid, R. Busa-Fekete, N. Casagrande, F.-D. Collin, and B. Kégl, "MultiBoost: a multi-purpose boosting package," *Journal of Machine Learning Research*, vol. 13, pp. 549–553, 2012.
- [14] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the Multiarmed Bandit Problem," *Machine Learning*, vol. 47, pp. 235–256, 2002.
- [15] R. Busa-Fekete and B. Kégl, "Fast boosting using adversarial bandits," in *Proceedings of ICML*, vol. 27, 2010, pp. 143–150.
- [16] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. 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.
- [17] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of AISTATS*, 2010, pp. 249–256.
- [18] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier networks," in *Proceedings of AISTATS*, 2011, pp. 315–323.
- [19] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proceedings of ASRU*, 2011, pp. 24–29.
- [20] N. Jaitly, P. Nguyen, A. Senior, and V. Vanhoucke, "Application of pretrained deep neural networks to large vocabulary conversational speech recognition," Dept. Comp. Sci., University of Toronto, Tech. Rep., 2012.
- [21] L. Tóth, "Combining time- and frequency-domain convolution in convolutional neural network-based phone recognition," in *Proceedings of ICASSP*, 2014, pp. 190–194.
- [22] L. Tóth, "Convolutional deep maxout networks for phone recognition," in *Proceedings of Interspeech*, 2014, pp. 1078–1082.
- [23] E. Gardner, "Exponential smoothing – the state of the art." *Journal of Forecasting*, vol. 1, no. 4, pp. 1–28, 1985.
- [24] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Prentice Hall, 1993.
- [25] *NIST Spoken Term Detection 2006 Evaluation Plan*, <http://www.nist.gov/speech/tests/std/docs/std06-evalplan-v10.pdf>, 2006.
- [26] M. S. Seigel, P. C. Woodland, and M. J. F. Gales, "A confidence-based approach for improving keyword hypothesis scores," in *Proceedings of ICASSP*, 2013, pp. 8565–8569.
- [27] S. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book*. Cambridge, UK: Cambridge University Engineering Department, 2006.
- [28] G. Gosztolya and L. Tóth, "Spoken term detection based on the most probable phoneme sequence," in *Proceedings of SAMI*, 2011, pp. 101–106.