Using Neural Networks for Data-Driven Backchannel Prediction: A Survey on Input Features and Training Techniques

Markus Mueller, David Leuschner, Lars Briem, Maria Schmidt, Kevin Kilgour, Sebastian Stueker, and Alex Waibel

Interactive Systems Lab
Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Germany,
email: m.mueller@kit.edu

Abstract. In order to make human computer interaction more social, the use of supporting backchannel cues can be beneficial. Such cues can be delivered in different channels like vision, speech or gestures. In this work, we focus on the prediction of acoustic backchannels in terms of speech. Previously, this prediction has been accomplished by using rule-based approaches. But like every rule-based implementation, it is dependent on a fixed set of handwritten rules which have to be changed every time the mechanism is adjusted or different data is used. In this paper we want to overcome these limitations by making use of recent advancements in the field of machine learning. We show that backchannel predictions can be generated by means of a neural network based approach. Such a method has the advantage of depending only on the training data, without the need of handwritten rules.

Keywords: backchannel, neural networks, data-driven prediction

1 Introduction

During a conversation, listeners usually provide feedback to the speaker. They indicate that they are still listening. These cues are often issued using different modalities. Examples are shaking the head or uttering short phrases like "OK.". The intention of this is to make the speaker feel more comfortable while talking as they provide some form (positive or negative) of feedback. Those cues, so-called backchannels (BCs) are usually provided and perceived unconsciously. In contrast to this, a lack of BCs is very well noticed. It leads to the speaker feeling uncomfortable or explicitly asking for some form of feedback. Providing BCs during Human Computer Interaction (HCI) is one method of making the interaction with a Spoken Dialog System (SDS) more natural. The speaker has the feeling of being listened to. This might also help during the interaction via an automated telephone system.

Our approach tackles this problem with neural networks, a machine learning technique inspired by biological neural networks. They are a versatile tool which can be used for different tasks like function approximation, prediction of sequences, encoding or classification. The key feature of neural networks is their ability to learn without the need of handwritten rules. We therefore selected them in order to build a predictor for BCs with few handwritten rules as possible.

This paper is structured as follows: In section 2, we look at other work in this area. Following that, we explain our approach in section 3. We continue with the description of the experiments we conducted (section 4) and an analysis of the results (section 5). We finish with a conclusion in section 6.

2 Related Work

Concerning the prediction of BCs, there have been many publications in the past years. On approach is to use a system that is rule based. Another approach is to use a classifier to predict BCs from a set of input features. With recent advancements in the field of neural networks, we trained a neural network to predict BCs.

2.1 Backchannel Prediction

There exist several approaches towards the prediction of BCs. They utilize different modalities in order to predict the occurrence of a BC, such as visual and auditory information. Examples are the tracking of head movement or prosodic features like pitch and power. All these information sources are based directly on signals originating from the speaker. Besides this information, derived sources like language models or part of speech tagged word sequences are also available. They rely on specially annotated data and provide information in addition to directly observable signals.

After their acquisition the input features need to be processed in order to determine the occurrence of a BC in a word sequence. Many approaches are rule based like the one described in Truong et al. (2010). These rule based approaches often make use of prosodic features to predict BCs and rely on handcrafted rules. Truong et al. (2010) claim that the most important indicators for the placement of a BC are phonetic phenomena occurring right before it. They emphasize pause and pitch, where the latter can either be falling or rising. One of the most important features in their approach is the duration of the pause as well as the duration of the pitch slope at the end of an utterance.

Creating rules for such systems is a time consuming process and includes manual work – which may be error-prone. With the availability of more computing power in recent years, the consequent paradigm shift towards data-driven methods also is reflected in the research of predicting BCs. In Morency et al. (2008), sequential probabilistic models (e.g., Hidden Markov Models, Conditional Random Fields) are trained on human-to-human conversations to predict multimodal listener BCs (e.g., eye gaze and spoken words). Another recent approach towards the generation of BCs is done by Kawahara et al. (2015) by means of a simple prediction model. They predict prosodic features of BCs based on the preceding utterance in order to overcome the BC monotony of most other systems.

In many research areas neural networks have seen a renaissance. Therefore, we used a neural network based approach to predict BCs in this work described in the upcoming sections.

Concerning the evaluation of BC systems, De Kok and Heylen (2012b) give an overview over many published papers. Most of the systems are only evaluated with objective metrics, either with Precision/Recall or with F1. A smaller number of systems is either only evaluated by subjective means (usually a user study) or by both, subjective and objective methods (e.g., De Kok and Heylen (2012a)). We also chose to perform both in our system because we liked to not only know the formal system performance, but also the usability for a potential SDS user. Furthermore, the BC systems named in De Kok and Heylen (2012a) used different margins of error: -500/500ms, -200/200ms, -100/500ms, 0/1000ms. We decided to use the error margin -200/200ms.

2.2 Neural Networks

Neural networks have been used for a variety of tasks like encoding, prediction or classification. In the area of dialogue modelling, Ries (1999) used them in a setup with HMMs to detect different speech acts. Something similar did Stolcke et al. (1998) and Stolcke et al. (2000) where they used a neural network to model dialogue acts. They did not focus on predicting BCs alone, instead they tried to model the different acts of a dialogue. We wanted to use deep belief neural networks (DNNs) Hinton (2006) to predict BCs. The hidden layers are pre-trained using denoising auto-encoders, similar to training of networks for extracting bottleneck features for speech recognition Gehring et al. (2013). After that, we use stochastic gradient descent combined with mini-batches for back-propagation training. We call this *fine-tuning* in this paper.

3 Backchannel Prediction with Neural Networks

We chose to use a neural network as part of our BC predictor because it is not only able to learn by itself how to perform a given classification task, but that it is also capable of generalizing to a great extend. By doing so, we can build a predictor for BCs without the need for writing extensive rules by hand. Since we have not built a BC predictor before, out goal is to build a system that archives a reasonable baseline, ideally matching the baseline from other works in the field.

We start by selecting an appropriate set of features to be fed into the network. The next step is the decide on a neural network design, as well as the training technique. The output of the neural network is then post-processed in order to produce the final set of predicted BCs.

Our experiments were conducted using the Janus Recognition Toolkit (JRTk) Woszczyna et al. (1993). Although this toolkit is mainly developed for speech recognition, it is versatile and can be used in many applications. We used it to extract the features from the audio files and to process the data using neural networks.

3.1 Input Features

Looking at work that has been done in this field, many publications make use of the pitch, the intonation and pause in terms of auditory features. We therefore selected pitch and power for our experiments as well. We used a pitch tracker Kjell (1999) and computed the signal power using methods provided by the JRTk. We did not explicitly use pause information, but we let the neural network extract this information implicitly from the provided energy envelope. To compute features from the input signal, we applied a window of 32ms length and shifted that window with a step size of 10ms over the data. Power and pitch are computed for each window, representing a frame. For each frame, this resulted in two coefficients, one for power and one for pitch. The entire setup of feature extraction and prediction is shown in figure 1

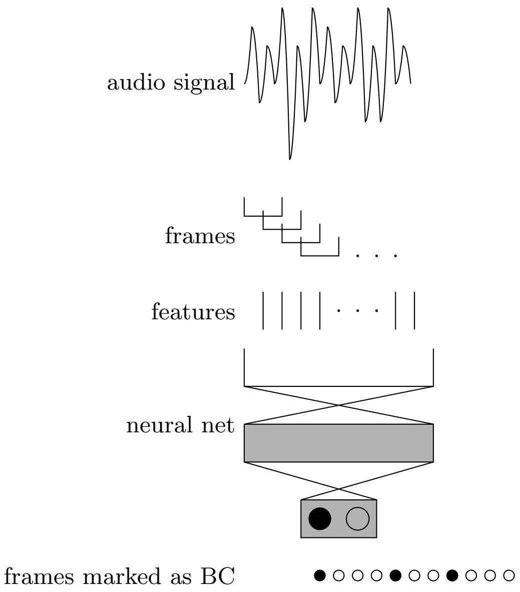

Fig. 1. Setup for extracting BCs

3.2 Neural Network Design

The input features are fed into a neural network for further classification. As the occurrence of a BC is not provoked by a single point in time, we fed a certain context around the current central frame into our network. By doing so, we provide information about rising or falling pitch, as well as variances in the signal power to the network.

The network itself consists of an input layer, one or more hidden layers and an output layer. An example network featuring two hidden layers is shown in figure 2. The input layer has as many neurons to match the dimensionality of the input data. We did not present a single frame to the network, but instead a context of several adjacent frames around a central frame. The output layer consists of only two nodes: One for predicting BCs and one for predicting non BCs.

For the training of the network, we use an approach similar to training a network to extract bottleneck features for a speech recognition system. First, we pre-train each hidden layer in an unsupervised fashion using denoising autoencoders to guide the network weights into an appropriate range. The hidden layers feature a sigmoid as activation function. The network is then fine-tuned via back-propagation using gradient descent. For the error function we use cross-entropy and soft-max as activation function. The training samples are presented to the network in a random order.

After each round of training (epoch), the validation error of the network is computed using a validation data set that the network has not seen during training. This serves as an indication whether the classification performance of the network improves after one iteration of training. We also use this measure as a first indicator of the performance of the final system.

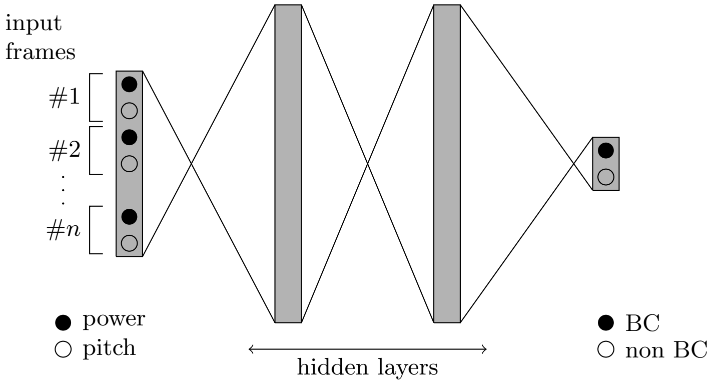

Fig. 2. Example of neural network featuring two hidden layers.

3.3 Post Processing

Our network features two output neurons, for predicting BCs and non-BCs. The output of the network is post-processed in order to obtain the label the current frame. We take the label from that neuron which has the highest output value.

As final step, we apply a filter to suppress any BCs that are predicted within a window of 1s after the last BC was output.

4 Experimental Setup

We first looked into the design and training of the neural network for the classification task. Afterwards, we then applied post processing to the output of the network to determine the final set of BCs. As pointed out in the related work section, the objective evaluation of the results is problematic. We therefore conducted a small user study to assess the subjective performance by humans.

As data we used dialogues from the Switchboard corpus, as the handling of the data is easier compared to data from meetings with a multitude of persons speaking and producing BCs at the same time.

4.1 Switchboard Corpus

The Switchboard Corpus (LDC97S62) consists of English telephone conversations that were collected within the United States. Each conversation features two channels, one for each speaker. The audio is sampled with 8kHz and quantized using the μ-law codec. We used in total 517 hours of speech, originating from 2400 dialogues. As BC, we marked the occurrence of the following utterances: "Um-hum", "Uh-huh", "Yeah right", "Oh", "Um", "Yes", "Huh", "Okay", "Hm", "Hum" and "Uh". Table 1 shows an overview of the used data sets. We assigned single conversations randomly to different data sets. We did not partition the data by individual speakers. Backchanneling is a mutual phenomenon between speaker and listener. Dividing speakers into different groups would therefore have required putting both participants of one conversation into one group. This was not feasible because two speakers had no more than one conversation.

Data set	**Length**	**# Dialogs**	**# BCs**
Total	517 hours	2,438	53,270
Train	424 hours	2,000	43,900
Dev	42 hours	200	4,200
Eval	51 hours	238	5,170

Table 1. Overview of datasets

4.2 Neural Network Design

As input for the network, we tested contexts of different sizes, covering 40,60 and 80 frames of context to the left and right. This results in a feature vector covering 0.81s up to 1.61s. We also varied the amount of nodes per hidden layer evaluating sizes of 64, 128, 256, 512 and 1024. In addition to that, we changed the amount of hidden layers and tested the performance on a network featuring just a single layer as well as up to 10 layers.

4.3 Training Data Selection

Initially, we estimated the appropriate mix of data for the training of the network. When training with data, the ratio between the different classes is important, as the network will implicitly learn an a-priori probability according to the distribution of the data. Hence, finding the right mixture is key.

When extracting the data for training the network, we extracted those parts of the audio, that caused the other speaker of the dialogue to utter a BC. Since other works use temporal features like the movement of the pitch over time, we extracted audio from the area around the occurrence of a BC. Our intention is to capture the data that lead to the utterance of that BC.

After having an initial network design, we also experimented with different history sizes before the appearance of a BC. We extracted data ranging from 1.5s up to 3.5s before a BC in order to train our network upon them.

4.4 User Study

We set up a user study as subjective evaluation means in order to be able to tell how well our BC production system works for users of a potential SDS. We designed an on-line questionnaire and embedded two audio files. We randomly chose two different conversations, extracted a middle piece with a length of ca. 60 seconds, and inserted BCs at the places which were predicted by the NN. The predictions of the first audio file were made by an NN trained on the audio data preceding the BCs by 2s, whereas the second audio file BC were predicted by an NN trained on 3s of audio before the BCs.

Similar to Huang et al. (2010), we asked the participants to rate the amount and the placement of BCs, i.e. whether there were too few or too many and whether they were placed well or whether even possible placement opportunities were missed. Furthermore, we posed the question how naturally or artificially the system sounded to the user. The just named questions have been rated either on a 5-point Likert scale or as yes/no questions. Finally, the upcoming follow-up questions were presented to the participants: Which of the two backchannel audio files did you like the most? (possible answers: 1st, 2nd), Which type of backchanneler are you: Do you produce few, medium or many spoken backchannels? (possible answers: few, medium, many). To account for potential demographic effects, we asked for the gender and age of the participants.

5 Results

We first present the results from various objective evaluations and conclude this section by presenting the results from a small user study that we conducted.

For the objective evaluation, we used two different kinds of measures. First, we used the validation error of the neural network training as an indicator of the performance of a predictor based on a certain neural network. In a second step, we applied the post-processing to the output of the neural network in order to obtain the final BC positions. Based on those occurrences, we computed precision, recall and F-score to assess the performance of our system. We counted a BC as correctly predicted if our system predicted it in a window of 200ms before and after the actual BC. This is one of the measures that has been used in previous publications.

5.1 Neural Network Design

With this set of experiments, we assessed the performance of different architectures of neural networks. We extracted data using a window size of 1s before and after the BC itself for training. Table 2 shows the validation error from different network configurations. The validation error decreases with adding additional layers. Configurations with 128 and 256 nodes show best results. The F-score of the systems using these different networks is shown in table 3. A similar improvement can be observed: Additional layers lead to a higher score. The best result is obtained with a configuration of 128 nodes per layer.

# Layers	1	3	5	7
128 Nodes	0.202	0.196	**0.195**	0.196
256 Nodes	0.201	0.196	**0.195**	**0.195**
512 Nodes	0.202	0.196	0.196	n/a
1024 Nodes	0.203	0.196	0.196	n/a

Table 2. Validation error of different network architectures.

# Layers	1	3	5	7
128 Nodes	0.045	0.053	0.051	**0.060**
256 Nodes	0.044	0.053	0.056	0.058
512 Nodes	0.049	0.054	0.058	n/a
1024 Nodes	0.050	0.043	0.057	n/a

Table 3. F-score of system based on multiple network architectures.

5.2 Context Size

We also considered different context sizes to be fed into the network. Using a context of 40 and 60 frames, we tested different configurations: In one experiment, we fixed the amount of nodes per layer to 256 and evaluated the performance of networks with a different amount of hidden layers. The validation error is shown in table 4, F-score in table 5. Both validation error and F-score again show better results when adding more layers. While the validation error benefits from a larger context, the best F-score result is archived by using a context of 60. In another experiment, we kept the amount of hidden layers fixed to 5 and varied the amount of nodes per layer. Table 6 shows the validation error of the network and table 7 the F-score. Here, the results differ. The best configuration in terms of validation error has a context of 60 and 128 nodes layer. Whereas the best F-score value originates from a system featuring 512 nodes per layer and a context of 40.

# Layers	1	5	7
Context 40	0.201	0.195	0.195
Context 60	0.190	0.174	**0.173**

Table 4. Validation error of two context sizes, tested against several layer configurations.

# Layers	1	5	7
Context 40	0.044	0.056	**0.058**
Context 60	0.022	0.022	0.023

Table 5. F-score of two context sizes, tested against multiple layer configurations.

# Nodes	128	256	512
Context 40	0.195	0.195	0.196
Context 60	**0.173**	0.174	0.175

Table 6. Validation error of various context sizes, amount of nodes per layer is changed.

# Nodes	128	256	512
Context 40	0.051	0.056	**0.058**
Context 60	0.023	0.022	0.023

Table 7. F-score of different context sizes, amount of nodes per layer is varied.

5.3 Training Data Selection

We also evaluated the use of different window sizes for the extraction of training data. We thereby varied the amount of non-BCs speech that is being extracted around the instance of one BC. For these experiments, we chose a network featuring 256 nodes, 5 hidden layers and a context of 40. Table 8 shows the validation error and the F-score of these experiments. The numbers indicate that a larger window size leads to both a better validation score and F-score. Although the validation error constantly decreases, the F-score peaks at a window size of 3s.

Metric	1.5s	2s	2.5s	3s	3.5s	4s
Validation error	0.206	0.195	0.162	0.136	0.117	**0.102**
F-score	0.020	0.056	0.082	**0.093**	0.078	n/a

Table 8. Validation error of extraction lengths.

5.4 Training Data Selection and Context Sizes

Since we saw improvements by increasing the amount of audio that is being extracted around one BC, we investigated the joint effect of increased context sizes with extraction lengths. The results are shown in table 9. Increasing both sizes has a positive effect on the validation error of the network as well as on the F-score of the entire system. By extracting data with a window size of 4s and feeding a context of 60 into the network, we could archive the best F-score with 0.109.

Context	Windowsize	Val. Error	F-score
40	3.5s	0.117	0.078
60	4.0s	0.101	**0.109**
80	5.0s	0.081	0.100

Table 9. Combination of Context and Windowsize.

5.5 Subjective Evaluation

In total, 7 people participated in our user study. Most of them are doctoral students at the same institute, but do not work on the topic of backchanneling themselves. Concerning the demographic characteristics, the participants' age ranges from 24 to 39 with an average of 29.3 years, 5 participants were male, 2 female.

As listed in Table 10, in total the first audio file is rated far better than the second one. Five of seven participants say judge the first audio file to contain an appropriate amount of BCs, while the all seven subjects say the second audio file does not contain an appropriate amount of BCs. The same tendency becomes visible in the second question whether there were too few or too many BCs in the audio: the average rating of 3.25 points on the 5-point Likert scale for the first audio also tells us that the participants are just fine with the amount of BCs inserted. On the contrary, audio file no. 2 gets the worst rating with 5.00 as all participants say the amount of BCs in the file is far too high.

Question #3 asked the participants about the placement of the BCs. They rate the placement in the first audio file with 3.00, so the placement is generally speaking "okay", but there is still space for improvement. At the same time, the second audio file is rated with only 1.29 meaning the placement is done badly. Question #4 mirrors whether many potential BCs were missed. Concerning this question, subjects reject this statement for the first (2.86) and the second audio file (1.29). Of course, the second file has missed fewer potential BCs as there are far too many in the audio right from the start, as the subjects state in the second question. This rating of the first audio file (2.86) is coherent with the rating of question no. 2 (3.25): on average there are slightly too many BCs in the audio, and the participants slightly disagree with the statement that many potential BCs would be missed. Question #5 about the perceived naturalness of the conversation with artificially inserted BCs was rated similarly to the general placement of the BCs: the naturalness of audio file no. 1 has an average rating of 3.00 and audio file no. 2 one of 1.43. This can be interpreted so that audio file no. 1 is quite natural, but there is still much potential for improvement, while audio file no. 2 is rated as far too artificial – as its placement of BCs is also rated badly. Concerning the follow-up questions, Table 10 clearly displays that all seven participants like the first audio file better as the analysis of all previous questions indicated. The results of the question, which type of backchanneler the subjects are themselves, whether they produce few, medium, or many BCs, are shown in table 11: 5 participants say they are using few BCs, while 2 say

Table 10. Results of the user study with a backchannel system with an NN trained on a 2s and 3s span

Questions	1 (3s)	2 (2s)
Amount of BCs appropriate (yes/no)?	5/2	0/7
Too few (1) / many (5) BCs?	3.25	5.00
Generally BCs placed well. (1=disagree,5=agree)	3.00	1.29
Many potential BCs missed. (1=disagree,5=agree)	2.86	1.29
Dialog with BCs sounds artificial (1) / natural (5)	3.00	1.43
Which audio file did you like the most?	7	0

they are using a medium amount of BCs. This question is of course quite vage, but the results have the same tendency as the what we saw in the questions beforehand: our participants rather like fewer BCs.

We can conclude that the first audio file is clearly rated better, which is based on the BC prediction of a 3s-trained NN. This fact is in accord with the results of our objective evaluation: Table 8 shows that the validation error of the 3s NN is smaller as the one of the 2s NN. At the same time, the harmonic mean F1 is higher for the 3s NN than for the 2s NN.

Backchannels	**few (A1)**	**medium (A2)**	**many (A3)**
# of people	5	2	0

Table 11. Which type of backchanneler are you: Do you produce few, medium or many spoken backchannels?.

6 Conclusion

We have presented a novel approach towards the prediction of backchannels using a neural network based system. We have performed experiments to evaluate different neural network architectures and training methods. Our approach is data-driven as it does not require a complex rule set. We only use one rule to prevent a new BC appearing within a 1s window after another.

We examined various NN architectures as well as methods for training them. Our experiments show that using a network with more layers increases the performance of our system. Using a larger window size for data extraction increases the performance as well. Combining a larger window size during data extraction with a larger context size of the network improves the performance even more.

We confirmed the objective choice of system features by means of a final user study. It proves that we created an acceptable baseline system which can be improved by further development. Generally speaking, we plan to increase the perceived naturalness of the system by adding more features. This will be achieved by integrating different BCs opposed to the current approach, in which we only use ”mhm”/”uh-huh”. These could either be randomly inserted or determined by a more intelligent language model in a next step. Another aim of our future work is go beyond dialogs and apply BC prediction to multi-party conversations.

References

Monika Woszczyna, N. Aoki-Waibel, Finn Dag Bu, Noah Coccaro, Keiko Horiguchi, Thomas Kemp, Alon Lavie, Arthur McNair, Thomas Polzin, Ivica Rogina, Carolyn Rose, Tanja Schultz, Bernhard Suhm, M. Tomita, and Alex Waibel: JANUS 93: Towards Spontaneous Speech Translation International Conference on Acoustics, Speech, and Signal Processing 1994

Stolcke, Andreas, et al. Dialog act modeling for conversational speech. AAAI Spring Symposium on Applying Machine Learning to Discourse Processing. 1998.

Kjell, Schubert Pitch tracking and his application on speech recognition Diploma Thesis, University of Karlsruhe (TH)

Ries, Klaus HMM and neural network based speech act detection. Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on. Vol. 1. IEEE, 1999.

Stolcke, Andreas, et al. Dialogue act modeling for automatic tagging and recognition of conversational speech. Computational linguistics 26.3 (2000): 339-373.

Ward, N., Tsukahara, W.: Prosodic Features Which Cue Back-Channel Responses in English and Japanese, Journal of Pragmatics 32, 1177–1207 (2000)

Hinton, Geoffrey, Simon Osindero, and Yee-Whye Teh.: A fast learning algorithm for deep belief nets. Neural computation 18.7 (2006): 1527-1554.

Morency, L.-P., de Kok, I., Gratch, J.: Predicting Listener Backchannels: A Probabilistic Multimodal Approach, Intelligent Virtual Agents, 176–190 (2008)

Huang, L., Morency, L.-P., Gratch, J.: Parasocial Consensus Sampling: Combining Multiple Perspectives to Learn Virtual Human Behavior, Autonomous Agents and Multiagent Systems (AAMAS), 176–190 (2010)

Truong, Khiet P., Poppe, Ronald, Heylen, Dirk A rule-based backchannel prediction model using pitch and pause information Interspeech, 3058–3061 (2010)

de Kok, I., Poppe, R., Heylen, D.: Iterative Perceptual Learning for Social Behavior Synthesis, Centre for Telematics and Information Technology University of Twente, Tech. Rep. (2012)

de Kok, I., Heylen, D.: A Survey on Evaluation Metrics for Backchannel Prediction Models, The Interdisciplinary Workshop on Feedback Behaviors in Dialog, 15–18 (2012)

Gehring, Jonas, et al.: Extracting deep bottleneck features using stacked auto-encoders. Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE, 2013.

Kawahara, Tatsuya, Uesato, Miki, Yoshino, Koichiro, Takanashi, Katsuya Toward Adaptive Generation of Backchannels for Attentive Listening Agents International Workshop Serien on Spoken Dialogue Systems Technology, 1–10 (2015)