Enhancing Backchannel Prediction Using Word Embeddings

Robin Ruede[1], *Markus Müller*[1], *Sebastian Stüker*[1], *Alex Waibel* [1,2]

[1]Karlsruhe Institute of Technology, Karlsruhe, Germany
[2]Carnegie Mellon University, Pittsburgh, PA, USA

robin.ruede@student.kit.edu, m.mueller@kit.edu, sebastian.stueker@kit.edu, waibel@kit.edu

Abstract

Backchannel responses like "uh-huh", "yeah", "right" are used by the listener in a social dialog as a way to provide feedback to the speaker. In the context of human-computer interaction, these responses can be used by an artificial agent to build rapport in conversations with users. In the past, multiple approaches have been proposed to detect backchannel cues and to predict the most natural timing to place those backchannel utterances. Most of these are based on manually optimized fixed rules, which may fail to generalize. Many systems rely on the location and duration of pauses and pitch slopes of specific lengths. In the past, we proposed an approach by training artificial neural networks on acoustic features such as pitch and power and also attempted to add word embeddings via word2vec. In this work, we refined this approach by evaluating different methods to add timed word embeddings via word2vec. Comparing the performance using various feature combinations, we could show that adding linguistic features improves the performance over a prediction system that only uses acoustic features.

Index Terms: speech recognition, human-computer interaction, computational paralinguistics

1. Introduction

While listening to someone, we find ourselves wanting to express sympathy, to acknowledge or reject what we are being told. This form of feedback is called a *backchannel* (BC). The BC provides feedback about the current state of the listener with regards to their interest or opinion about the conversation topic. A BC is generally defined as any kind of feedback a listener provides to a speaker in a segment of conversation that is primarily one-way. BCs include short phrases ("uh-huh", "right", etc.) and physical responses like nodding [1]. BCs are said to help building rapport, which is the feeling of being "in sync" with conversation partners [2]. Artificial assistants like Siri, Alexa, etc. are becoming increasingly popular, but they are still distinctively non-human. Adding BC responses could help make a conversation with AIs feel more natural. In this work, we focus on short phrasal BCs and attempt to predict these based on a given speaker audio channel using only past information. This allows the system to be used in an online environment, predicting BCs with low-latency.

This paper is organized as follows: In the next section, we provide an overview of related work. In Section 3, we describe our proposed approach, followed by the experimental setup in Section 4. The results are presented in Section 5. This paper concludes with Section 6, where we also provide an outlook to future work.

This work has been conducted in the SecondHands project which has received funding from the European Union's Horizon 2020 Research and Innovation programme (call:H2020- ICT-2014-1, RIA) under grant agreement No 643950.

2. Related Work

Most related work is either completely rule-based or uses at least some manual components in combination with data-driven learning. [3] were the first to propose specific rules for when to produce BC feedback based on acoustic cues. This approach was used as reference in many following publications. In general, most related work is based on some variations of pitch and pause, for example [4] extracted multiple different pitch slope features and binary pause regions and then trained sequential probabilistic models like Hidden Markov Models (HMMs) or Conditional Random Fields (CRFs) to extract the relevant subset of those features. [5] trained a decision tree with the C4.5 learning algorithm to distinguish between BC responses, turn-taking, and turn-keeping without a BC. Many approaches have been proposed based on Japanese speech [3, 5, 6, 7, 8, 9], but also on English data [3, 10, 11, 12] and on Dutch speech [4, 13]. [14] compared different evaluation metrics for BC predictors. As an objective evaluation method, the use of the F1-Score has been established. Most research uses a fixed margin of error to accept a prediction as correct. The width of this margin ranges from 400 ms to 1000 ms, and the center of the margin of error ranges from $\pm$0 ms to +500 ms. With no existing standard data set, comparison between different evaluations is difficult. A first approach for distinguishing different speech acts, including BCs, using a combination of HMMs and neural networks was published by [15]. We recently proposed different neural network training methods, feature combinations, context widths and network layouts, while focusing on acoustic features [16]. In this work, we refined this approach by using word embeddings as additional data, and evaluate the results both in an objective and subjective manner.

3. Backchannel Prediction

The definition of BCs varies in literature. There are multiple categories of phrasal BCs, they can be non-committal ("uh huh", "yeah"), positive / confirming ("oh how neat", "great"), negative / surprised ("you're kidding", "oh my god"), questioning ("oh are you", "is that right"), et cetera. To simplify the problem, we only try to predict the trigger times for any type of BC, ignoring the distinction between different kinds of responses.

3.1. Feature Selection

The most commonly used acoustic features in related research are fast and slow voice pitch slopes and pauses of varying lengths. A neural network is able to learn advantageous feature representations on its own. By inputting the absolute pitch and power (signal energy) values for a given time context, the network is able to automatically extract the pitch slopes and pause triggers by subtracting the neighboring values in the time context for each feature. We also added the fundamental frequency variation

spectrum (FFV) [17], which is a seven-dimensional representation of changes in the fundamental frequency over time, giving a more accurate view of the pitch progression than the single-dimensional pitch value.

3.2. Training Area Selection

We generally assume to have two separate but synchronized audio tracks, one for the speaker and one for the listener, each with the corresponding transcriptions. As we aimed at predicting BCs without future information, we need to train the network to detect segments of audio from the speaker track that probably cause a BC in the listener track. We chose the beginning of the BC utterance in the transcript of the listener channel as an anchor t, and then used a fixed context range of width w before that as the audio range $[t-w,t]$ to train the network to predict a BC. The width can range from a few hundred milliseconds to multiple seconds. This approach may not be optimal, because the delay between the last utterance of the speaker and the BC can vary significantly in the training data, causing the need for the network to first learn to align its inputs. In addition to selecting the *positive prediction area* as defined above, we also need to choose areas to predict zero i.e. "no BC". We choose the range a few seconds before each BC as a negative sample. This gives us an evenly balanced data set, and the negative samples are intuitively meaningful, because in that area the listener explicitly decided not to give a BC response yet, so it is sensible to assume whatever the speaker is saying during this period is not a trigger for BCs.

3.3. Neural Network Design and Training

The input layer consists of all the chosen features over a fixed time context. We train the network on the outputs $[1,0]$ for BCs and $[0,1]$ for non-BCs. A visualization of this architecture can be seen in Figure 1.

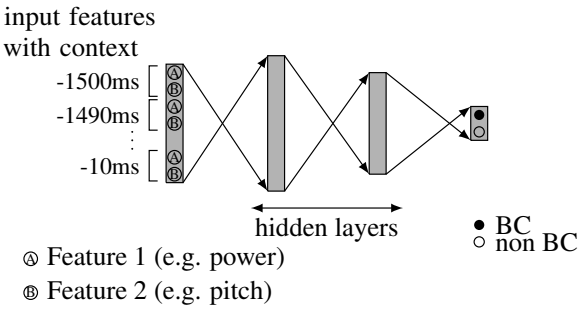

Figure 1: *Feed-forward neural network architecture for backchannel prediction.*

The placement of BCs is dependent on previous BCs: If the previous BC utterance was a while ago, the probability of a BC happening shortly is higher and vice versa. After each BC, the probability of a new BC rises over time. To accommodate for this, we want the neural network to also take its previous internal state or outputs into account. We do this by using Long-short term memory layers (LSTM) instead of dense feed forward layers.

3.4. Postprocessing

Our goal is to generate an artificial audio track containing utterances such as "uh-huh" or "yeah" at appropriate times. The neural network outputs a noisy value between 0 and 1. To generate an audio track from this output, we need to convert this noisy floating probability value into discrete trigger time stamps: We first run a low-pass filter over the network output, which removes all the higher frequency components and yields to a less noisy and more continuous output function. To ensure our predictor does not use any future information, this low-pass filter must be causal. The commonly used gaussian filter is symmetric, which in our case means it uses future information as well as past information. To prevent this, we cut the right side of the filter off asymmetrically at some multiple c of the standard deviation σ. Then we shift the filter to the left so the last frame it uses is ± 0 ms from the prediction target time. This means the latency of our prediction increases by $c \cdot \sigma\, ms$. If we choose $c = 0$, we cut off the complete right half of the bell curve, meaning we do not need to shift the filter, which keeps the latency at 0 ms, but at the cost of accuracy of the low-pass filter.

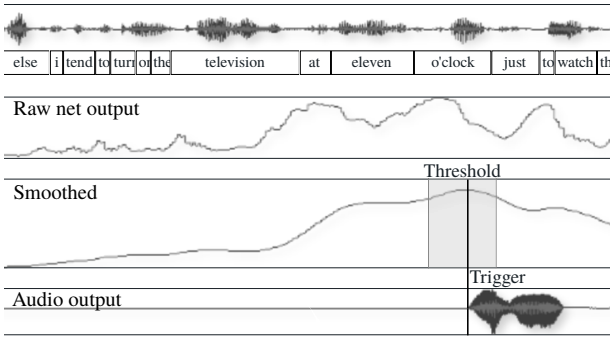

Figure 2: *Example of the postprocessing process. The top shows the input audio channel together with the corresponding transcriptions. The raw output of the neural network is fairly noisy, so we smooth it. The highlighted range on the smoothed output is where the value is larger than 0.7. We trigger at the first local maximum of this range, emitting an "uh-huh" sound sample.*

After this filter we use a fixed trigger threshold to extract the ranges in the output where the predictor is fairly confident that a BC should occur. We trigger exactly once for each of these ranges. Within each range, multiple possibilities to choose the exact trigger time point exist. One possibility would be to use the first local maximum within the thresholded range or the left edge of the range. Using the left edge has the lowest latency, but can give us worse results because it might force the trigger to happen earlier than the time the network would give the highest probability rating. An example of our postprocessing procedure can be seen in Figure 2. We determined the optimal postprocessing hyperparameters for each network configuration and allowed margin of error automatically using Bayesian optimization with the validation F1-Score as the utility function.

3.5. Evaluation

For a simple subjective assessment of the results, we choose some random audio segments where only one person is talking in a monologue. For each segment, we removed the original listener channel and replaced it with the artificial one. This audio data was generated by inserting a random BC audio sample at every predicted time stamp. We selected these audio samples from randoms speakers out of the training data, keeping the speaker the same over the whole segment so it sounds like a specific person is listening to the speaker.

3.5.1. Objective Evaluation

To get an objective evaluation of the performance of our predictions, we take a set of monologuing segments from the evaluation data set and compare the prediction with the ground truth, i.e., all the time stamps where a BC happens in the original data. We interpret a prediction as correct if it is within a specific margin of the nearest real BC in the dataset. In other research, varying margins of error have been used. We used a margin of 0 ms to +1000 ms for the objective evaluation. We calculated the precision, recall, and F1-Score to objectively measure the performance of our prediction systems.

3.5.2. Subjective Evaluation

Because the placement of BCs is subjective as, e.g., not every BC opportunity is taken by the listener, we did a subjective evaluation to complement the objective data. We first extracted all monologuing segments of more than fifteen seconds from the evaluation data set where the ground truth contained at least three BCs. Then we randomly choose six of these, excluding those that contained noise leaking from from one channel to the other. The selected segments were of an average length of 30 seconds. For each segment, we generated three versions of audio:

1. Neural net predictor: Predictions from our best trained LSTM according to the objective evaluation[1].
2. Ground truth predictor: Predictions as read from the real BC audio track.
3. Random predictor (baseline): For each segment, we read the real BC count, and then uniformly distributed that amount of triggers throughout the segment. Note that this method actually has additional information compared to the neural net predictor because it knows the expected BC count.

For each of these, we generated the BC audio track by putting a BC audio sample at each trigger time. We then down-mixed the speaker and artificial listener channel to a mono mp3 file to maximize accessibility. We chose the BC audio samples randomly from all neutral BCs with a minimum loudness in a fixed set of 11 conversations that had a lot of neutral BCs without leaking audio. This gave us a total of $6 \cdot 3 = 18$ audio files. For every participant we randomly selected six audio files so that everyone heard every speaker track exactly once and two samples of every method (Truth, Random, NN). The order was shuffled to remove any structural effects. We asked the participants to rate the audio samples based on how natural the recording sounded in general and how appropriate they thought the BC timing was.

4. Experimental Setup

4.1. Dataset

We used the switchboard dataset [18], which consists of 2438 english telephone conversations of five to ten minutes, 260 hours in total. They are annotated with transcriptions and word alignments [19] with a total of 3 million words. We split the dataset randomly into 2000 conversations for training, 200 for validation and 238 for evaluation. As opposed to many other datasets, the transcriptions also contain BC utterances like *uh-huh* and *yeah*, making it ideal for this task.

[1] Best LSTM with the postprocessing hyperparameters optimized for a margin of error of −200 ms to 200 ms, because a short subjective assessment of the results showed a margin of error of 0 ms to 1000 ms lead to BCs that sounded too delayed.

4.2. Extraction

4.2.1. Backchannel Utterance Selection

We used annotations from The Switchboard Dialog Act Corpus (SwDA) [20] to decide which utterances to classify as BCs. The SwDA contains categorical annotations for utterances for about half of the data of the Switchboard corpus. We chose to use the top 150 unique utterances marked as BCs from the SwDA. The most common BCs in the data set are "yeah", "um-hum", "uh-huh" and "right", adding up to 70% of all extracted BC phrases. To select which utterances should be categorized as BCs and used for training, we first filter noise and other markers from the transcriptions and then compare the resulting text to our list of BC phrases. Additionally we only choose utterances that have either silence or another BC before them, to exclude utterances like "uh" that are a actually speech disfluencies. This selected method results in us a total of 62k BCs out of 390k utterances (16%) or 71k out of 3.2 million words (2.2%). Note that the percentage of words is much lower because BC utterances are shorter than other utterances on average.

4.2.2. Feature Extraction

We used the Janus Recognition Toolkit (JRTk) [21] for the acoustic feature extraction. All features are normalized to fit in $[-1, 1]$. They are extracted for 32 ms frame windows, with a frame shift of 10 ms. This gives us 100 frames per feature per second. In addition, we also trained Word2Vec, an "Efficient Estimation of Word Representations in Vector Space" [22], on the Switchboard dataset. Because our dataset is fairly small, we used relatively small word vectors (10 – 50 dimensions). Word2Vec automatically chooses a fitting vocabulary size. For words not in the vocabulary, we output 0 on all dimensions. We trained Word2Vec only on utterances that are not BCs or silence to reduce the required vocabulary and prevent it from learning about meta information such as annotated pauses, noises and laughter. To increase the amount of data available, we also added the transcriptions from the ICSI meeting corpus [23] to the training data for Word2Vec. This increased the word count from 3 million to 4 million.

We extracted these features parallel to those output by JRTk, with a 10 millisecond frame shift. To ensure we don't use any future information, we extract the word vector for the last word that ended *before* the current frame timestamp. This way the predictor is in theory still online, though this assumes the availability of a speech recognizer with instant output. An example of the Word2Vec feature extracted with this method can be seen in Figure 3.

4.3. Training

We used Theano [24] with Lasagne v1.0-dev [25] for rapid prototyping and testing of different parameters. We trained neural networks with various context lengths (500 ms to 2000 ms), context strides (1 to 4 frames), network depths (one to four hidden layers), layer sizes (15 to 125 neurons), activation functions (tanh and relu), optimization methods (SGD, Adadelta and Adam), weight initialization methods (constant zero and Glorot [26]), and layer types (feed forward, LSTM, dropout).

4.4. Evaluation

The switchboard dataset contains alternating conversations. Because our predictor is only capable of handling situations where one speaker is consistently speaking and the other consistently

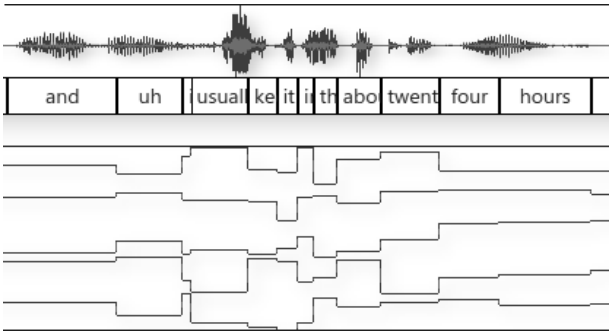

Figure 3: *Five-dimensional Word2Vec feature for some speech. The encoding is offset by one word, for example the encoding for "twenty" is seen below the word "four", because we encode the word that ended* before *the current time. Note that with this method we indirectly encode the length of the words and the time since the last word change.*

listening (silent or producing BCs), we need to evaluate it on only those segments. We call these segments "monologuing segments". Only segments with a minimum length of five seconds were considered to exclude sections of alternating conversation.

5. Results

5.1. Subjective Results

A total of 20 participants participated in the survey, mostly from the university group *ML-KA – Machine Learning University Group at Karlsruhe Institute of Technology*. Because every participant rated two samples for every evaluation method, this gives us a sample size of $N = 40$. The detailed results of the survey can be seen in Table 1. The results show that our neural network performs significantly better regarding timing than a random predictor ($p = 0.005\%$), but significantly worse than human performance ($p = 0.4\%$). For *naturalness*, our predictor is not significantly better than random performance ($p = 7.7\%$). The significance was tested using Welch's t-test.

Participants could leave comments on the survey. One person was confused because they could still hear the leaking original listener audio track beneath the artificial one, even though we explicitly chose segments where this problem should be minimal. Another person commented "Telephone conversations are not in the best sound quality. For me it's hard to follow and it's not easy to rate these conversations."

5.2. Objective Results

All of the results in Table 2 use the following setup: LSTM; Configuration: input → 70 neurons → 35 neurons → output; Context width: 1500 ms; Context frame stride: 2; Margin of error: 0 ms to +1000 ms. We initialized the weights using Glorot uniform initialization and used Adam for optimization. These parameters were chosen to give the best results on the validation data set. A more detailed comparison of training methods, feature combinations, context widths and network layouts can be seen in our previous work [16]. The experimental setup is slightly different there, so the absolute F1-values should not be directly compared. Precision, recall and F1-Score are given for the completely independent evaluation data set. The performance with both linguistic and acoustic features is significantly better than with just acoustic features ($p < 1\%$). The improvement of adding the ICSI meeting corpus to Word2Vec training is not statistically significant.

Table 1: *Survey results. Shown are the average ratings the participants gave regarding timing and naturalness for each prediction method and sound sample.*

Predictor	Sample	Timing	Naturalness	Sample Size
random	**average**	**2.33 points**	**2.63 points**	**40**
	1	1.63 points	2.00 points	8
	2	2.75 points	3.13 points	8
	3	2.43 points	2.14 points	7
	4	2.00 points	2.33 points	3
	5	2.00 points	2.33 points	6
	6	2.88 points	3.50 points	8
nn	**average**	**3.48 points**	**3.08 points**	**40**
	1	3.33 points	3.33 points	3
	2	2.60 points	2.20 points	5
	3	2.38 points	2.25 points	8
	4	3.89 points	3.33 points	9
	5	4.25 points	3.88 points	8
	6	4.00 points	3.29 points	7
truth	**average**	**4.20 points**	**4.08 points**	**40**
	1	4.78 points	4.33 points	9
	2	4.57 points	4.43 points	7
	3	3.80 points	3.80 points	5
	4	3.75 points	3.50 points	8
	5	3.83 points	3.67 points	6
	6	4.20 points	4.80 points	5

Table 2: *Objective results with various input feature combinations.*

Features	Precision	Recall	F1-Score
power only	0.244	0.516	0.331
acoustic (power, pitch, ffv)	0.279	0.515	0.362
linguistic (word2vec$_{dim=30}$)	0.244	0.478	0.323
acoustic and linguistic (power, pitch, ffv, word2vec$_{dim=30}$):			
· word2vec trained on Switchboard only	0.298	0.510	0.376
· word2vec trained on Switchboard and ICSI Meeting	0.305	0.519	**0.385**

6. Conclusion

We have shown that BC prediction can be improved by the use of linguistic features by evaluating the use of different corpora for training the word embeddings. Since BCs are highly subjective in nature, we evaluated our systems using both the F1-Score, as well as a small user study.

Further improvements could be made by using a separate, longer context for the linguistic features than the acoustic features. In our survey, we did not compare the subjective performance when optimizing the hyperparameters for different margins of error to evaluate which margin is subjectively more acceptable.

7. References

[1] T. Watanabe and N. Yuuki, "A Voice Reaction System with a Visualized Response Equivalent to Nodding," in *Proceedings of the Third International Conference on Human-Computer Interaction, Vol.1 on Work with Computers: Organizational, Management, Stress and Health Aspects*. New York, NY, USA: Elsevier Science Inc., 1989, pp. 396–403.

[2] L. Huang, L.-P. Morency, and J. Gratch, "Virtual Rapport 2.0," in *Intelligent Virtual Agents*, ser. Lecture Notes in Computer Science, H. H. Vilhjálmsson, S. Kopp, S. Marsella, and K. R. Thórisson, Eds. Springer Berlin Heidelberg, Sep. 2011, pp. 68–79.

[3] N. Ward and W. Tsukahara, "Prosodic features which cue back-channel responses in English and Japanese," *Journal of pragmatics*, vol. 32, no. 8, pp. 1177–1207, 2000.

[4] L.-P. Morency, I. de Kok, and J. Gratch, "A probabilistic multimodal approach for predicting listener backchannels," *Autonomous Agents and Multi-Agent Systems*, vol. 20, no. 1, pp. 70–84, 2010.

[5] M. Takeuchi, N. Kitaoka, and S. Nakagawa, "Timing detection for realtime dialog systems using prosodic and linguistic information," in *Speech Prosody 2004, International Conference*, 2004.

[6] Y. Okato, K. Kato, M. Kamamoto, and S. Itahashi, "Insertion of interjectory response based on prosodic information," in , *Third IEEE Workshop on Interactive Voice Technology for Telecommunications Applications, 1996. Proceedings*, Sep. 1996, pp. 85–88.

[7] N. Ward, "Using prosodic clues to decide when to produce back-channel utterances," in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference On*, vol. 3. IEEE, 1996, pp. 1728–1731.

[8] S. Fujie, K. Fukushima, and T. Kobayashi, "A conversation robot with back-channel feedback function based on linguistic and non-linguistic information," in *Proc. ICARA Int. Conference on Autonomous Robots and Agents*, 2004, pp. 379–384.

[9] N. Kitaoka, M. Takeuchi, N. R, and N. S, "Response Timing Detection Using Prosodic and Linguistic Information for Human-friendly Spoken Dialog Systems," *Transactions of the Japanese Society for Artificial Intelligence*, vol. 20, no. 3, pp. 220–228, 2005.

[10] L.-P. Morency, I. de Kok, and J. Gratch, "Predicting Listener Backchannels: A Probabilistic Multimodal Approach," in *Intelligent Virtual Agents*. Springer, Berlin, Heidelberg, Sep. 2008, pp. 176–190.

[11] L. Huang, L.-P. Morency, and J. Gratch, "Learning Backchannel Prediction Model from Parasocial Consensus Sampling: A Subjective Evaluation," in *Intelligent Virtual Agents*. Springer, Berlin, Heidelberg, Sep. 2010, pp. 159–172.

[12] D. Ozkan, K. Sagae, and L.-P. Morency, "Latent Mixture of Discriminative Experts for Multimodal Prediction Modeling," in *Proceedings of the 23rd International Conference on Computational Linguistics*, ser. COLING '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 860–868.

[13] I. de Kok, D. Ozkan, D. Heylen, and L.-P. Morency, "Learning and Evaluating Response Prediction Models Using Parallel Listener Consensus," in *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction*, ser. ICMI-MLMI '10. New York, NY, USA: ACM, 2010, pp. 3:1–3:8.

[14] I. de Kok and D. K. J. Heylen, "A survey on evaluation metrics for backchannel prediction models," in *Proceedings of the Interdisciplinary Workshop on Feedback Behaviors in Dialog*, 2012.

[15] K. Ries, "HMM and neural network based speech act detection," in *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*, vol. 1, 1999, pp. 497–500 vol.1.

[16] R. Ruede, M. Müller, S. Stüker, and A. Waibel, "Yeah, right, uh-huh: A deep learning backchannel predictor," *accepted to the International Workshop on Spoken Dialogue Systems*, 2017.

[17] K. Laskowski, M. Heldner, and J. Edlund, "The fundamental frequency variation spectrum," *Proceedings of FONETIK 2008*, pp. 29–32, 2008.

[18] J. Godfrey and E. Holliman, "Switchboard-1 Release 2," https://catalog.ldc.upenn.edu/ldc97s62, 1993.

[19] D. Harkins and others, "ISIP Switchboard word alignments," https://www.isip.piconepress.com/projects/switchboard/, 2003.

[20] D. Jurafsky, C. Van Ess-Dykema, and others, "Switchboard discourse language modeling project," http://compprag.christopherpotts.net/swda.html, 1997.

[21] L. Levin, A. Lavie, M. Woszczyna, D. Gates, M. Gavaldá, D. Koll, and A. Waibel, "The Janus-III Translation System: Speech-to-Speech Translation in Multiple Domains," *Machine Translation*, vol. 15, no. 1, pp. 3–25, 2000.

[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *arXiv:1301.3781 [cs]*, Jan. 2013.

[23] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke *et al.*, "The icsi meeting corpus," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, vol. 1. IEEE, 2003, pp. I–I.

[24] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, May 2016.

[25] S. Dieleman, J. Schlüter, C. Raffel, E. Olson, S. K. Sønderby, and others, "Lasagne: First release." Aug. 2015.

[26] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10). Society for Artificial Intelligence and Statistics*, 2010.