Backchannel Prediction for Conversational Speech Using Recurrent Neural Networks

Definition

What are backchannels?

- Nodding / head movement
- Eye gaze shift
- Short phrases like “uh-huh”, “yeah”, “right”
- Different from culture to culture (e.g. Japanese)

Motivation / Goal

- BCs help build *rapport* (comfortableness between conversation partners)

$\rightarrow$ Improve conversation with artificial assistants

How?

- Simplify backchannels to only short phrases
- Predict when to emit backchannels
- (Predict what kind of backchannel to emit)

Related Work

Related Work

Common approach: manually tuned rules.

Ward (2000):

produce backchannel feedback after 700ms of detection of:

- *a region of pitch less than the 26th-percentile pitch level and*
- *continuing for at least 110 milliseconds,*
- *coming after at least 700 milliseconds of speech,*
- *providing you have not output back-channel feedback within the preceding 800 milliseconds*

Almost always based on pitch and power

Related Work

Common approach: manually tuned rules.

- error-prone
- a lot of manual work
- hard to generalize

semi-automatic approaches, e.g. Morency (2010)

Preprocessing

Dataset

Switchboard dataset:

- 2400 English telephone conversations
- 260 hours total
- Randomly selected topics
- Transcriptions and word alignments that include BC utterances

BC Utterance Selection

- Get a list of all backchannel phrases
- BC phrase list from the *Switchboard Dialog Act Corpus* (SwDA)

BC Utterance Selection

- Get a list of all backchannel phrases
- BC phrase list from the *Switchboard Dialog Act Corpus* (SwDA)

SwDA incomplete

→ Identify utterances only from their text

Something like “uh” can be a disfluency or a BC.

→ only include phrases with silence or BC before them.

Training Area Selection

Figure 1: Sample Audio Segment

Training Area Selection

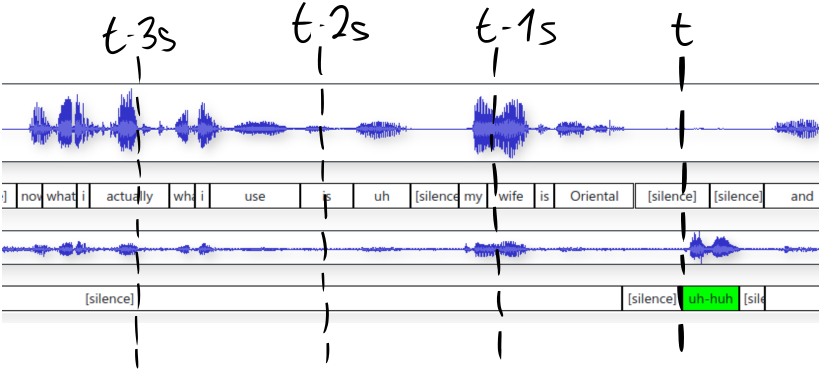
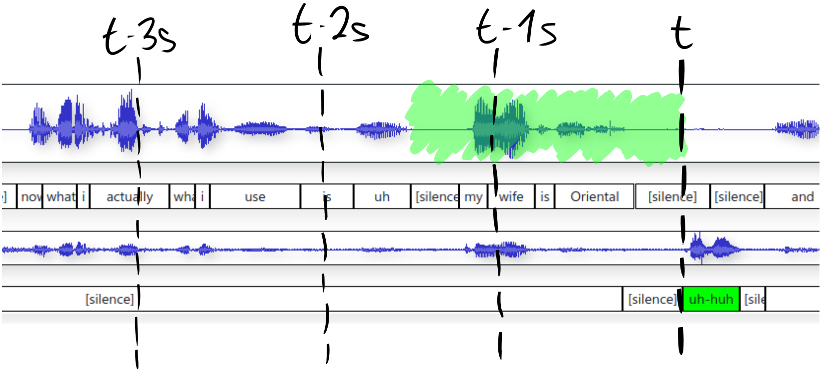


Figure 2: Sample Audio Segment

Training Area Selection



Training Area Selection

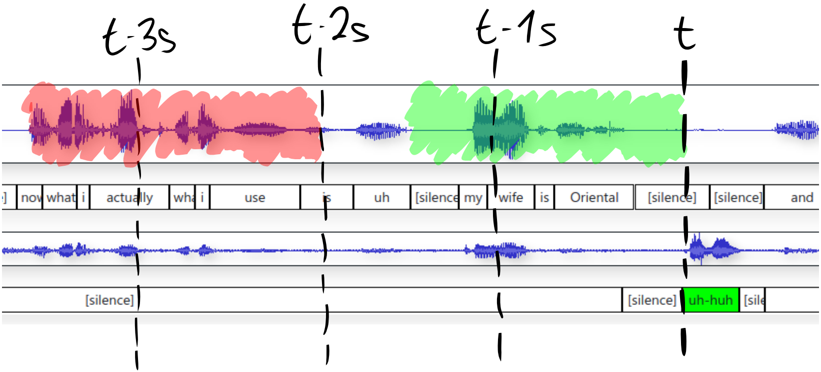


Figure 3: Pos/Neg Training areas

→ Balanced data

Feature Selection

- Acoustic features like power, pitch
- Linguistic features (from the transcriptions)

Feature Selection – Acoustic

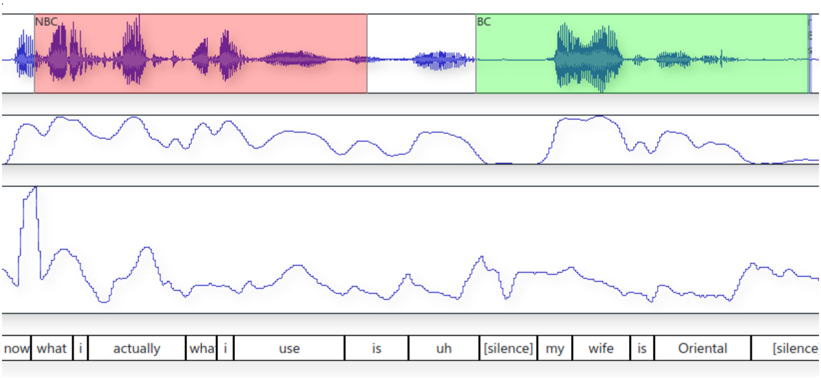


Figure 4: Audio, Power, Pitch

Feature Selection – Linguistic

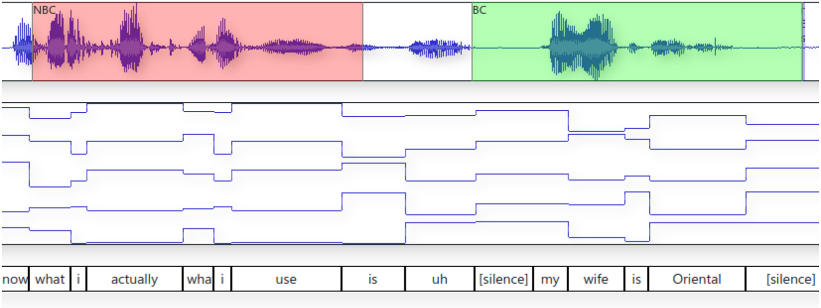


Figure 5: Word2Vec

Neural network design

Input layer

$$\text{for } t\text{-}0\,\text{ms} \begin{cases} \text{Power} = 0.82 \\ \text{Pitch} = 0.55 \end{cases}$$

Figure 6:

Input layer

$$\text{for } t-1500\text{ms} \begin{cases} \text{Power} = \ldots \\ \text{Pitch} = \ldots \end{cases}$$

$$\vdots$$

$$\text{for } t-100\text{ms} \begin{cases} \text{Power} = \ldots \\ \text{Pitch} = \ldots \end{cases}$$

$$\text{for } t-0\text{ms} \begin{cases} \text{Power} = 0.82 \\ \text{Pitch} = 0.55 \end{cases}$$

Figure 7:

Input layer

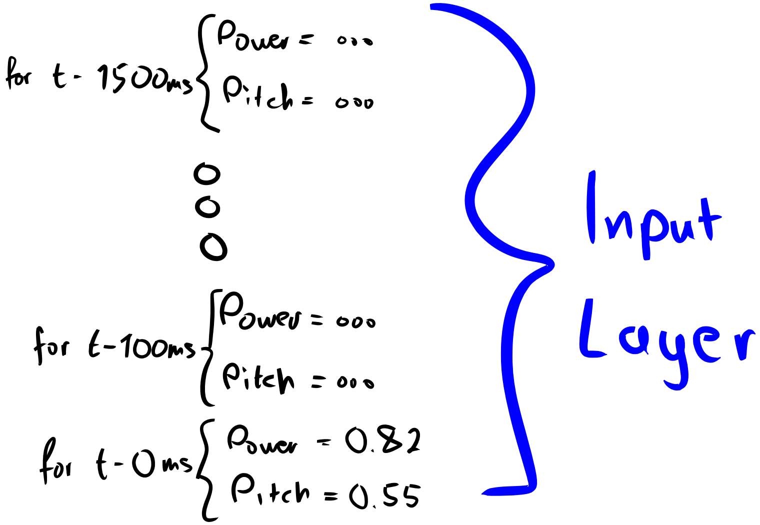


Figure 8:

Hidden layers (Feed forward)

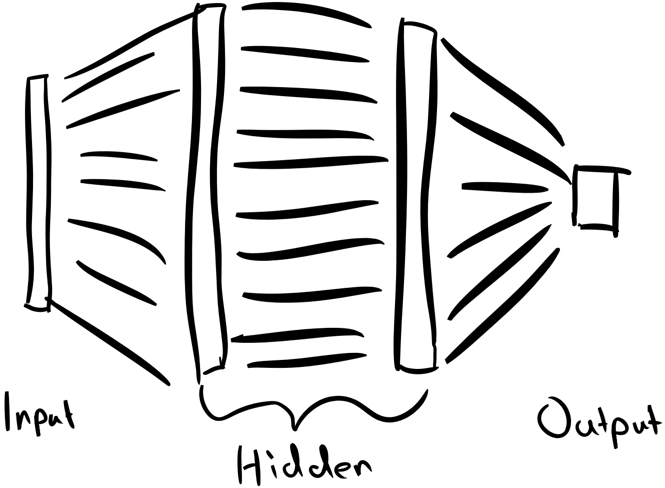


Figure 9:

Recurrent NNs

BCs are more probable after a longer period without BCs.

$\rightarrow$ Use RNN / LSTM

Recurrent NNs

BCs are more probable after a longer period without BCs.

$\rightarrow$ Use RNN / LSTM

LSTM is able to take into account it's own past internal state.

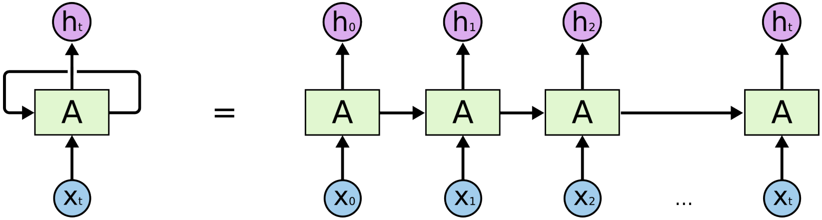


Figure 10: Recurrent Neural Net architecture (Christopher Olah)

Postprocessing

NN output is

- a value between 0 and 1
- quickly changing
- noisy

Postprocessing – Low-pass filter

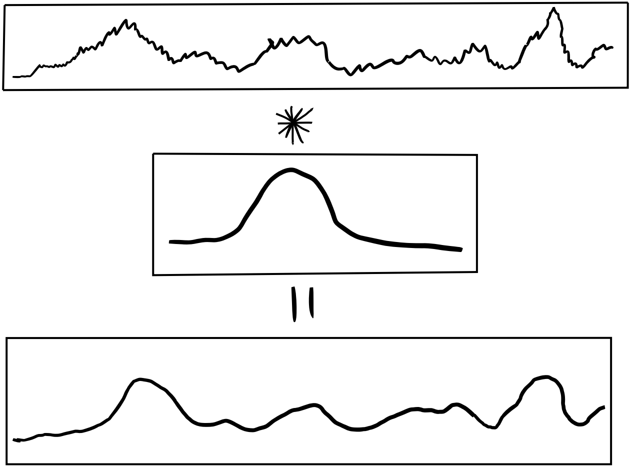
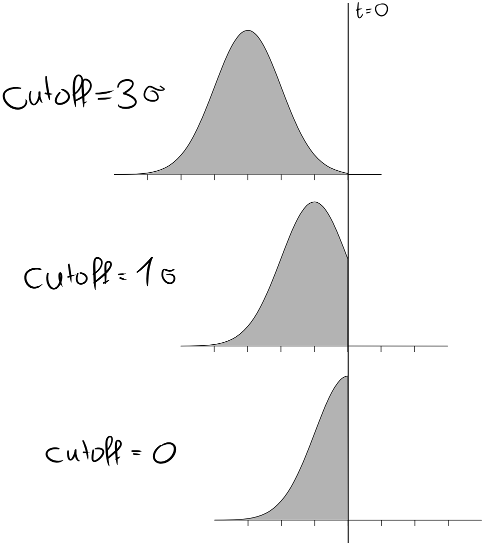


Figure 11: Low-pass convolution

Postprocessing – Low-pass filter

Gauss filter looks into future

$\rightarrow$ Cut off filter and shift it



Thresholding / Triggering

- Use areas of output > threshold ($0 < t < 1$)
- Trigger at local maximum

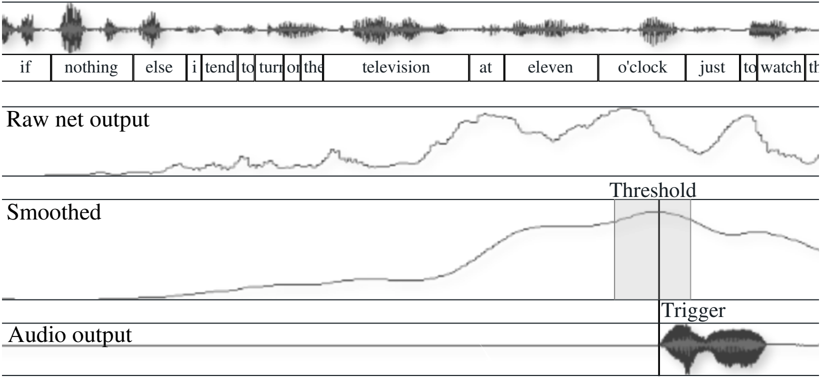


Figure 13: Example of the postprocessing process.

Evaluation

Objective Evaluation

- Precision (portion of predictions that were correct)
- Recall (portion of correct BCs that were predicted)
- F1-Score (harmonic mean of Precision and Recall)

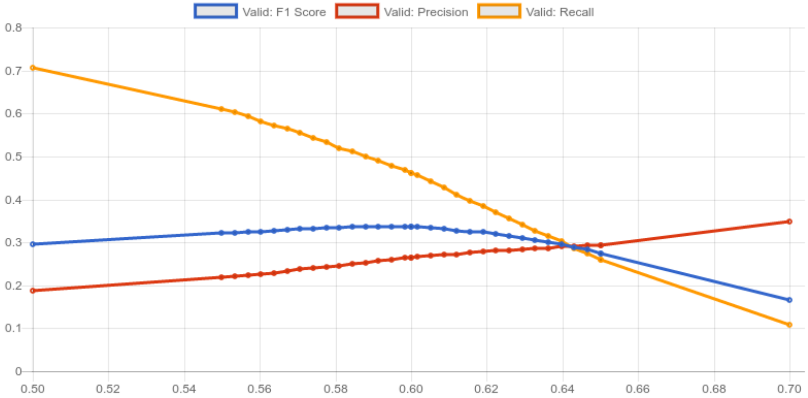


Figure 14: Evaluating the performance of a network while varying the threshold. Note the inverse relationship between *Precision* and *Recall*.

Lots of parameters to tune

- Context width
- Context stride
- Which features
- NN depth
- NN layer sizes
- LSTM vs Feed forward
- Trigger threshold
- Gaussian filter sigma
- Gaussian filter cutoff
- Prediction delay

Lots of parameters to tune

manually through trial and error:

- Context width
- Context stride
- Which features
- NN depth
- NN layer sizes
- LSTM vs Feed forward

automatically with Bayesian optimization:

- Trigger threshold
- Gaussian filter sigma
- Gaussian filter cutoff
- Prediction delay

Results

Context width

Context	Precision	Recall	F1-Score
500 ms	0.219	0.466	0.298
1000 ms	0.280	0.497	0.358
1500 ms	0.305	0.488	**0.375**
2000 ms	0.275	0.577	0.373

Table 1: Results with various context lengths. Performance peaks at 1500 ms.

LSTM vs FF

Layers	Parameter count	Precision	Recall	F1-Score
FF (56 : 28)	40k	0.230	0.549	0.325
FF (70 : 35)	50k	0.251	0.468	0.327
FF (100 : 50)	72k	0.242	0.490	0.324
LSTM (70 : 35)	38k	0.305	0.488	**0.375**

Table 2: LSTM outperforms feed forward architectures.

Layer sizes

Layer sizes	Precision	Recall	F1-Score
100	0.280	0.542	0.369
50 : 20	0.291	0.506	0.370
70 : 35	0.305	0.488	**0.375**
100 : 50	0.303	0.473	0.369
70 : 50 : 35	0.278	0.541	0.367

Table 3: Comparison of different network configurations. Two LSTM layers give the best results.

Features

Features	Precision	Recall	F1-Score
power	0.244	0.516	0.331
power, pitch	0.307	0.435	0.360
power, pitch, mfcc	0.278	0.514	0.360
power, ffv	0.259	0.513	0.344
power, ffv, mfcc	0.279	0.515	0.362
power, pitch, ffv	0.305	0.488	**0.375**
word2vec$_{dim=30}$	0.244	0.478	0.323
power, pitch, word2vec$_{dim=30}$	0.318	0.486	0.385
power, pitch, ffv, word2vec$_{dim=15}$	0.321	0.475	0.383
power, pitch, ffv, word2vec$_{dim=30}$	0.322	0.497	**0.390**
power, pitch, ffv, word2vec$_{dim=50}$	0.304	0.527	0.385

Table 4: Results with various input features, separated into only acoustic features and acoustic plus linguistic features.

Other research

Predictor	Precision	Recall	F1-Score
Baseline (random)	0.042	0.042	0.042
Müller et al. (offline)	–	–	0.109
Our results (offline, context of -750 ms to 750 ms)	0.114	0.300	**0.165**
Our results (online, context of -1500 ms to 0 ms)	0.100	0.318	0.153

Table 5: Comparison with previous research.

Varying margin of error

Margin of Error	Constraint	Precision	Recall	F1-Score
−200 ms to 200 ms		0.172	0.377	0.237
−100 ms to 500 ms		0.239	0.406	0.301
−500 ms to 500 ms		0.247	0.536	0.339
0 ms to 1000 ms	Baseline (random)	0.079	0.323	0.127
	Only acoustic features	0.294	0.488	0.367
	Acoustic and linguistic features	0.312	0.511	0.388

Table 6: Results with various margins of error used in other research. Performance improves with a wider margin width and a later margin center.

Survey

Randomly show participants 6 samples of the following categories

1. Random predictor
2. NN predictor
3. Ground truth

Survey

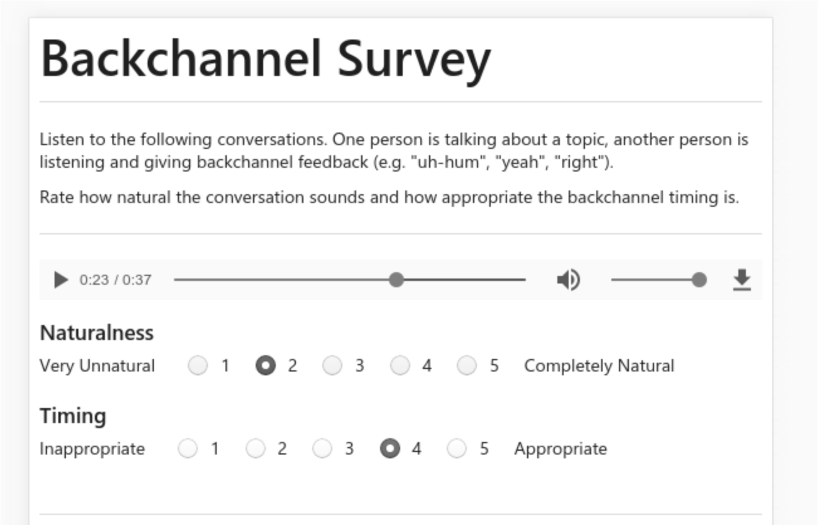


Figure 15: Screenshot of the survey interface.

Survey Results

Predictor	Sample	Timing	Naturalness	Sample Size
random	average	2.33 points	2.63 points	40
nn	average	3.48 points	3.08 points	40
truth	average	4.20 points	4.08 points	40

Table 7: Average user ratings of different BC predictors

Addendum

SwDA categories

	name	act_tag	example	full_count
1	Statement-non-opinion	sd	Me, I'm in the legal department.	75145
2	**Acknowledge (Backchannel)**	b	Uh-huh.	38298
3	Statement-opinion	sv	I think it's great	26428
4	Agree/Accept	aa	That's exactly it.	11133
5	Abandoned or Turn-Exit	%	So, -	15550
6	Appreciation	ba	I can imagine.	4765
7	Yes-No-Question	qy	Do you have to have any special training?	4727

Figure 16: SwDA categories

Context stride

Stride	Precision	Recall	F1-Score
10ms	0.290	0.490	0.364
20ms	0.305	0.488	**0.375**
40ms	0.285	0.498	0.363

Table 8: Results with various context frame strides.