Towards Deep End-of-Turn Prediction for Situated Spoken Dialogue Systems

Angelika Maier[1], *Julian Hough*[2], *David Schlangen*[2]

[1]Semantic Computing Group, CITEC, Bielefeld University
[2]Dialogue Systems Group, CITEC, Bielefeld University

angelika.maier@uni-bielefeld.de, julian.hough@uni-bielefeld.de,
david.schlangen@uni-bielefeld.de

Abstract

We address the challenge of improving live end-of-turn detection for situated spoken dialogue systems. While traditionally silence thresholds have been used to detect the user's end-of-turn, such an approach limits the system's potential fluidity in interaction, restricting it to a purely reactive paradigm. By contrast, here we present a system which takes a predictive approach. The user's end-of-turn is predicted live as acoustic features and words are consumed by the system. We compare the benefits of live lexical and acoustic information by feature analysis and testing equivalent models with different feature sets with a common deep learning architecture, a Long Short-Term Memory (LSTM) network. We show the usefulness of incremental enriched language model features in particular. Training and testing on Wizard-of-Oz data collected to train an agent in a simple virtual world, we are successful in improving over a reactive baseline in terms of reducing latency whilst minimising the cut-in rate.

Index Terms: turn-taking, end-of-turn detection, VAD, human-computer interaction

1. Introduction

Spoken dialogue systems in situated settings, where the system and user have access to a shared environment, are becoming increasingly prevalent in everyday life and in research. The most recent applications include those within industrial and social robots and virtual personal assistants [1, 2, 3, 4].

These systems carry out specific tasks in response to spoken commands from the user, such as selecting an object to manipulate in a shared workspace. For effective and timely responses to user commands, whether verbal or non-verbal, the system must be able to predict the end of the user's turn to ensure a fluid interaction with little delay. In accordance with [5]'s observation that "human turn-taking is so precise that next speakers tend to start with no gap and no overlap" ([6], p.555), in this paper we present a practical framework for learning automatic *predictive end-of-turn detection* in situated interactions from data.

2. End-of-Turn Detection

There are two different debates on end-of-turn ('EOT' largely from here) detection and the turn-taking cues used in human-human conversation: firstly, the *reactive* [7, 8, 9] vs. *predictive* [5, 10, 11, 12] paradigm debate and, secondly, the debate as to whether *lexical and syntactic (linguistic)* information [11] or *acoustic* [13] information is sufficient or most crucial for predicting and detecting an EOT.

Much work has gone into leveraging acoustic information for reactive EOT detection systems [14, 15], using silence duration as a feature with an optimization based on minimizing the trade-off between latency in detecting the EOT and avoiding 'cut-in's to ongoing turns [16].

We argue the reactive approach limits a dialogue system's potential fluidity in interaction. The system we describe here models the predictive viewpoint, following [17, 18], and the results will be compared to a reactive baseline, where an EOT is simply predicted after a silence threshold. We compare the benefits of linguistic versus acoustic information by testing equivalent models with different feature sets. We use a common deep learning architecture, the Long Short-Term Memory (LSTM) recurrent neural network to investigate this, posing the following research questions:

- Is it possible to improve over a silence-thresholding baseline for EOT detection by using live acoustic and linguistic features?
- To what degree do acoustic features and linguistic features affect EOT detection success when using a sequential deep learning architecture?
- Is it possible to use the output from a sequential deep learning network in decoding to control the trade-off between latency and cut-in rate?

3. Data and Features

We use the TAKE corpus from the `PentoRef` release [19], a German corpus of situated Wizard-Of-Oz (WOZ) interactions. Excluding one non-native German speaker, the data is from 6 participants (18–30 years, mean = 24.1; 4 male, 2 female). We use 933 episodes (the participants' turn plus following silence) in total for our experiments, divided into training, heldout and test sets as explained in §5. The participants' task was to instruct the system to select a Pentomino puzzle piece from a virtual scene of several pieces, while in reality they spoke to a human Wizard who selected the Pentomino he thought the participant intended as soon as he could recognize it in the virtual scene. A standard user turn can be seen in (1):

(1) ähm das grüne Objekt oben links *[pause]* in der Ecke
um the green object top left [pause] in the corner

The turns we focus on consist of a single dialogue act type, namely instructions. We did not include the confirmation utterances which frequently follow the event of the Wizard selecting the piece on the board. 52% of the turns featured **mid-turn pauses** after the speaker had begun the turn, such as the *[pause]* above, before resuming, whilst the others consisted of a single utterance. In either case, we characterize the end of the user's final utterance before the Wizard responded as the gold standard end-of-turn time point.

To simulate the conditions of a live dialogue system, we extract feature vectors for every 10ms frame of an episode. In line

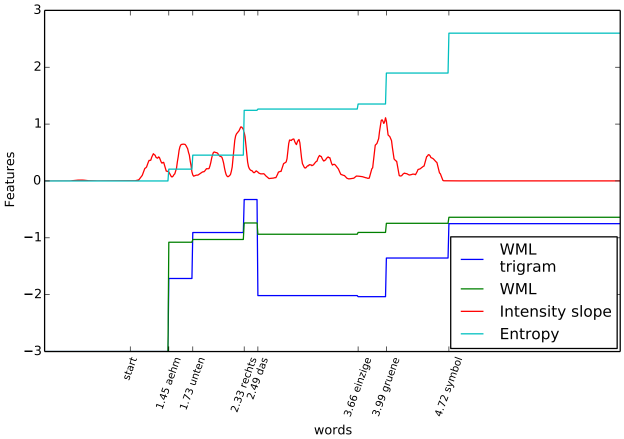

Figure 1: *Local tri-gram and utterance WML values and Entropy values and Intensity slope for an example turn. X-axis annotations are words and their end times.*

with our research questions we use both linguistic and acoustic features, which we describe below.

3.1. Acoustic Features

Following [18, 20, 13] we use raw pitch (F0), a smoothed F0 contour, the Root Mean Square (RMS) signal energy, the logarithmized signal energy, intensity, loudness (as the normalised intensity raised to a power of 0.3) and derived features for each 10ms frame as acoustic features.

For the derived acoustic features, we use the window of the last 50ms (5 frames of an audio signal) of the RMS energy values, and the previous 150ms (15 frames) of intensity and raw pitch values, an approximation to the length of a syllable, to compute slope and mean values for those windows.

We follow [15] and [21] in using the duration of vowels, and additionally the duration of nasals and fricatives. Instead of raw duration we use the z-score, the normalization over the standard deviation of duration of the last vowel/nasal/fricative for that speaker.[1]

The acoustic features were extracted with `OpenSmile`[2] and phoneme durations were extracted offline using Forced Alignment, but only made available to our systems at the *end* of each phoneme. In total there are 16 acoustic features per time-step. All values of each frame were centered by removing the mean value of each feature, then scaled by dividing non-constant features by the standard deviation.[3]

3.2. Linguistic Features with Enriched Language Models

For linguistic features, we make the words available to our detection system on the last frame of the word, simulating perfect zero-latency speech recognition (ASR). The words were automatically Forced Aligned and then manually corrected, substituting all partial or mispronounced words with the corresponding complete word, all mis-spelled words with the correct spelling, and words uttered in a dialect with standard German forms. We use insights on modelling grammaticality and (dis-)fluency [22, 23] by using information-theoretic features from enriched language models, rather than using the word values themselves. In this paper we use Weighted Mean Log trigram probability (WML) [22], which is an approximation to incremental syntactic probability by factoring out lexical frequency, as given in (1) where p_{kn} is a Kneser-Ney smoothed trigram language model.

$$\mathit{WML}(w_0 \ldots w_n) = \frac{\sum_{i=2}^{n} \log_2 p_{kn}(w_i \mid w_{i-2}, w_{i-1})}{-\sum_{j=2}^{n} \log_2 p_{kn}(w_j)} \tag{1}$$

In the spirit of Hidden Event Language Models (HELMs) [24] we introduce a hidden word to model our event of interest, `<EOT/>`, which is appended to the end of turns in training the language model. Then in testing we compute the probability of `<EOT/>` occurring after current prefix of the utterance consumed so far. Specifically, we compute two related *WML* features: the first for the latest word w_t consumed by considering the trigram $w_{t-1}, w_t,$ `<EOT/>` at time-step t; the second is the WML of $w_1, ..., w_t,$ `<EOT/>` for the whole utterance. While the first measure estimates the likelihood of an EOT given the local context, the latter computes a smoother probability which factors in the *WML* of the utterance consumed so far.

Furthermore, we go beyond [15]'s use of the current length of the utterance as an approximation to information content by word-by-word computation of *Entropy* (the negative sum of probability weighted log probabilities of the trigrams in the utterance). This serves as a good compliment to the WML features in that it should discourage classifiers from predicting `<EOT/>` until there is sufficient information in the ongoing turn– in a simple situated dialogue situation like ours this is similar across turns, regardless of length.

We extract these linguistic features using STIR (STrongly Incremental Repair detection) [23]'s language models.[4] The language model is trained iteratively over the six speakers with one speaker held out on each iteration for which the 3 features are obtained. We do this to avoid obtaining values for speakers whose utterances are in the language model's training data– this would provide over-fitted features which are unavailable at testing time with unknown speakers. An example of how the feature values evolve over a turn is given in Fig. 1 – overall WML (green line) peaks on the final word as desired, whilst the local trigram WML measure (blue) is more jittery. The steadily rising entropy measure (turquoise, highest line) should allow a classifier to take account of there being a sufficient amount of information in the utterance for a potential EOT.

3.3. Feature Analysis for Predictive Ability

We would like to know how predictive these features are, and in the following experiments, how they could be complementary to one another. We examine a few of them here briefly.

For the linguistic features, Fig. 2 (left) shows the distribution of *WML* trigram values for final and non-final words, when tested on utterances not seen during language model training. There is some overlap in the distributions, however there is still some useful separation (final words mean= -0.648 (std.=0.493), non-final words mean= -1.121 (std.=0.529)).

The central graph in Fig. 2 plots the mean entropy values for each of the 6 speakers' utterances as a function of time over

[1] This is the only feature where we use information about the rest of the speaker's utterances to obtain values. Given the small number of speakers this is the only way the feature could be useful. In future work we intend to compute the z-score without using the same speaker's utterances.

[2] `http://audeering.com/research/opensmile/`

[3] Concretely, saved as a Sci-kit learn StandardScaler object– see `http://scikit-learn.org/stable/modules/preprocessing.html#preprocessing-scaler`

[4] Available within STIR at `https://bitbucket.org/julianhough/stir`

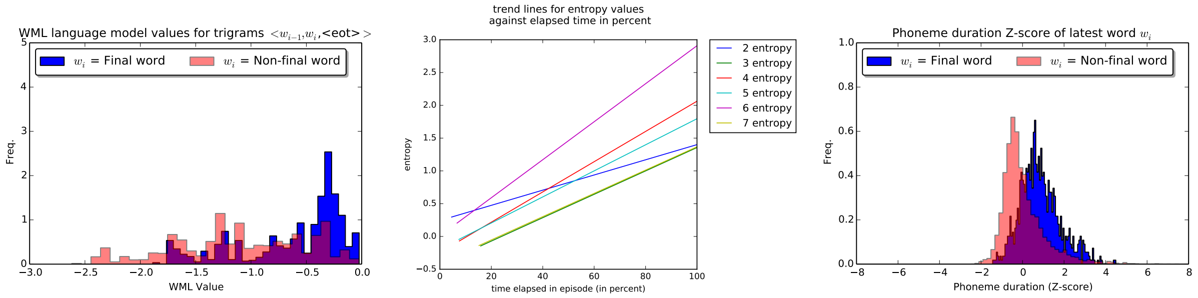

Figure 2: *Features: Local tri-gram WML values (left) overall Entropy values as a function of time for each speaker in the corpus (middle) and vowel/nasal/fricative phoneme durations (right) for predicting the end-of-turn.*

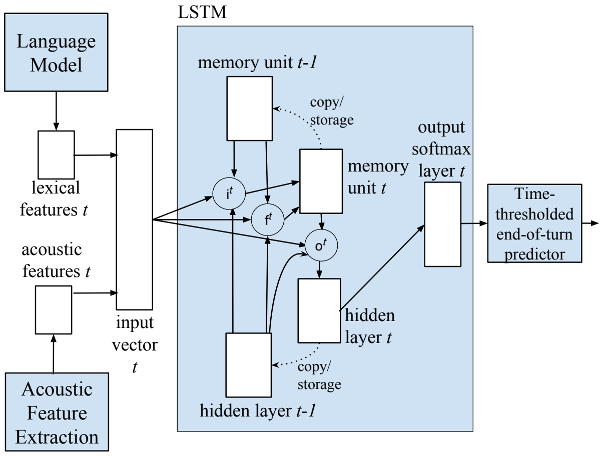

Figure 3: *Overall architecture of the system*

the turn. In line with Fig. 1 this generally rises towards the end of turns across speakers.

For our vowel, fricative and nasal phoneme duration Z-score feature there is a lengthening turn-finally as can be seen in Fig. 2 (right) (final words mean= 0.846 (std.=0.952), non-final words mean= 0.05 (std.=1.084)).

4. EOT Detection System Components

As we have features which correlate well with end-of-turn events, we investigate how predictive they can be in a time-series classifier by using them in a simple deep learning set-up with a simple decoder as a proof of concept.

We pose the problem as a sequence labelling task over 10ms windows, where the three labels can be speech (S), mid-turn pause (MTP) and end-of-turn (EOT). We are principally concerned with the first point at which the EOT label is predicted, but the other labels form part of this task. The overall architecture of the system is as in Fig. 3.

4.1. LSTM

We use the Keras LSTM model [25], using an LSTM layer with the standard input, output and forget gates with the $tanh$ activation, and a softmax output layer over the three classes. The input to the network is a concatenated vector of the 16 acoustic and 3 linguistic features explained above. We train using negative log likelihood loss (NLL) as a cost function and the Adam optimizer [26]. We experimented with different hidden layer sizes with grid search and an LSTM layer with 68 nodes worked best on heldout data. To train the network we use a batch size of 10 and maintain the hidden state from one batch to the next.

4.2. Decoding optimization through windowing

The task of the system is to decide, given there is a speech-to-silence boundary, whether this is the end of the user's turn. To simulate that, we assume that we have access to perfect Voice Activity Detection (VAD), so whilst in training we train on the label 'S', in decoding we interpolate the output from our networks with the gold standard information about whether there was speech or not, and hence the principal decision is whether to label the current frame as MTP or EOT.

We experimented with several thresholding techniques to optimize performance. We found a simple windowing technique with two different windows was most effective for controlling the trade-off between predicting the EOT as quickly as possible, and not cutting in. The **arg max window** is used for computing the class with the highest probability over a fixed window of previous frames. Making this window longer increases the latency of detection whilst ensuring more confidence. The **silence threshold window** is the number of frames of silence (or frames of an MTP are predicted) beyond which an EOT is predicted automatically for the current frame (and the previous frames originally labelled MTP are also reclassified). As with standard silence thresholding, making this shorter risks cutting in, while making it longer risks increasing latency.

We experimented with different values for the arg max window size and silence threshold window size, and found that a ratio of 0.3 from the latter to the former value gave the best performance on the heldout data.

5. Experimental Set-up

We test our system against a silence-threshold baseline which predicts an EOT once a given threshold of silence has been reached. We use thresholds of 50ms–6000ms in 50ms increments to give a target curve. Given this technique has been used in previous work with good effect [15], while simple, this is a competitive baseline.

We evaluate against two principal metrics of interest: the **mean latency** at which a turn is detected by a system after the gold EOT frame, and the **cut-in rate**, which is the proportion of times the system predicts an EOT before the gold standard EOT point. Given achieving the best trade-off between these two, as per [16], is the objective, we formalize this as follows as a mean between the cut-off rate and mean latency normalised by maximal acceptable values for each as in (2).

$$\textit{EOT Trade-Off} = 0.5 \times \left[\frac{\textit{cut-in rate}}{\textit{max cut-in rate}} + \frac{\textit{mean latency}}{\textit{max latency}} \right] \quad (2)$$

In this paper we set *max cut-in rate* to 100% and the *max latency* to 10 seconds. The precise values of these can be tested empirically, but this simple assumption seems to work well for a situated dialogue system setting– a wait of 10 seconds for a system response is likely to be as irritating as the system cutting in every turn, and a very small latency of one tenth of a second is equivalently effective as only cutting in once every one-hundred turns. The lower this value, the better, and systems must do well in both individual metrics to do well overall.

We employ a cross-validation strategy for evaluation to simulate the demands of a real system. Our systems are trained on 5 speakers, then tested on the heldout sixth one. We divided each fold into heldout and test data so we could experiment before getting our final results.[5] After training the LSTM, we decode using different silence threshold back-offs incrementing from 50ms up to 2000ms in 50ms increments, and then from 2000ms to 6000ms using 500ms increments with the arg max window size always 0.3 times this value. The purpose is to see if good trade-offs can be found which can beat the silence threshold baseline when operating with different latencies.

We run all the different threshold settings across an LSTM trained on the acoustic features alone, one trained on the linguistic features alone, and one trained on all features. We use early stopping on heldout data with a patience of 5 epochs without improvement.

6. Results

Our results are as in Table 1. The best overall performance is in the top block of the table and the best performance for detection latencies under 750ms and 500ms are in the second and third blocks, respectively. As can be seen, our best LSTM system outperforms the baseline in all three categories in terms of the best trade-off, with the overall best having an average of roughly **1.195 seconds** of latency with a cut-in rate of **18%**, while it can perform well at the lower latency settings achieving a latency of 0.733 seconds with a cut-in rate of 24% and a latency of 0.494s with a cut-in rate of 29%. The model using both linguistic and acoustic features outperforms the individual settings, and only in the <500ms part of the evaluation does one of our systems (linguistic features only) get beaten by the baseline.

The curve in Fig. 4 shows the performance of the system with different feature sets. As can be seen there is some room under the baseline curve for our LSTM that uses linguistic and acoustic features. A T-test shows the three LSTM systems have points on the curve with mean trade-off values significantly lower (better) than the baseline, while the model with joint acoustic and linguistic features is significantly better than both the acoustic features only ($p<0.02$) and linguistic features only systems ($p<0.04$). There is no significant difference between the linguistic features only and acoustic features only system performances, showing that the fusion of the feature sets is maximising the utility of both feature sets when fused together.

There is significant performance variation over the different speaker folds, with best trade-offs ranging from a modest 0.294 (latency=3.683s, cut-in=21.9%) in a speaker with unusually long pauses, to the best performance of **0.024 (latency=0.153s, cut-in=3.3%)**. The variation is part of a general problem we face as dialogue system designers for systems with a small number of diverse users, but we plan to do future work on online adaptation [27].

System (silence threshold, arg max window)	Latency (ms)	Cut-in rate(%)	Trade-off
LSTM Acoust+Ling (2000,600)	1195.3	18.0	**0.150**
LSTM Ling (2000,600)	1396.2	17.8	0.159
LSTM Acoust (1650,490)	1101.7	21.2	0.161
Baseline (1600,-)	1600.0	17.5	0.168
LSTM Acoust+Ling (1200,360)	732.6	24.2	**0.160**
LSTM Ling (1000,300)	717.7	26.7	0.169
LSTM Acoust (1100, 330)	729.6	27.2	0.173
Baseline (750,-)	750.0	27.2	0.174
LSTM Acoust+Ling (800,240)	494.1	29.2	**0.173**
LSTM Acoust (750,220)	485.8	30.7	0.178
Baseline(500,-)	500.0	31.0	0.180
LSTM Ling (650,190)	460.2	32.0	0.183

Table 1: *Results from our systems against the baseline.*

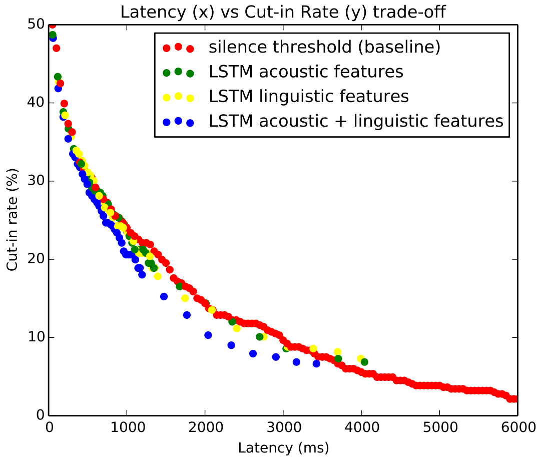

Figure 4: *The trade-off between latency (x) and cut-in rate (y) with our LSTM systems against a silence-thresholding baseline.*

7. Conclusion

Our work most closely follows [15] in building a system which allows the trade-off between latency and cut-in rate in end-of-turn detection. We specifically do this for a situated dialogue system setting, and use an LSTM with threshold-based decoding which uses acoustic and linguistic features to predict end-of-turn events incrementally, improving over a reactive silence thresholding baseline. We show the combination of linguistic and acoustic features yields improvements when used together, suggesting the fusion in the LSTM is working effectively. We suggest our method could be applied to both dialogue systems and segmentation of human-human dialogues, where according to [5] there is often no silence between turns.

We are aware of the simplifying assumptions we have made here. In future work, we will make our system fully live by testing it with the input of real users, and therefore using output from a real VAD and ASR. Additionally, we aim to use our predictive approach in dialogue situations that are less easily predictable such as user interruptions or emotive speech.

8. Acknowledgements

We thank the InterSpeech reviewers for their helpful comments. This work was supported by the Cluster of Excellence Cognitive Interaction Technology 'CITEC' (EXC 277) at Bielefeld University, funded by the German Research Foundation (DFG), and the DFG-funded DUEL project (grant SCHL 845/5-1).

[5] The experiments are fully reproducible from the repositories `https://bitbucket.org/anmaier/eot_detection` and `https://bitbucket.org/anmaier/eot_detection_data`.

9. References

[1] P. Lison, "Structured probabilistic modelling for dialogue management," Ph.D. dissertation, University of Oslo, 2013.

[2] R. Yaghoubzadeh, M. Kramer, K. Pitsch, and S. Kopp, "Virtual agents as daily assistants for elderly or cognitively impaired people," in *International Workshop on Intelligent Virtual Agents*. Springer, 2013, pp. 79–91.

[3] P. Rane, V. Mhatre, and L. Kurup, "Study of a home robot: Jibo," in *International Journal of Engineering Research and Technology*, vol. 3, no. 10 (October-2014). ESRSA Publications, 2014.

[4] J. Hough and D. Schlangen, "Investigating fluidity for human-robot interaction with real-time, real-world grounding strategies," in *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Los Angeles: Association for Computational Linguistics, September 2016, pp. 288–298. [Online]. Available: http://www.aclweb.org/anthology/W16-3637

[5] H. Sacks, E. A. Schegloff, and G. Jefferson, "A simplest systematics for the organization of turn taking for conversation," *Language*, vol. 50, pp. 696–735, 1974.

[6] M. Heldner and J. Edlund, "Pauses, gaps and overlaps in conversations," *Journal of Phonetics*, vol. 38, no. 4, pp. 555–568, 2010.

[7] S. J. Duncan, "Some signals and rules for taking speaking turns in conversations," *Journal of Personality and Social Psychology*, vol. 23, no. 2, pp. 283–292, 1972.

[8] A. Kendon, "Some functions of gaze – direction in social interaction," *Acta Psychologica*, vol. 26, pp. 22–63, 1967.

[9] V. H. Yngve, "On getting a word in edgewise," in *Papers from the sixth regional meeting of the Chicago Linguistic Society*. Chicago Linguistic Society, 1970, pp. 567–578.

[10] G. Bockgaard, "Syntax och prosodi vid turbytesplatser: Till beskrivningen av svenskans turtagning," in *Interaktion och kontext: Nio studier av svenska samtal*, 1st ed., E. Engdahl and A.-M. London, Eds. Lund: Studentlitteratur, 2007, pp. 139–183.

[11] J. P. De Ruiter, H. Mitterer, and N. J. Enfield, "Predicting the end of a speakers turn; a cognitive cornerstone of conversation," *Language*, vol. 82, no. 3, pp. 515–535, 2006.

[12] S. C. Levinson, *Pragmatics*, ser. Cambridge textbooks in linguistics. Cambridge [u.a.]: Cambridge Univ. Pr., 1983. [Online]. Available: http://digitool.hbz-nrw.de:1801/webclient/DeliveryManager?pid=2865895&custom_att_2=simple_viewerInterna:Inhaltsverzeichnis

[13] S. Bögels and F. Torreira, "Listeners use intonational phrase boundaries to project turn ends in spoken interaction," *Journal of Phonetics*, vol. 52, pp. 46–57, 2015.

[14] A. Raux and M. Eskenazi, "A finite-state turn-taking model for spoken dialog systems," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2009, pp. 629–637.

[15] ——, "Optimizing the turn-taking behavior of task-oriented spoken dialog systems," *ACM Trans. Speech Lang. Process.*, vol. 9, no. 1, 5 2012.

[16] A. Raux, "Flexible turn-taking for spoken dialog systems," *http://www-2.cs.cmu.edu/ antoine/thesis_antoine.pdf*, 5 2012. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.151.9300

[17] D. Schlangen, "From reaction to prediction: Experiments with computational models of turn-taking," *Proceedings of Interspeech 2006, Panel on Prosody of Dialogue Acts and Turn-Taking*, 2006.

[18] M. Atterer, T. Baumann, and D. Schlangen, "Towards incremental end-of-utterance detection in dialogue systems," *Proceedings of the 22nd International Conference on Computational Linguistics*, 2008.

[19] S. Zarrieß, J. Hough, C. Kennington, R. Manuvinakurike, D. DeVault, R. Fernández, and D. Schlangen, "Pentoref: A corpus of spoken references in task-oriented dialogues," in *10th edition of the Language Resources and Evaluation Conference*, 2016.

[20] K. Sakhnov, E. Verteletskaya, and B. Simak, "Dynamical energy-based speech/silence detector for speech enhancement applications," 2009.

[21] L. Ferrer, E. Shriberg, and A. Stolcke, "A prosody-based approach to end-of-utterance detection that does not require speech recognition," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, vol. 1. IEEE, 2003, pp. I–608.

[22] A. Clark, G. Giorgolo, and S. Lappin, "Statistical representation of grammaticality judgements: the limits of n-gram models," in *Proceedings of the Fourth Annual Workshop on Cognitive Modeling and Computational Linguistics (CMCL)*. Sofia, Bulgaria: Association for Computational Linguistics, August 2013, pp. 28–36. [Online]. Available: http://www.aclweb.org/anthology/W13-2604

[23] J. Hough and M. Purver, "Strongly incremental repair detection," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, October 2014, pp. 78–89. [Online]. Available: http://www.aclweb.org/anthology/D14-1009

[24] A. Stolcke and E. Shriberg, "Statistical language modeling for speech disfluencies," in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, vol. 1. IEEE, 1996, pp. 405–408.

[25] F. Chollet, "Keras: Deep learning library for theano and tensorflow," 2015.

[26] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[27] I. De Kok and D. Heylen, "Multimodal end-of-turn prediction in multi-party meetings," in *Proceedings of the 11th International Conference on Multimodal Interfaces*. ICMI, 11 2009.