An Incremental Turn-Taking Model For Task-Oriented Dialog Systems

Andrei C. Coman[1], *Koichiro Yoshino*[2], *Yukitoshi Murase*[2], *Satoshi Nakamura*[2], *Giuseppe Riccardi*[1]

[1]Department of Information Engineering and Computer Science
University of Trento, Italy
[2]Graduate School of Information Science
Nara Institute of Science and Technology, Japan

andreicatalin.coman@studenti.unitn.it, giuseppe.riccardi@unitn.it
{koichiro, s-nakamura, y-murase}@is.naist.jp

Abstract

In a human-machine dialog scenario, deciding the appropriate time for the machine to take the turn is an open research problem. In contrast, humans engaged in conversations are able to timely decide when to interrupt the speaker for competitive or non-competitive reasons. In state-of-the-art *turn-by-turn* dialog systems the decision on the next dialog action is taken at the end of the utterance. In this paper, we propose a *token-by-token* prediction of the dialog state from incremental transcriptions of the user utterance. To identify the point of maximal understanding in an ongoing utterance, we a) implement an incremental Dialog State Tracker which is updated on a token basis (iDST) b) re-label the Dialog State Tracking Challenge 2 (DSTC2) dataset and c) adapt it to the incremental turn-taking experimental scenario. The re-labeling consists of assigning a binary value to each token in the user utterance that allows to identify the appropriate point for taking the turn. Finally, we implement an incremental Turn Taking Decider (iTTD) that is trained on these new labels for the turn-taking decision. We show that the proposed model can achieve a better performance compared to a deterministic handcrafted turn-taking algorithm.

Index Terms: Incremental Dialog State Tracking, Incremental Turn-Taking Decider, Dialog Systems, Recurrent Neural Networks, Long Short-Term Memory

1. Introduction

The creation of dialog systems capable of holding conversations at the same level of naturalness as those between people is still a challenge and is far from being considered solved, in spite of the numerous studies in this field. A conversation is to be considered productive when the emphasis is given not only to the content that is conveyed, but also to the moment in which this exchange takes place. Within this paper, we address both issues by developing an incremental system capable of tracking the content of the conversation and identifying the appropriate moment for replying to the human counterpart.

In contrast to previous studies based on *turn-by-turn* systems [1, 2, 3, 4, 5, 6, 7, 8], which generate system utterances only after detecting the end of the input from the user, we place ourselves in a more challenging situation, where the system is expected to generate an utterance after each token coming from an ongoing user utterance, thus relying on its incremental processing [9, 10].

While we believe that there should be a harmonious integration between prosodic signals and lexical features [11, 12], in this work we will focus only on the utterance transcription in order to explore its potential in an incremental setting. Lexical features can be employed not only for determining the turn-taking point, but also as a decision point of where to start the post-processing phase, which could include dialog management and response generation.

Each interlocutor who takes part in a dialog ”maintains” a so-called internal state of the dialog. This state is enriched with new information or updated during the evolution of the conversation itself. In the case of a human-machine conversation, the machine must be able to maintain a description of the human counterpart’s intentions, including the grounding in the domain semantics. Consequently, an effective dialog system must be equipped with a tracker able to accumulate evidence over the sequence of utterances in the dialog and update the dialog state according to the observations. This state directly influences the behavior of the machine and its capability of identifying the point of maximal understanding of an ongoing user utterance. In an incremental scenario, the system can conduct post-processing and provide related responses as soon as it receives new information from the human counterpart. However, there is a trade-off between how early the turn is taken and the dialog state tracker accuracy. The latter will be reduced if the remaining part of the user utterance turns out to be informative. We thus define a new problem which consists of predicting the balance point, that trades off the accuracy reduction and the incremental processing.

We build our incremental Dialog State Tracker (iDST) by taking as reference the *LecTrack* model [9]. The *iDST* was used as a starting block for the implementation of our incremental Turn-Taking Decider (iTTD), which in turn is responsible for identifying the best point that balances the accuracy of the *iDST* and the early turn-taking moment using the least amount of tokens possible. We exploit the dialog corpus annotated with dialog state released within the Dialog State Tracking Challenge 2 (DSTC2) [13] as our dialog domain. This dataset only provides the dialog state at the end of a turn and not for each token, highlighting the need for more granular feedback. To meet this need, we use the *iDST* to identify the balance points within the user utterances. Those points are exploited as new labels for the pre-existing *DSTC2* dataset to train the *iTTD* in a supervised fashion. Source code, trained models and re-labeled dataset are available on GitHub[1].

2. Task Definition

Our final goal is to create an *iTTD* (Subsection 2.2) capable of identifying the point within an ongoing user utterance from where the state of the dialog relative to a specific turn no longer varies, even if new tokens are subsequently provided. To reach

[1] https://github.com/ahclab/iDST

it, we need an *iDST* (Subsection 2.1) to adapt the *DSTC2* dataset to the incremental setting (Subsection 4.1).

2.1. Incremental Dialog State Tracking

The dialog state at time t can be seen as a vector $\mathbf{s}_t \in C_1 \times C_2 \times \cdots \times C_k$ of k dialog state components where each component $c_i \in C_i = \{v_1, \ldots, v_{n_i}\}$ takes one of the n_i values [9]. The goal of a dialog state tracker consists of mapping a sequence of words $w_1, \ldots, w_n$ in a specific dialog state $\mathbf{s}_t$ at time step t. This mapping is equivalent to the estimation of $p(\mathbf{s}_t | w_1, \ldots, w_n)$. The latter can therefore be seen as the estimation of a joint probability over components values:

$$p(\mathbf{s}_t | w_1, \ldots, w_n) = p(c_1, \ldots, c_k | w_1, \ldots, w_n, \theta) \quad (1)$$

or as a product of probabilities over component values:

$$p(\mathbf{s}_t | w_1, \ldots, w_n) = \Pi_i^k p(c_i | w_1, \ldots, w_n, \theta_i) \quad (2)$$

if independence between components holds. In both cases, it is necessary to determine the values of the θ parameters.

A *turn-by-turn* dialog state tracker estimates the dialog state $p(\mathbf{s}_t)$ only after processing all the tokens in the user's utterance. In contrast, a *token-by-token iDST* attempts to estimate the same dialog state $p(\mathbf{s}_t)$ after each token, thus using prefixes of the entire user utterance.

2.2. Incremental Turn-Taking Decider

The dialog system must be able to determine the point when it has enough information for taking the turn. Therefore, this system is able to estimate, for each token in a specific turn, the following joint probability:

$$p(take\_turn | \mathbf{s}_t) = p(0 | \mathbf{s}_t) = p(c_1 = 0, \ldots, c_k = 0 | \mathbf{s}_t, \theta) \quad (3)$$

where $\mathbf{s}_t$ represents the current dialog state estimation and 0 indicates that there is no difference between the current dialog state estimation and the one at the end of the user's utterance. If independence between components holds, the same probability can be seen as:

$$p(take\_turn | \mathbf{s}_t) = p(0 | \mathbf{s}_t) = \Pi_i^k p(c_i = 0 | \mathbf{s}_t, \theta_i) \quad (4)$$

In both cases, the complementary probability $p(wait | \mathbf{s}_t) = p(1 | \mathbf{s}_t) = 1 - p(take\_turn | \mathbf{s}_t)$ is also estimated.

The dichotomous decision made by the *iTTD* should, therefore, reflect the binary labeling that has been performed on the pre-existing *DSTC2* dataset according to the dialog state estimations of the *iDST*. This allows us to create a supervised version of the *iTTD*. Details regarding the labeling function are provided in Subsection 4.1.

3. Proposed Approach

The *iDST* consists of an encoder-based classifier. It takes as input the set of tokens W, which consists of the concatenation of the system output transcript and the ASR 1-best user utterance hypothesis. Each token $w_t \in W$ is associated with a confidence score a_t[2]. To be used by the model, each token is then mapped to its corresponding fixed-size vector representation by means of the embedding function:

$$\mathbf{w}_t = emb(w_t) \quad (5)$$

To reflect the uncertainty of the ASR, a further fully connected layer was added. This layer takes as input the concatenation of the confidence score and the previous embedding representation, thus creating a new embedding:

$$\mathbf{w}'_t = emb\_plus(\mathbf{w}_t, a_t) \quad (6)$$

This representation is then used in conjunction with the previous hidden state $\mathbf{q}_{t-1}$ by an *LSTM* function, which in turn creates a new hidden state as follows:

$$\mathbf{q}_t = LSTM(\mathbf{w}'_t, \mathbf{q}_{t-1}) \quad (7)$$

where

$$\mathbf{q}_{t-1} = (\mathbf{c}_{t-1}, \mathbf{h}_{t-1}) \quad (8)$$

which contains a context vector **c** and a hidden vector **h**. This last vector is then used to compute, by means of a *softmax* layer, the probability distribution over all the possible values that a given component can assume, which can be represented by:

$$p_t = F(\mathbf{h}_t) \quad (9)$$

As for the *iTTD*, it must decide the ideal prefix point within the user's utterance for the dialog state prediction. In doing so, it must use as few tokens as possible while trying to maintain the same performance it would have if the entire utterance were to be used. The decision of whether to take the turn can be modeled as a probability distribution over two binary values by means of a *softmax* layer. The same formulation used in Equation (9) can be replicated for this purpose. Function F takes as input the hidden vector $\mathbf{h}_t$ estimated previously by the *iDST*, but instead of predicting the value of a component, it predicts a binary value that indicates whether the turn should be taken. Further details on data specifications are provided in Section 4.

4. Experiments

4.1. Dataset

The *DSTC2* dataset operates in the restaurant information domain. The dialog state is described by means of three macro-components: *Goal*, *Requested* and *Method*. The first macro-component is defined as the value assumed jointly by four micro-components, namely *Pricerange*, *Area*, *Name* and *Food*.

The dataset is divided into three parts including *train*, *dev* and *test* sets. A brief data analysis, taking into account only the user utterances, is provided in Table 1. The *train-dev* Out-Of-Vocabulary (OOV) rate is equal to 0.21, while the *train-test* one is equal to 0.29. In terms of tokens distribution, all three datasets follow the Zipf's law [14] and have the same top-10 tokens set. Since the dataset in question does not provide any *token-level* feedback, it was necessary to find a way to propagate this information from the *turn-level* to a more granular one. One of the ways in which this can be conducted is via re-labeling the

Table 1: *Dataset analysis based on user utterances*

	Train	**Dev**	**Test**
number of dialogs	1612	506	1117
number of tokens	896	720	892
max. seq. length	28	25	27
avg. tokens per turn	3.88	3.92	3.67
avg. turns per dialog	4.93	5.45	5.98

[2] a_t assumes a fixed value for tokens in the system's utterance. For those that are part of the user's utterance, the value assigned to the full utterance by the ASR 1-best hypothesis is used.

Table 2: *iDST model layers and parameters*

Layer	Parameter	Value
emb	$num\_embeddings$	897
	$embedding\_dim$	170
emb_plus	$in\_features$	171
	$out\_features$	300
LSTM	$input\_size$	300
	$hidden\_size$	100
classifier	$in\_features$	100
	$out\_features$	component dependent
	$activation$	$log\_softmax$ or $sigmoid$

Table 3: *iTTD model layers and parameters*

Layer	Parameter	Value
classifier	$in\_features$	100
	$out\_features$	2
	$activation$	$log\_softmax$

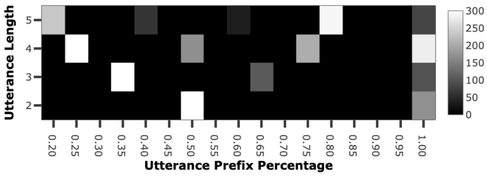

Figure 1: *The frequency (z-axis) with which the turn is taken by the iTTD_ASR(d = 0.85) model is computed based on the user utterance length (y-axis) and the prefix point (x-axis) of the utterance where the actual turn is taken. The frequency has been clipped to 300 for plotting enhancement.*

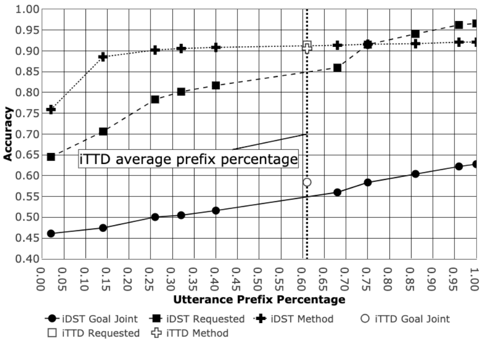

Figure 2: *The accuracy (y-axis) of the three macro-components, namely Goal, Method and Requested, is computed in an incremental fashion based on the prefix points (x-axis) of the user utterance. The iTTD_ASR(d = 0.85) model on average takes the turn at 61% of the user utterance given that all micro-components in the ensemble have a confidence of at least 85% on the predicted 0 label (turn-taking).*

dataset based on the accuracy of the dialog state prediction at the *token-level*. Each token in the user utterance is assigned a binary value by means of Function (10). This label becomes 1 when the dialog state estimated by *iDST* at *i*-th token is different from the one at the last *n*-th token. This also affects the accuracy value, which therefore will change. If the estimation is correct, hence the accuracy value will be the same, the label function assigns the 0 label.

$$label_i = \begin{cases} 1 & \text{if } Acc_i \text{ not equal to } Acc_n \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

Presumably, the first tokens of the user utterance will have label 1, while the last ones will have label 0. The transition point from 1 to 0 indicates a suitable moment for taking the turn. This labeling method is therefore concerned with reflecting the imposed objective, namely minimizing the number of tokens used to predict the dialog state while maintaining the same performance as if the entire utterance were to be used. These new labels, therefore, can be employed for training the *iTTD*. It receives as input, for each token, the hidden vector $\mathbf{h}_t$ coming from the *iDST*, and then tries to predict a binary value that reflects the labels in the new dataset, thus learning the transition point from 1 to 0. If the *iTTD* confidence value on the predicted 0 label is greater than the imposed threshold, the $\mathbf{h}_t$ vector is then simply used by *iDST* in order to estimate the dialog state.

4.2. Experimental setting

The structure of the *iDST* and *iTTD* models, together with their parameters, can be found in Table 2 and 3 respectively. The classifier layer in Table 2 is a fully-connected layer and is the only one that varies according to the individual micro-components of the dialog state. The criterion used for the training procedure is based on the *cross-entropy* loss [15]. Since we are in an incremental *token-by-token* setting, the value of the loss with respect to each turn is accumulated over the user tokens. This differs from a *turn-by-turn* approach, where the loss value is computed only based on the prediction at the last token of the user turn. As an optimizer, we used the AMSGrad [16] variant of the Adam [17] algorithm with $learning\ rate$, β_1, β_2, eps, and $weight\_decay$ set to 0.001, 0.9, 0.999, $1e^{-8}$ and 0 respectively. The metrics used to measure the tracker performance are *accuracy* and *L2-norm*. The first measures the raw 1-best *accuracy* of the ratio of turns in which the tracker's hypothesis is correct. The second, on the other hand, measures the *L2-norm* between the distribution of scores output by the tracker and the label [13].

4.3. Experimental results

iDST and *iTTD* implement Equation (2) and (4) respectively, by creating an ensemble of independent models, each of which refers to one of the micro-components. The results obtained by those models and a comparison with the reference one are reported in Table 4. As a comparative analysis, we also decided to train models which, instead of using the ASR 1-best hypothesis (ASR suffix), use the manual transcription of the user utterances (TRA suffix). Variables r and d in brackets indicate the ratio of used utterance and confidence of turn-taking respectively. Since the length of the user utterances is not fixed, it was necessary to shift the imposed 60% prefix point to the nearest token, which would inevitably alter this imposed percentage. The exact ratio has therefore been computed as the average of the actual percentage values, including shifts. This implies that iDST_ASR(r = 0.6) actually uses, as an average percentage value, 61% and 68% of the user utterance on the *dev* and *test* set respectively, instead of the 60% imposed value. iDST_{ASR, TRA}(r = 1.0) and iDST_{ASR, TRA}(r = 0.6) use respectively 100% and 60% of the user utterance for dialog state estimation. iTTD_{ASR, TRA}(d = 0.85) indicates that all

Table 4: *Models with ASR suffix have been trained with the ASR 1-best user's utterance hypothesis. The ones with TRA suffix have been trained using the manual user utterance transcript. Bold indicates which one among iDST and iTTD prevailed in terms of accuracy.*

	Dev						**Test**					
	Goal		**Method**		**Requested**		**Goal**		**Method**		**Requested**	
Model	**Acc.**	**L2**	**Acc.**	**L2**	**Acc.**	**L2**	**Acc.**	**L2**	**Acc.**	**L2**	**Acc.**	**L2**
LecTrack [9]	0.63	0.74	0.90	0.19	0.96	0.08	0.62	0.75	0.92	0.15	0.96	0.07
iDST_ASR(r = 1.0)	0.64	0.53	0.90	0.17	0.96	0.07	0.63	0.56	0.92	0.13	0.97	0.06
iDST_TRA(r = 1.0)	0.87	0.23	0.94	0.10	0.99	0.02	0.82	0.30	0.94	0.09	0.99	0.02
iDST_ASR(r = 0.6)	0.57	0.61	**0.89**	0.18	0.86	0.23	0.56	0.62	0.91	0.14	0.86	0.21
iTTD_ASR(d = 0.85)	**0.59**	0.60	0.88	0.19	**0.91**	0.16	**0.58**	0.61	0.91	0.15	0.91	0.15
iDST_TRA(r = 0.6)	0.77	0.34	**0.93**	0.11	0.88	0.18	0.73	0.39	**0.94**	0.10	0.88	0.18
iTTD_TRA(d = 0.85)	**0.80**	0.31	0.92	0.12	**0.91**	0.15	**0.76**	0.37	0.93	0.11	**0.91**	0.15

micro-components in the ensemble must have a confidence of at least 85% on the predicted 0 label (turn-taking). Setting the confidence threshold value to 85% causes the iTTD to take the turn on average at 61% (r = 0.61) of the user utterance, thus making it comparable with the deterministic iDST_{ASR, TRA}(r = 0.6). The trend of the accuracy curves obtained by *iDST_ASR(r = 1.0)* relative to the three macro-components, together with the prefix point value selected on average by *iTTD_ASR(d = 0.85)* for the turn-taking decision, is shown in Figure 2. It can, therefore, be observed that the 0.61 ratio prefix point chosen on average by the *iTTD_ASR(d = 0.85)* obtains a better or comparable performance with respect to the deterministic iDST_ASR(r = 0.6) which uses only 60% of the user utterance. The *iDST_TRA* and *iTTD_TRA* models trained on the manual user transcript show how the output of the ASR negatively affects the performance of the models. Figure 1 instead shows in greater detail the frequency with which the *iTTD_ASR(d = 0.85)* decides to take the turn based on the prefix point and the length of the user utterance. For instance, if we consider user utterances of length 2 (e.g. "phone number", "thank you", "good bye", "price range"), it can be observed that a single word, which corresponds to 50% of the utterance, is often enough for the turn taking decision.

5. Related Work

Capability of emulating humans behavior together with naturalness and effectiveness during a conversation, are characteristics that automatic dialogue systems must have. In this sense, several studies have been conducted such as that of [18], which considered user satisfaction through automatic analysis of behavior by measuring emotional states and providing a description as the interaction evolves. The user barge-in problem was addressed by [19], who developed a barge-in-able conversational dialog system that accepts user's barge-in utterances. To coordinate smooth exchange for speaking turns, [20] made use of prosodic, syntactic and gesture features for detecting suitable feedback response locations in the user speech. To cope with incorrectly segmented utterances, [21] proposed an a posteriori restoration methodology. To better understand the behavior of the human counterpart, [22] tried to simulate the user by creating a model that takes into account both her initial goal and responses during the conversation. Despite the numerous improvements introduced, these systems have a common denominator consisting of a relatively rigid structure due to their *turn-by-turn* nature. For a system to be able to replicate human behaviour during a conversation, a paradigm shift is necessary, i.e. the system must be incremental. This means that the system does not need to wait for the end of the user utterance to process it and can, therefore, perform different actions or provide feedback while listening to the human counterpart [23, 24, 25, 26]. To improve the efficiency of the dialogue, [27] defined a turn-taking phenomenon taxonomy, and showed that only some phenomena are worth replicating. ASR and NLU features have been exploited by [11] and [12] in order to detect the end of the turn in an incremental setting. They showed that the combination of prosodic and lexical features can lead to promising results. A turn-taking model based on multitask learning was proposed by [28], which also took into account the prediction of backchannels and fillers. An incremental turn-taking model with active system barge-in was proposed by [29], who modeled the turn-taking problem as a Finite State Machine and learned the turn-taking policy by means of reinforcement learning. Our problem setting is similar to the one posed by [10], where they exploited the ASR and NLU for learning the point of maximal understanding of an ongoing user utterance. In our case, we exploit the sole ASR 1-best hypothesis and the re-labeled dataset, and try to predict the dialog state of the full utterance before it has been completed.

6. Conclusions

In this paper, we proposed a methodology that exploits lexical features to build an automated system capable of estimating the dialog state and the appropriate point for taking the turn in an incremental setting. An automatic re-labeling method that allows for the propagation of *turn-level* feedback to a more granular *token-level* one was introduced. Thanks to these labels we were able to create a system that, on average, performs better than a deterministic decider concerning the turn-taking problem. The decision of the threshold value regarding the confidence of the *iTTD* model still remains a hyperparameter that must be manually set. This limitation opens up for future work that can focus on this problem. In addition, it is evident that there is a need for a metric capable of measuring the turn-taking performance, which is specific to the incremental setting. As future work, we would like to replace our supervised model for the decision on turn-taking with a version based on reinforcement learning. The aim is to analyze how much this methodology is able to manage the adversative relationship between minimizing the number of needed tokens and maximizing the performance.

7. Acknowledgments

The research leading to these results has received funding from the European Union — H2020 Programme under grant agreement 826266: COADAPT.

8. References

[1] M. Henderson, B. Thomson, and S. Young, "Word-based dialog state tracking with recurrent neural networks," in *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, 2014, pp. 292–299.

[2] R. Smith, "Comparative error analysis of dialog state tracking," in *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, 2014, pp. 300–309.

[3] K. Sun, L. Chen, S. Zhu, and K. Yu, "The sjtu system for dialog state tracking challenge 2," in *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, 2014, pp. 318–326.

[4] B.-J. Lee, W. Lim, D. Kim, and K.-E. Kim, "Optimizing generative dialog state tracker via cascading gradient descent," in *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, 2014, pp. 273–281.

[5] O. Plátek, P. Bělohlávek, V. Hudeček, and F. Jurčíček, "Recurrent neural networks for dialogue state tracking," *arXiv preprint arXiv:1606.08733*, 2016.

[6] K. Yoshino, T. Hiraoka, G. Neubig, and S. Nakamura, "Dialogue state tracking using long short term memory neural networks," in *Proceedings of Seventh International Workshop on Spoken Dialog Systems*, 2016, pp. 1–8.

[7] R. Masumura, T. Tanaka, A. Ando, R. Ishii, R. Higashinaka, and Y. Aono, "Neural dialogue context online end-of-turn detection," in *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, 2018, pp. 224–228.

[8] N. G. Ward and D. DeVault, "Ten challenges in highly-interactive dialog system," in *2015 AAAI Spring Symposium Series*, 2015.

[9] L. Zilka and F. Jurcicek, "Incremental lstm-based dialog state tracker," in *2015 Ieee Workshop on Automatic Speech Recognition and Understanding (Asru)*. IEEE, 2015, pp. 757–762.

[10] D. DeVault, K. Sagae, and D. Traum, "Can i finish?: learning when to respond to incremental interpretation results in interactive dialogue," in *Proceedings of the SIGDIAL 2009 Conference: The 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics, 2009, pp. 11–20.

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

[12] A. Raux and M. Eskenazi, "Optimizing endpointing thresholds using dialogue features in a spoken dialogue system," in *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*. Association for Computational Linguistics, 2008, pp. 1–10.

[13] M. Henderson, B. Thomson, and J. D. Williams, "The second dialog state tracking challenge," in *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, 2014, pp. 263–272.

[14] D. M. Powers, "Applications and explanations of zipf's law," in *Proceedings of the joint conferences on new methods in language processing and computational natural language learning*. Association for Computational Linguistics, 1998, pp. 151–160.

[15] R. Y. Rubinstein and D. P. Kroese, *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer Science & Business Media, 2013.

[16] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," *arXiv preprint arXiv:1904.09237*, 2019.

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

[18] S. A. Chowdhury, E. A. Stepanov, and G. Riccardi, "Predicting user satisfaction from turn-taking in spoken conversations." in *INTERSPEECH*, 2016, pp. 2910–2914.

[19] K. Matsuyama, K. Komatani, T. Ogata, and H. G. Okuno, "Enabling a user to specify an item at any time during system enumeration-item identification for barge-in-able conversational dialogue systems," in *Tenth Annual Conference of the International Speech Communication Association*, 2009.

[20] R. Meena, G. Skantze, and J. Gustafson, "Data-driven models for timing feedback responses in a map task dialogue system," *Computer Speech & Language*, vol. 28, no. 4, pp. 903–922, 2014.

[21] K. Komatani, N. Hotta, S. Sato, and M. Nakano, "User-adaptive a posteriori restoration for incorrectly segmented utterances in spoken dialogue systems," *Dialogue & Discourse*, vol. 8, no. 2, pp. 206–224, 2017.

[22] I. Gur, D. Hakkani-Tür, G. Tür, and P. Shah, "User modeling for task oriented dialogues," *CoRR*, vol. abs/1811.04369, 2018. [Online]. Available: http://arxiv.org/abs/1811.04369

[23] M. K. Tanenhaus, M. J. Spivey-Knowlton, K. M. Eberhard, and J. C. Sedivy, "Integration of visual and linguistic information in spoken language comprehension," *Science*, vol. 268, no. 5217, pp. 1632–1634, 1995.

[24] D. Schlangen and G. Skantze, "A general, abstract model of incremental dialogue processing," in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2009, pp. 710–718.

[25] K. Dohsaka and A. Shimazu, "System architecture for spoken utterance production in collaborative dialogue," in *Working Notes of IJCAI 1997 Workshop on Collaboration, Cooperation and Conflict in Dialogue Systems*, 1997.

[26] J. Allen, G. Ferguson, and A. Stent, "An architecture for more realistic conversational systems," in *Proceedings of the 6th international conference on Intelligent user interfaces*. ACM, 2001, pp. 1–8.

[27] H. Khouzaimi, R. Laroche, and F. Lefevre, "Turn-taking phenomena in incremental dialogue systems," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1890–1895.

[28] K. Hara, K. Inoue, K. Takanashi, and T. Kawahara, "Prediction of turn-taking using multitask learning with prediction of backchannels and fillers," *Listener*, vol. 162, p. 364, 2018.

[29] T. Zhao, A. W. Black, and M. Eskenazi, "An incremental turn-taking model with active system barge-in for spoken dialog systems," in *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2015, pp. 42–50.