BARGE-IN EFFECTS IN BAYESIAN DIALOGUE ACT RECOGNITION AND SIMULATION

Heriberto Cuayáhuitl, Nina Dethlefs, Helen Hastie, Oliver Lemon

School of Mathematical and Computer Sciences
Heriot-Watt University, Edinburgh, United Kingdom
{h.cuayahuitl,n.s.dethlefs,h.hastie,o.lemon}@hw.ac.uk

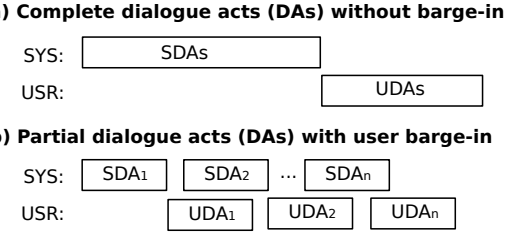
ABSTRACT

Dialogue act recognition and simulation are traditionally considered separate processes. Here, we argue that both can be fruitfully treated as interleaved processes within the same probabilistic model, leading to a synchronous improvement of performance in both. To demonstrate this, we train multiple Bayes Nets that predict the *timing* and *content* of the next user utterance. A specific focus is on providing support for barge-ins. We describe experiments using the Let's Go data that show an improvement in classification accuracy (+5%) in Bayesian dialogue act recognition involving barge-ins using partial context compared to using full context. Our results also indicate that simulated dialogues with user barge-in are more realistic than simulations without barge-in events.

Index Terms— spoken dialogue systems, dialogue act recognition, dialogue simulation, Bayesian nets, barge-in

1. INTRODUCTION AND MOTIVATION

Modelling dialogue phenomena incrementally has been highlighted as one of the (remaining) challenges for spoken dialogue systems [1, 2]. Whereas non-incremental architectures wait until the end of an incoming user utterance before starting to process it, incremental ones become active as soon as the first input units are available. In addition, whereas non-incremental architectures assume communication based on complete dialogue acts, incremental ones assume communication based on partial dialogue acts. This difference is illustrated in the figure below and has been shown to account for shorter processing times and higher user acceptance [3].



This research was funded by the European Commission FP7 programme FP7/2011-14 under grant agreement no. 287615 (PARLANCE).

In this paper, we focus on user dialogue act recognition and user simulation for spoken dialogue systems at the semantic level. The former involves mapping a set of features of the Automatic Speech Recognition (ASR) input and dialogue history onto a unique user dialogue act. The latter are typically used for training system policies which require a large amount of dialogue data. While both problems are well investigated for the non-incremental case, little work exists on incremental approaches; see [4, 5] for some first advances. More work in this direction is therefore needed to enhance the efficiency and quality of spoken dialogue systems. In addition, previous work has so far neglected the fact that user dialogue act recognition and simulation can often be treated fruitfully within the same probabilistic model. Given a dialogue act recogniser which finds the most likely user dialogue act based on the history of previous system and user dialogue context, we can treat simulation as an equivalent problem: given a history of system and user dialogue context, what action is the user most likely to perform next? This double-function model is advantageous because an improved dialogue act recognition accuracy automatically leads to improved realism of simulated dialogues[1]. Our solution here consists of using multiple statistical classifiers that predict both *when* the user will start speaking next (which can be at any point during a system utterance) as well as *what* the user is most likely going to say. This paper describes and analyses our proposed approach, advocating recognition and simulation from the same set of statistical models. A particular focus will be on recognising and simulating barge-ins in natural interaction.

2. RELATED WORK

2.1. Incremental Natural Language Understanding

Related work in incremental language understanding has focused on finding the intended semantics in a user utterance as soon as possible while the user is still speaking. This has been shown to lead to faster system responses and increased human acceptability. [6] were among the first to demonstrate

[1] The converse may not apply, however, especially if the simulations apply some constraints or distortions.

that incremental understanding is not only substantially faster than its non-incremental counterpart, but is also significantly preferred by human judges. [7] use a classifier to map ASR input features onto (full) semantic frames from partial inputs. They show that better results can be achieved by training the classifier from partial dialogue acts rather than from full dialogue acts. [8] present an incremental parser which finds the best semantic interpretation based on syntactic and pragmatic constraints, especially taking into account whether a candidate parse has a denotation in the current context. [9] perform incremental language understanding taking visual, discourse and linguistic context into account. They show that while employing an incremental analysis module provides some benefits, hypotheses are not always stable, especially at early processing states. Our results extend previous work in analysing the performance of dialogue act recognition and simulation in dialogue turns with and without barge-in events.

2.2. User Simulation for Incremental Dialogue

Despite the growing popularity of incremental architectures, little work exists on incremental simulation. Some authors have used non-incremental simulations to optimize incremental dialogue or language generation systems [10, 11]. Similarly, [12] discuss the option of integrating POMDP-based dialogue managers with incremental processing, but leave the question of simulation unaddressed. [5] present a first model of incremental simulation, but focus exclusively on the problem of turn taking. Given the increased interest in incremental processing, the absence of incremental phenomena in user simulations represents an important limitation.

2.3. Statistical Dialogue Act Recognition and Simulation

Approaches to (non-incremental) dialogue act recognition from spoken input have explored a wide range of different methods and feature sets. [13] use Hidden Markov Models (HMMs) in a joint segmentation and classification model. [14] also use HMMs but explore decision trees and neural networks in addition. Several authors have explored Bayesian methods, such as Naive Bayes classification [15] and Bayesian Networks [16]. [17] use Bayesian networks to re-rank dialogue acts from n-best lists. Other authors have used Support Vector Machines (SVMs), such as [18] (who use a combination of SVMs and HMMs) or [19] who show that an active learning framework for dialogue act classification can outperfom passive learning. [20] use max-ent classification. A wide range of feature sets have also been explored, including discourse and sentence features [14, 21], multi-level information features [22], affective facial expression features [23], or speaker-specific features [24].

In terms of dialogue act simulation, a similarly wide range of methods has been investigated. Several authors have explored Bayesian methods, such as [25] who use Bayesian Networks (BNs) to estimate user actions independently from natural language understanding. Similarly, [26] use Bayesian Networks that can deal with missing data in simulation and [27] use a dynamic BN in an unsupervised learning setting. Other graphical models for simulation have been used by [28], who compare different types of HMMs and [29] who use conditional random fields. [30] use an agenda-based model for user simulation, in which the agenda is represented by a stack of user goals which needs to be satisfied before successfully achieving the next dialogue goal. The agenda can be treated as hidden from the dialogue manager to represent the uncertainty that also exists with respect to real users. This model has been extended to be trainable directly from real users rather than corpora [31]. Other methods have been based on "advanced" n-grams [32], clustering [33], or random selection of user utterances from a corpus [34]. Finally, some authors model the user as an agent similar to the dialogue manager [35] or use inverse reinforcement learning to simulate user actions [36]. For a detailed overview of the different user simulations see [37, 38] and [39] for an overview of possible evaluation metrics.

3. A BARGE-IN BASED APPROACH FOR DIALOGUE ACT RECOGNITION AND SIMULATION

The sort of dialogue act recognition (also referred to as shallow semantic parsing in the literature) that we aim for takes into account dialogue act types, attributes and values. An example dialogue act is *confirm(to=Pittsburgh Downtown)*. While dialogue act recognition and user simulation are typically treated as separate components, our approach is based on the premise that user simulation can make use of a statistical dialogue act recogniser to generate user responses. This is possible by sampling from the estimated probability distributions induced by the dialogue act recogniser. The user simulations in this case, would be mimicking the user responses from some training data, but with more variation for wider coverage in terms of conversational behaviours. To make simulations account for unseen situations, the statistical models would have to include all possible combinations of dialogue act types and slot value pairs with non-zero probabilities. Algorithm 1 shows a fairly generic dialogue act recogniser assuming multiple statistical classifiers $\lambda = \{\lambda^{dat}, \lambda^{att}, \lambda^{val(i)}\}$ with features $X = \{x_1, ..., x_n\}$, labels $Y = \{y_1, ..., y_n\}$, and the evidence $\mathbf{e} = \{x_1{=}val(x_1), ..., x_n{=}val(x_n)\}$ collected during the system-user interaction. While model λ^{dat} is used to predict system dialogue act types, model λ^{att} is used to predict attributes, and the remaining models ($\lambda^{val(i)}$) are used to predict slot value pairs. In our case we use Bayes Nets (BNs) to obtain the most likely label y (i.e. a dialogue act type, attribute, or value) from domain values $\mathcal{D}(Y)$ expressed as

$$y^* = \arg \max_{y \in \mathcal{D}(Y)} P(y|\mathbf{e}).$$

For incremental simulators, it is important to model when

Algorithm 1 Statistical recogniser of user dialogue acts

function DIALOGACTRECOGNITION(StatisticalModels λ, Evidence $\mathbf{e}$)
  $\lambda^{dat} \leftarrow$ statistical model for predicting dialogue act types (DAT)
  $\lambda^{att} \leftarrow$ statistical model for predicting attributes (ATT)
  $\lambda^{val(i)} \leftarrow$ statistical models for predicting attribute values (VAL)
  $y \leftarrow$ output (class label) of the statistical classifier in turn
  $dat = \arg\max_{y \in \mathcal{D}(DAT)} P(y|\mathbf{e}; \lambda^{dat})$
  $att = \arg\max_{y \in \mathcal{D}(ATT)} P(y|\mathbf{e}; \lambda^{att})$
  $pairs \leftarrow []$
  for each attribute i in att **do**
    $val = \arg\max_{y \in \mathcal{D}(VAL^{(i)})} P(y|\mathbf{e}; \lambda^{val(i)})$
    $pairs \leftarrow$ APPEND(attribute i=val)
  end for
  return $dat(pairs)$
end function

Algorithm 2 Statistical simulator of user dialogue acts

function DIALOGACTSIMULATION(StatisticalModels λ, Evidence $\mathbf{e}$)
  $\lambda^{ut} \leftarrow$ statistical model to predict when the user takes the turn (UT)
  $\lambda^{dat} \leftarrow$ statistical model for predicting dialogue act types (DAT)
  $\lambda^{att} \leftarrow$ statistical model for predicting attributes (ATT)
  $\lambda^{val(i)} \leftarrow$ statistical models for predicting attribute values (VAL)
  $y \leftarrow$ output (class label) of the statistical classifier in turn
  $ut = \arg\max_{y \in \mathcal{D}(UT)} P(y|\mathbf{e}; \lambda^{ut})$
  if ut is true **then**
    $dat \leftarrow$ sample from $P(Y|\mathbf{e}; \lambda^{dat})$
    $att \leftarrow$ sample from $P(Y|\mathbf{e}; \lambda^{att})$
    $pairs \leftarrow []$
    for each attribute i in att **do**

$$val = \begin{cases} \text{with probability } \epsilon\text{, get value from user goal } g(i) \\ \text{with probability } 1-\epsilon\text{, sample from } P(Y|\mathbf{e}; \lambda^{val^{(i)}}) \end{cases}$$

      pairs $\leftarrow$ APPEND(attribute i=val)
    end for
    return $dat(pairs)$
  end if
  return $null$
end function

the user speaks assuming that system dialogue acts are received incrementally. For example, should the user speak after the first, second, ..., or last system dialogue act? This event can be modelled probabilistically as

$$ut^* = \arg \max_{ut \in \{0,1\}} P(ut|\mathbf{e}; \lambda^{ut}),$$

where ut is a binary value and λ^{ut} in our case is an additional statistical model (Bayes net) in our set of classifiers λ. If the result ut^* is true then the main body of Algorithm 2 is invoked, otherwise a null event is returned. Our approach queries (with observed features) the set of Bayes nets incrementally after each partial system dialogue act.

In the rest of the paper, we analyse a corpus of dialogues using Algorithms 1 and 2 for dialogue act recognition and simulation. The simulations below focus on *when* the user speaks and *what* they say. For goal-directed user dialogue acts, the slot values can be derived from user goal g with probability ϵ and from models $\lambda^{val(i)}$ with probability 1-ϵ.

4. EXPERIMENTS AND RESULTS

4.1. Data

Our experiments are based on the Let's Go corpus [40]. Let's Go contains recorded interactions between a spoken dialogue system and human users who make enquiries about the bus schedule in Pittsburgh. Dialogues are system-initiative and query the user sequentially for five slots: *an optional bus route, a departure place, a destination, a desired travel date,* and *a desired travel time*. Each slot needs to be explicitly (or implicitly) confirmed by the user. Our analyses are based on a subset of this data set containing 779 dialogues with 7275 turns, collected in Summer of 2010. From these dialogues, we used 70% for training our classifiers and the rest for testing (with 50 random splits). Brief statistics of this data set are as follows. Table 1 shows all dialogue act types that occur in the data together with their frequency of occurrence. System dialogue acts are shown on top and user dialogue acts in the bottom. Table 2 shows the main attribute types for the dialogue acts again paired with their frequency of use by system and user. Notice that the combination of all possible dialogue acts, attributes and values leads to a large number of triplets. While a whole dialogue act is represented as a sequence of tuples <*dialogue_act*(*attribute*=*value* pairs)>, a partial dialogue act is represented as <*dialogue_act*(*attribute*=*value* pair)>.

4.2. Statistical Classifiers

We trained our statistical classifiers in a supervised learning manner, and used 43 discrete features plus a class label (also discrete), see Table 3. The feature set is described by three main subsets: 24 system-utterance-level binary features derived from the system dialogue act(s) in the last turn; 16 user-utterance-level binary features derived from (a) what the user heard prior to the current turn, or (b) what keywords the system recognised in its list of speech recognition hypotheses; and 4 non-binary features corresponding to the last system dialogue act type, duration in seconds, previous and current label. Predicted labels are restricted to those that occur in the N-best parsing hypotheses from the Let's Go data and an additional dialogue act "silence". See [17] for details.

Figure 1 shows the Bayesian network corresponding to the classifier that predicts when the user speaks, queried incrementally after each partial system dialogue act. The structures of our Bayesian classifiers were derived from the K2 algorithm[2], their parameters were derived from maximum likelihood estimation, and probabilistic inference using the Junction tree algorithm[3]. We trained a set of 14 Bayesian classifiers to predict (1) when the user speaks, (2) the dialogue act

[2] http://www.cs.waikato.ac.nz/ml/weka/
[3] http://www.cs.cmu.edu/~javabayes/Home/

Agent	Dialogue Act Type	Frequency (%)
Sys	ack	0.52
Sys	canthelp	1.45
Sys	example	59.04
Sys	expl-conf	5.23
Sys	goback	0.37
Sys	hello	1.90
Sys	impl-conf	8.71
Sys	morebuses	0.46
Sys	request	11.37
Sys	restart	0.25
Sys	schedule	3.44
Sys	sorry	7.24
Usr	affirm	14.10
Usr	bye	1.35
Usr	goback	2.36
Usr	inform	41.68
Usr	negate	5.02
Usr	nextbus	11.00
Usr	prevbus	1.63
Usr	repeat	3.83
Usr	restart	1.44
Usr	silence	17.36
Usr	tellchoices	0.22

Table 1. *Frequencies of dialogue act types in our data set.*

Attribute (slot)	System Freq. (%)	User Freq. (%)
date.absday	0.50	0.38
date.absmonth	0.50	0.38
date.day	1.62	4.71
date.relweek	0.41	0.0
from	26.26	24.44
route	36.70	33.36
time.ampm	1.73	2.45
time.arriveleave	1.67	2.91
time.hour	2.19	3.60
time.minute	2.19	3.60
time.rel	2.80	0.31
to	23.41	23.86

Table 2. *Frequencies of system and user slots in our data set.*

type, (3) the attributes (also called 'slots'), and (4) the slot values. The advantage of using multiple Bayes Nets over just one is that a multiple classifier system is a powerful solution to complex classification problems involving a large set of inputs and outputs. This approach not only decreases training time but has also been shown to increase the performance of classification [41].

4.3. Evaluation Metrics

The accuracy of dialogue act recognition is computed as the proportion of correct classifications among the all classifications. The comparison is made against labelled gold standard data from human annotations.

We compute the quality of simulations with the Kullback-Leibler (KL) divergence [42], which measures the similarity between a gold standard data set and a target data set.

Type	Features (^{b}=binary, ^{nb}=non-binary)
System	heardAck^b, heardCantHelp^b, heardExample^b, heardExplConf^b, heardGoBackDAT^b, heardHello^b, heardImplConf^b, heardMoreBuses^b, heardRequest^b, heardRestartDAT^b, heardSchedule^b, heardSorry^b, heardDate^b, heardFrom^b, heardRoute^b, heardTime^b, heardTo^b, heardNext^b, heardPrevious^b, heardGoBack^b, heardChoices^b, heardRestart^b, heardRepeat^b, heardDontKnow^b, lastSystemDialActType^{nb}, duration^{nb} (in seconds: 0,1,2,3,4,>5), currentLabel (e.g. userSpeaks^b, dialActType^{nb}, slot$_i^{nb}$), prevLabel
User	hasRoute^b, hasFrom^b, hasTo^b, hasDate^b, hasTime^b, hasYes^b, hasNo^b, hasNext^b, hasPrevious^b, hasGoBack^b, hasChoices^b, hasRestart^b, hasRepeat^b, hasDontKnow^b, hasBye^b, hasNothing^b.

Table 3. *Features for dialogue act recognition & simulation.*

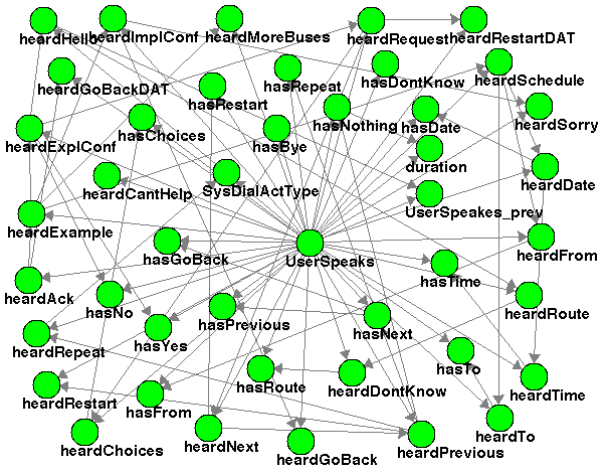


Fig. 1. *Bayesian network for predicting when the user speaks.*

4.4. Experimental Results in Dialogue Act Recognition

Our dialogue act recognition results compared 4 different recognisers with and without barge-in events: (a) *Semi-Random*: a recogniser choosing a random dialogue act from the Let's Go N-best parsing hypotheses; (b) *LetsGo*: a recogniser choosing the most likely dialogue act from the Let's Go N-best parsing hypotheses; (c) *Bayes Nets*: a Bayesian recogniser using Algorithm 1; and (d) *Ceiling*: a recogniser choosing the correct dialogue act from the Let's Go N-best parsing hypotheses. The latter was used as a gold standard from manual annotations, which reflects the proportion of correct labels in the N-best parsing hypotheses. Figure 2 shows the dialogue act recognition results in this order, which can be described as follows. First, we can observe that recognition accuracy in dialogue turns without user barge-in events consistently performs better than its counterpart with barge-ins (significant at $p<0.003$)[4]. Second, it can be noted that the Let's Go baseline is substantially outperformed by the Bayesian recogniser (also significant at $p<0.004$). This is

[4]Based on a 2-tailed Wilcoxon signed rank test.

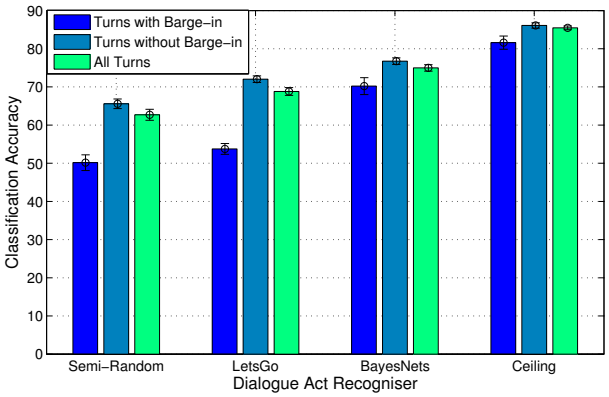


Fig. 2. *Dialogue act recognition results with(out) barge-in.*

partially due to the fact that the Let's Go system always tries to recognise a dialogue act even when there was only noise in the environment, which causes the ASR to produce incorrect recognition hypotheses. Our Bayesian classifiers model probability distributions over dialogue acts that are closer to a human gold standard than several baselines. Third, we compared the performance of Bayesian dialogue act recognition of turns with barge-in based on partial and full context (see Figure 3). Whilst partial context considers evidence until the barge-in point, full context considers evidence based on all system dialogue acts within a turn. This comparison revealed that recognition using partial context improves its counterpart with full context by 4.6% (significant at $p<0.0024$). This suggests that the degradation in recognition of turns with barge-in can be mitigated with incremental processing, i.e. context from partial dialogue acts. These results are relevant for spoken dialogue systems because they suggest how to achieve more efficient interactions: in our data set the average duration of system turns with barge-in events is 8.6 seconds, and the average duration of system turns without barge-in events is 10 seconds[5], in favour of incremental processing. This could potentially improve the user experience by making spoken dialogue systems more accurate in the face of user barge-ins, leading to more timely relevant responses.

4.5. Experimental Results in Dialogue Act Simulation

Our simulation results compared Let's Go system-user dialogue act tuples (from correct labels) against simulated dialogues with and without barge-in events. The latter, a typical type of simulations in the literature, represents our baseline. We consider tuples of dialogue act type and attribute names (but without attribute values to avoid the data sparsity problem). While tuples without barge-in considered evidence until the very last system dialogue act in the turn, tuples with barge-in considered evidence until a barge-in. The exact point of a barge-in over a system dialogue act is not logged in the data. Barge-ins (11.5% in our data) were extracted from the data based on the overlap time between system and user turns. The partial system dialogue act with the overlap is marked as the point of barge-in. These results are shown in Table 4. It can be noted that the simulated dialogues with barge-in events (compared with real Let's Go dialogues) obtain lower divergences than its counterpart without barge-in events. This result suggests that dialogue simulators should incorporate user barge-ins based on partial system dialogue acts rather than complete ones to achieve more realistic simulated interactions.

[5]Since our data only had durations per system turn rather than per partial dialogue act, we estimated such durations using a linear regressor based on TTS durations. Predictor variables for the linear regressor: numerical ID of dialogue act type, number of slots, number of words, number of characters.

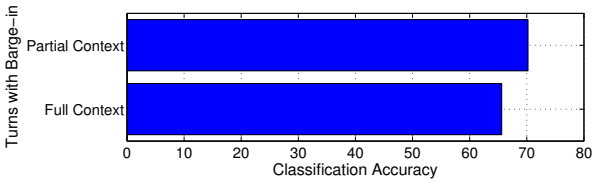


Fig. 3. *Bayesian dialogue act recognition with partial and full context, based on turns with only barge-in. The bar with partial context is the 7th bar in Figure 2 from left to right.*

Classifier (simulator)	Turns	Divergence	p-value
Bayesian Networks	with barge-in	5.1731	<0.0074
	without barge-in	5.4123	

Table 4. KL divergences (the smaller the better) between dialogue turns with and without barge-in events.

5. CONCLUSION AND FUTURE WORK

We have presented an approach to incremental user dialogue act recognition and simulation which treats both problems as interleaved processes within the same probabilistic model. Multiple classifiers are used to (a) predict dialogue acts from dialogue history features, and (b) predict when the user should speak after each partial system dialogue act. Applying our approach to the Let's Go data we found the following. First, we found an improvement in classification accuracy (+5%) in Bayesian dialogue act recognition involving barge-ins using partial context compared to using full context. Second, dialogue simulation taking into account user barge-in events represent more realistic interactions than their counterpart without barge-in events. This should be a feature of dialogue simulators used for training future dialogue systems.

Future work includes a comparison of our Bayesian classifiers with other statistical models and forms of training (for example by using semi-supervised learning) [43], and investigating the effects of barge-in on dialogue act recognisers and simulators in different (multi-modal) domains [44, 45].

6. REFERENCES

[1] Victor Zue and James Glass, "Conversational interfaces: Advances and challenges," vol. 88, no. 8, pp. 1166–1180, 2000.

[2] Antoine Raux, *Flexible Turn-Taking for Spoken Dialog Systems*, Ph.D. thesis, School of Computer Science, Carnegie Mellon University, 2008.

[3] Gabriel Skantze and David Schlangen, "Incremental Dialogue Processing in a Micro-Domain," in *EACL*, Athens, Greece, 2009.

[4] Volha Petukhova and Harry Bunt, "Incremental Dialogue Act Understanding," in *IWCS*, Oxford, UK, 2011.

[5] Ethan Selfridge and Peter Heeman, "A Temporal Simulator for Developing Turn-Taking Methods for Spoken Dialogue Systems," in *SIGDIAL*, Seoul, South Korea, 2012.

[6] Gregory Aist, James Allen, Ellen Campana, Carlos Gomez Gallo, Scott Stoness, Mary Swift, and Michael Tanenhaus, "Incremental Understanding in Human-Computer Dialogue and Experimental Evidence for Advantages over Nonincremental Methods," in *DECOLOG*, Trento, Italy, 2007.

[7] David DeVault, Kenji Sagae, and David Traum, "Can I Finish? Learning When to Respond to Incremental Interpretation Results in Interactive Dialogue," in *SIGDIAL*, London, UK, 2009.

[8] Andreas Peldszus, Okko Buss, Timo Baumann, and David Schlangen, "Joint Satisfaction of Syntactic and Pragmatic Constraints Improves Incremental Spoken Language Understanding," in *EACL*, Avignon, France, 2012.

[9] Casey Kennington and David Schlangen, "Markov Logic Networks for Situated Incremental Natural Language Understanding," in *SIGDIAL*, Seoul, South Korea, 2012.

[10] Nina Dethlefs, Helen Hastie, Verena Rieser, and Oliver Lemon, "Optimising Incremental Generation for Spoken Dialogue Systems: Reducing the Need for Fillers," in *INLG*, Chicago, IL, USA, 2012.

[11] Nina Dethlefs, Helen Hastie, Verena Rieser, and Oliver Lemon, "Optimising Incremental Dialogue Decisions Using Information Density for Interactive Systems," in *EMNLP-CoNLL*, Jeju, South Korea, 2012.

[12] Ethan Selfridge, Iker Arizmendi, Peter Heeman, and Jason Williams, "Integrating Incremental Speech Recognition and POMDP-based Dialogue Systems," in *SIGDIAL*, Seoul, South Korea, 2012.

[13] Matthias Zimmermann, Yang Liu, Elizabeth Shriberg, and Andreas Stolcke, "Toward Joint Segmentation and Classification of Dialog Acts in Multiparty Meetings," in *MLMI*, 2005, pp. 187–193.

[14] Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca A. Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer, "Dialog Act Modeling for Automatic Tagging and Recognition of Conversational Speech," *Comp. Linguistics*, vol. 26, no. 3, pp. 339–373, 2000.

[15] Sergio Grau, Emilio Sanchis, Maria Jose Castro, and David Vilar, "Dialogue Act Classification Using a Bayesian Approach," in *SPECOM*, 2004.

[16] Simon Keizer and Rieks op den Akker, "Dialogue Act Recognition Under Uncertainty Using Bayesian Networks," *Natural Language Engineering*, vol. 13, no. 4, pp. 287–316, 2007.

[17] Heriberto Cuayáhuitl, Nina Dethlefs, Helen Hastie, and Oliver Lemon, "Impact of ASR N-Best Information on Bayesian Dialogue Act Recognition," in *SIGDIAL*, 2013.

[18] Dinoj Surendran and Gina-Anne Levow, "Dialog Act Tagging with Support Vector Machines and Hidden Markov Models," in *INTERSPEECH*, 2006.

[19] Björn Gambäck, Fredrik Olsson, and Oscar Täckström, "Active Learning for Dialogue Act Classification," in *INTERSPEECH*, 2011, pp. 1329–1332.

[20] Vivek Kumar Rangarajan Sridhar, Srinivas Bangalore, and Shrikanth Narayanan, "Combining Lexical, Syntactic and Prosodic Cues for Improved Online Dialog Act Tagging," *Computer Speech & Language*, vol. 23, no. 4, pp. 407–422, 2009.

[21] Keyan Zhou and Chengqing Zong, "Dialog-Act Recognition Using Discourse and Sentence Structure Information," in *IALP*, 2009, pp. 11–16.

[22] Tina Klüwer, Hans Uszkoreit, and Feiyu Xu, "Using Syntactic and Semantic based Relations for Dialogue Act Recognition," in *COLING*, 2010, pp. 570–578.

[23] Kristy Elizabeth Boyer, Joseph F. Grafsgaard, Eunyoung Ha, Robert Phillips, and James C. Lester, "An Affect-Enriched Dialogue Act Classification Model for Task-Oriented Dialogue," in *ACL*, 2011, pp. 1190–1199.

[24] Congkai Sun and Louis-Philippe Morency, "Dialogue Act Recognition Using Reweighted Speaker Adaptation," in *SIGDIAL*, 2012, pp. 118–125.

[25] Olivier Pietquin and Thierry Dutoit, "A Probabilistic Framework for Dialog Simulation and Optimal Strategy Learning," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 14, no. 2, pp. 589–599, 2006.

[26] Stéphane Rossignol, Olivier Pietquin, and Michel Ianotto, "Training a BN-Based User Model for Dialogue Simulation with Missing Data," in *IJCNLP*, Chiang Mai, Thailand, November 2011, pp. 598–604.

[27] Sungjin Lee and Maxine Eskenazi, "An Unsupervised Approach to User Simulation: Toward Self-Improving Dialog Systems," in *SIGDIAL*, 2012, pp. 50–59.

[28] Heriberto Cuayáhuitl, Steve Renals, Oliver Lemon, and Hiroshi Shimodaira, "Human-Computer Dialogue Simulation Using Hidden Markov Models," in *ASRU*, San Juan, Puerto Rico, Dec 2005, pp. 290–295.

[29] Sangkeun Jung, Cheongjae Lee, Kyungduk Kim, Minwoo Jeong, and Gary Geunbae Lee, "Data-Driven User Simulation for Automated Evaluation of Spoken Dialog Systems," *Computer Speech & Lang.*, vol. 23, no. 4, pp. 479–509, 2009.

[30] Jost Schatzmann and Steve Young, "The Hidden Agenda User Simulation Model," *IEEE Transactions on Speech, Audio and Language Processing*, vol. 17, no. 4, pp. 733–747, 2009.

[31] Simon Keizer, Milica Gasic, Filip Jurcícek, François Mairesse, Blaise Thomson, Kai Yu, and Steve Young, "Parameter Estimation for Agenda-Based User Simulation," in *SIGDIAL*, 2010, pp. 116–123.

[32] Kallirroi Georgila, James Henderson, and Oliver Lemon, "User Simulation for Spoken Dialogue Systems: Learning and Evaluation," in *INTERSPEECH*, Pittsburgh, PA, USA, Sep 2006, pp. 267–659.

[33] V. Rieser and O. Lemon, "Cluster-Based User Simulations for Learning Dialogue Strategies," in *INTERSPEECH*, Pittsburgh, PA, USA, Sep 2006, pp. 1766–1769.

[34] R. López-Cózar, Z. Callejas, and M. McTear, "Testing the Performance of Spoken Dialogue Systems by Means of an Artificial User," *Artificial Intelligence Review*, vol. 26, no. 4, pp. 291–323, 2008.

[35] F. Torres, E. Sanchis, and E. Segarra, "User Simulation in a Stochastic Dialog System," *Computer Speech and Language*, vol. 22, no. 3, pp. 230–255, 2008.

[36] Senthilkumar Chandramohan, Matthieu Geist, Fabrice Lefèvre, and Olivier Pietquin, "User Simulation in Dialogue Systems Using Inverse Reinforcement Learning," in *INTERSPEECH*, 2011, pp. 1025–1028.

[37] J. Schatzmann, K. Weilhammer, M. Stuttle, and S. Young, "A Survey on Statistical User Simulation Techniques for Reinforcement Learning of Dialogue Management Strategies," *Knowledge Eng. Review*, vol. 21, no. 2, pp. 97–126, 2006.

[38] Hua Ai and Diane J. Litman, "Comparing User Simulations for Dialogue Strategy Learning," *TSLP*, vol. 7, no. 3, pp. 9, 2011.

[39] Olivier Pietquin and Helen Hastie, "A Survey on Metrics for the Evaluation of User Simulations," *Knowledge Engineering Review*, vol. 28, no. 01, pp. 59–73, 2013.

[40] Antoine Raux, Brian Langner, Dan Bohus, Alan W. Black, and Maxine Eskenazi, "Let's go public! Taking a Spoken Dialog System to the Real World," in *INTERSPEECH*, 2005, pp. 885–888.

[41] David M. Tax, Martijn van Breukelen, Robert P. Duin, and Josef Kittler, "Combining multiple classifiers by averaging or by multiplying?," *Pattern Recognition*, vol. 33, no. 9, pp. 1475–1485, Sept. 2000.

[42] Heriberto Cuayáhuitl, Steve Renals, Oliver Lemon, and Hiroshi Shimodaira, "Evaluation of a hierarchical reinforcement learning spoken dialogue system," *Computer Speech and Language*, vol. 24, no. 2, pp. 395–429, 2010.

[43] Heriberto Cuayáhuitl, Martijn van Otterlo, Nina Dethlefs, and Lutz Frommberger, "Machine learning for interactive systems and robots: a brief introduction," in *MLIS*, 2013, pp. 19–28.

[44] Heriberto Cuayáhuitl and Nina Dethlefs, "Optimizing situated dialogue management in unknown environments," in *INTERSPEECH*, 2011, pp. 1009–1012.

[45] Heriberto Cuayáhuitl and Ivana Kruijff-Korbayová, "An interactive humanoid robot exhibiting flexible sub-dialogues," in *HLT-NAACL*, 2012, pp. 17–20.