

# Importance-Driven Turn-Bidding for Spoken Dialogue Systems

Ethan O. Selfridge and Peter A. Heeman

Center for Spoken Language Understanding

Oregon Health & Science University

20000 NW Walker Rd., Beaverton, OR, 97006

*selfridg@ohsu.edu, heemanp@ohsu.edu*

## Abstract

Current turn-taking approaches for spoken dialogue systems rely on the speaker releasing the turn before the other can take it. This reliance results in restricted interactions that can lead to inefficient dialogues. In this paper we present a model we refer to as *Importance-Driven Turn-Bidding* that treats turn-taking as a negotiative process. Each conversant bids for the turn based on the importance of the intended utterance, and Reinforcement Learning is used to indirectly learn this parameter. We find that Importance-Driven Turn-Bidding performs better than two current turn-taking approaches in an artificial collaborative slot-filling domain. The negotiative nature of this model creates efficient dialogues, and supports the improvement of mixed-initiative interaction.

## 1 Introduction

As spoken dialogue systems are designed to perform ever more elaborate tasks, the need for mixed-initiative interaction necessarily grows. Mixed-initiative interaction, where agents (both artificial and human) may freely contribute to reach a solution efficiently, has long been a focus of dialogue systems research (Allen et al., 1999; Guinn, 1996). Simple slot-filling tasks might not require the flexible environment that mixed-initiative interaction brings but those of greater complexity, such as collaborative task completion or long-term planning, certainly do (Ferguson et al., 1996). However, translating this interaction into working systems has proved problematic (Walker et al., 1997), in part to issues surrounding turn-taking: the transition from one speaker to another.

Many computational turn-taking approaches seek to minimize silence and utterance overlap

during transitions. This leads to the speaker controlling the turn transition. For example, systems using the Keep-Or-Release approach will not attempt to take the turn unless it is sure the user has released it. One problem with this approach is that the system might have important information to give but will be unable to get the turn. The speaker-centric nature of current approaches does not enable mixed-initiative interaction and results in inefficient dialogues. Primarily, these approaches have been motivated by smooth transitions reported in the human turn-taking studies of Sacks et al. (1974) among others.

Sacks et al. also acknowledge the negotiative nature of turn-taking, stating that the “the turn as unit is interactively determined”(p. 727). Other studies have supported this, suggesting that humans negotiate the turn assignment through the use of cues and that these cues are motivated by the importance of what the conversant wishes to contribute (Duncan and Niederehe, 1974; Yang and Heeman, 2010; Schegloff, 2000). Given this, any dialogue system hoping to interact with humans efficiently and naturally should have a negotiative and importance-driven quality to its turn-taking protocol. We believe that, by focusing on the rationale of human turn-taking behavior, a more effective turn-taking system may be achieved. We propose the Importance-Driven Turn-Bidding (IDTB) model where conversants bid for the turn based on the importance of their utterance. We use Reinforcement Learning to map a given situation to the optimal utterance and bidding behavior. By allowing conversants to bid for the turn, the IDTB model enables negotiative turn-taking and supports true mixed-initiative interaction, and with it, greater dialogue efficiency.

We compare the IDTB model to current turn-taking approaches. Using an artificial collaborative dialogue task, we show that the IDTB model enables the system and user to complete

the task more efficiently than the other approaches. Though artificial dialogues are not ideal, they allow us to test the validity of the IDTB model before embarking on costly and time-consuming human studies. Since our primary evaluation criteria is model comparison, consistent user simulations provide a constant needed for such measures and increase the external validity of our results.

## 2 Current Turn-Taking Approaches

Current dialogue systems focus on the release-turn as the most important aspect of turn-taking, in which a listener will only take the turn after the speaker has released it. The simplest of these approaches only allows a single utterance per turn, after which the turn necessarily transitions to the next speaker. This Single-Utterance (SU) model has been extended to allow the speaker to keep the turn for multiple utterances: the Keep-Or-Release (KR) approach. Since the KR approach gives the speaker sole control of the turn, it is overwhelmingly speaker-centric, and so necessarily unnegotiative. This restriction is meant to encourage smooth turn-transitions, and is inspired by the order, smoothness, and predictability reported in human turn-taking studies (Duncan, 1972; Sacks et al., 1974).

Systems using the KR approach differ on how they detect the user’s release-turn. Turn releases are commonly identified in two ways: either using a silence-threshold (Sutton et al., 1996), or the predictive nature of turn endings (Sacks et al., 1974) and the cues associated with them (e.g. Gravano and Hirschberg, 2009). Raux and Eskenazi (2009) used decision theory with lexical cues to predict appropriate places to take the turn. Similarly, Jonsdottir, Thorisson, and Nivel (2008) used Reinforcement Learning to reduce silences between turns and minimize overlap between utterances by learning the specific turn-taking patterns of individual speakers. Skantze and Schlangan (2009) used incremental processing of speech and prosodic turn-cues to reduce the reaction time of the system, finding that that users rated this approach as more human-like than a baseline system.

In our view, systems built using the KR turn-taking approach suffer from two deficits. First, the speaker-centricity leads to inefficient dialogues since the speaker may continue to hold the turn even when the listener has vital information to give. In addition, the lack of negotiation forces

the turn to necessarily transition to the listener after the speaker releases it. The possibility that the dialogue may be better served if the listener *does not* get the turn is not addressed by current approaches.

Barge-in, which generally refers to allowing users to speak at any time (Ström and Seneff, 2000), has been the primary means to create a more flexible turn-taking environment. Yet, since barge-in recasts speaker-centric systems as user-centric, the system’s contributions continue to be limited. System barge-in has also been investigated. Sato et al. (2002) used decision trees to determine whether the system should take the turn or not when the user pauses. An incremental method by DeVault, Sagae, and Traum (2009) found possible points that a system could interrupt without loss of user meaning, but failed to supply a reasonable model as to when to use such information. Despite these advances, barge-in capable systems lack a negotiative turn-taking method, and continue to be deficient for reasons similar to those described above.

## 3 Importance-Driven Turn-Bidding (IDTB)

We introduce the IDTB model to overcome the deficiencies of current approaches. The IDTB model has two foundational components: (1) The importance of speaking is the primary motivation behind turn-taking behavior, and (2) conversants use turn-cue strength to bid for the turn based on this importance. Importance may be broadly defined as how well the utterance leads to some predetermined conversational success, be it solely task completion or encompassing a myriad of social etiquette components.

Importance-Driven Turn-Bidding is motivated by empirical studies of human turn-conflict resolution. Yang and Heeman (2010) found an increase of turn conflicts during tighter time constraints, which suggests that turn-taking is influenced by the importance of task completion. Schlegoff (2000) proposed that persistent utterance overlap was indicative of conversants having a strong interest in holding the turn. Walker and Whittaker (1990) show that people will interrupt to remedy some understanding discrepancy, which is certainly important to the conversation’s success. People communicate the importance of their utterance through turn-cues. Duncan and

Niederehe (1974) found that turn-cue strength was the best predictor of who won the turn, and this finding is consistent with the use of volume to win turns found by Yang and Heeman (2010).

The IDTB model uses turn-cue strength to bid for the turn based on the importance of the utterance. Stronger turn-cues should be used when the intended utterance is important to the overall success of the dialogue, and weaker ones when it is not. In the prototype described in Section 5, both the system and user agents bid for the turn after every utterance and the bids are conceptualized here as utterance onset: conversants should be quick to speak important utterances but slow with less important ones. This is relatively consistent with Yang and Heeman (2010). A mature version of our work will use cues in addition to utterance onset, such as those recently detailed in Gravano and Hirshberg (2009).<sup>1</sup>

A crucial element of our model is the judgment and quantization of utterance importance. We use Reinforcement Learning (RL) to determine importance by conceptualizing it as maximizing the reward over an entire dialogue. Whatever actions lead to a higher return may be thought of as more important than ones that do not.<sup>2</sup> By using RL to learn both the utterance and bid behavior, the system can find an optimal pairing between them, and choose the best combination for a given conversational situation.

#### 4 Information State Update and Reinforcement Learning

We build our dialogue system using the Information State Update approach (Larsson and Traum, 2000) and use Reinforcement Learning for action selection (Sutton and Barto, 1998). The system architecture consists of an Information State (IS) that represents the agent’s knowledge and is updated using a variety of rules. The IS also uses rules to propose possible actions. A condensed and compressed subset of the IS — the Reinforcement Learning State — is used to learn which proposed action to take (Heeman, 2007). It has been shown that using RL to learn dialogue policies is generally more effective than “hand crafted” di-

<sup>1</sup>Our work (present and future) is distinct from some recent work on user pauses (Sato et al., 2002) since we treat turn-taking as an integral piece of dialogue success.

<sup>2</sup>We gain an inherent flexibility in using RL since the reward can be computed by a wide array of components. This is consistent with the broad definition of importance.

ologue policies since the learning algorithm may capture environmental dynamics that are unintended to by human designers (Levin et al., 2000).

Reinforcement Learning learns an optimal policy, a mapping between a state  $s$  and action  $a$ , where performing  $a$  in  $s$  leads to the lowest expected cost for the dialogue (we use minimum cost instead of maximum reward). An  $\epsilon$ -greedy search is used to estimate Q-scores, the expected cost of some state–action pair, where the system chooses a random action with  $\epsilon$  probability and the  $\text{argmin}_a Q(s, a)$  action with  $1-\epsilon$  probability. For Q-learning, a popular RL algorithm and the one used here,  $\epsilon$  is commonly set at 0.2 (Sutton and Barto, 1998). Q-learning updates  $Q(s, a)$  based on the best action of the next state, given by the following equation, with the step size parameter  $\alpha = 1/\sqrt{N(s, a)}$  where  $N(s, a)$  is the number of times the  $s, a$  pair has been seen since the beginning of training.

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[\text{cost}_{t+1} + \text{argmin}_a Q(s_{t+1}, a) - Q(s_t, a_t)]$$

The state space should be formulated as a Markov Decision Process (MDP) for Q-learning to update Q-scores properly. An MDP relies on a first-order Markov assumption in that the transition and reward probability from some  $s_t, a_t$  pair is completely contained by that pair and is unaffected by the history  $s_{t-1}a_{t-1}, s_{t-2}a_{t-2}, \dots$ . For this assumption to be met, care is required when deciding which features to include for learning. The RL State features we use are described in the following section.

#### 5 Domain and Turn-Taking Models

In this section, we show how the IDTB approach can be implemented for a collaborative slot filling domain. We also describe the Single-Utterance and Keep-Or-Release domain implementations that we use for comparison.

##### 5.1 Domain Task

We use a food ordering domain with two participants, the system and a user, and three slots: drink, burger, and side. The system’s objective is to fill all three slots with the available fillers as quickly as possible. The user’s role is to specify its desired filler for each slot, though that specific filler may not be available. The user simulation, while intended to be realistic, is not based on empirical data. Rather, it is designed to provide a rich turn-

taking domain to evaluate the performance of different turn-taking designs. We consider this a collaborative slot-filling task since both conversants must supply information to determine the intersection of available and desired fillers.

Users have two fillers for each slot.<sup>3</sup> A user’s top choice is either available, in which case we say that the user has adequate filler knowledge, or their second choice will be available, in which we say it has inadequate filler knowledge. This assures that at least one of the user’s filler is available. Whether a user has adequate or inadequate filler knowledge is probabilistically determined based on user type, which will be described in Section 5.2.

Table 1: Agent speech acts

Agent	Actions
System	query slot, inform [yes/no], inform avail. slot fillers, inform filler not available, bye
User	inform slot filler, query filler availability

We model conversations at the speech act level, shown in Table 1, and so do not model the actual words that the user and system might say. Each agent has an Information State that proposes possible actions. The IS is made up of a number of variables that model the environment and is slightly different for the system and the user. Shared variables include *QUD*, a stack which manages the questions under discussion; *lastUtterance*, the previous utterance, and *slotList*, a list of the slot names. The major system specific IS variables that are not included in the RL State are *availSlot-Fillers*, the available fillers for each slot; and three *slotFiller* variables that hold the fillers given by the user. The major user specific IS variables are three *desiredSlotFiller* variables that hold an ordered list of fillers, and *unvisitedSlots*, a list of slots that the user believes are unfilled.

The system has a variety of speech actions: *inform [yes/no]*, to answer when the user has asked a filler availability question; *inform filler not available*, to inform the user when they have specified an unavailable filler; three *query slot* actions (one for each slot), a query which asks the user for a filler and is proposed if that specific slot is unfilled;

<sup>3</sup>We use two fillers so as to minimize the length of training. This can be increased without substantial effort.

three *inform available slot fillers* actions, which lists the available fillers for that slot and is proposed if that specific slot is unfilled or filled with an unavailable filler; and *bye*, which is always proposed.

The user has two actions. They can inform the system of a desired slot filler, *inform slot filler*, or query the availability of a slot’s top filler, *query filler availability*. A user will always respond with the same slot as a system query, but may change slots entirely for all other situations. Additional details on user action selection are given in Section 5.2.

Specific information is used to produce an instantiated speech action, what we refer to as an utterance. For example, the speech action *inform slot filler* results in the utterance of “inform drink d1.” A sample dialogue fragment using the Single-Utterance approach is shown in Table 2. Notice that in Line 3 the system informs the user that their first filler, *d1*, is unavailable. The user then asks about the availability of its second drink choice, *d2* (Line 4), and upon receiving an affirmative response (Line 5), informs the system of that filler preference (Line 6).

Table 2: Single-Utterance dialogue

Spkr	Speech Action	Utterance
1 S:	<i>q. slot</i>	q. drink
2 U:	<i>i. slot filler</i>	i. drink d1
3 S:	<i>i. filler not avail</i>	i. not have d1
4 U:	<i>q. filler avail</i>	q. drink have d2
5 S:	<i>i. slot</i>	i. yes
6 U:	<i>i. slot filler</i>	i. drink d2
7 S:	<i>i. avail slot fillers</i>	i. burger have b1

**Implementation in RL:** The system uses RL to learn which of the IS proposed actions to take. In this domain we use a cost function based on dialogue length and the number of slots filled with an available filler:  $C = \text{Number of Utterances} + 25 \cdot \text{unavailablyFilledSlots}$ . In the present implementation the system’s *bye* utterance is costless. The system chooses the action that minimizes the expected cost of the entire dialogue from the current state.

The RL state for the speaker has seven variables:<sup>4</sup> *QUD-speaker*, the stack of speakers who have unresolved questions; *Incorrect-Slot-Fillers*,

<sup>4</sup>We experimented with a variety of RL States and this one proved to be both small and effective.

a list of slot fillers (ordered chronologically on when the user informed them) that are unavailable and have not been resolved; *Last-Sys-Speech-Action*, the last speech action the system performed; *Given-Slot-Fillers*, a list of slots that the system has performed the *inform available slot filler* action on; and three booleans variables, *slot-RL*, that specify whether a slot has been filled correctly or not (e.g. Drink-RL).

## 5.2 User Types

We define three different types of users — *Experts*, *Novices*, and *Intermediates*. User types differ probabilistically on two dimensions: slot knowledge, and slot belief strength. We define experts to have a 90 percent chance of having adequate filler knowledge, intermediates a 50 percent chance, and novices a 10 percent chance. These probabilities are independent between slots. Slot belief strength represents the user’s confidence that it has adequate domain knowledge for the slot (i.e. the top choice for that slot is available). It is either a strong, warranted, or weak belief (Chu-Carroll and Carberry, 1995). The intuition is that experts should know when their top choice is available, and novices should know that they do not know the domain well.

Initial slot belief strength is dependent on user type and whether their filler knowledge is adequate (their initial top choice is available). Experts with adequate filler knowledge have a 70, 20, and 10 percent chance of having Strong, Warranted, and Weak beliefs respectfully. Similarly, intermediates with adequate knowledge have a 50, 25, and 25 percent chance of the respective belief strengths. When these user types have inadequate filler knowledge the probabilities are reversed to determine belief strength (e.g. Experts with inadequate domain knowledge for a slot have a 70% chance of having a weak belief). Novice users always have a 10, 10, and 80 percent chance of the respective belief strengths.

The user chooses whether to use the *query* or *inform* speech action based on the slot’s belief strength. A strong belief will always result in an *inform*, a warranted belief resulting in an *inform* with  $p = 0.5$ , and weak belief will result in an *inform* with  $p = 0.25$ . If the user is informed of the correct fillers by the system’s *inform*, that slot’s belief strength is set to strong. If the user is informed that a filler is not available, than that filler

is removed from the desired filler list and the belief remains the same.<sup>5</sup>

## 5.3 Turn-Taking Models

We now discuss how turn-taking works for the IDTB model and the two competing models that we use to evaluate our approach. The system chooses its turn action based on the RL state and we add a boolean variable *turn-action* to the RL State to indicate when the system is performing a turn action or a speech action. The user uses belief to choose its turn action.

**Turn-Bidding:** Agents bid for the turn at the end of each utterance to determine who will speak next. Each bid is represented as a value between 0 and 1, and the agent with the lower value (stronger bid) wins the turn. This is consistent with the use of utterance onset. There are 5 types of bids, *highest*, *high*, *middle*, *low*, and *lowest*, which are spread over a portion of the range as shown in Figure 1. The system uses RL to choose a bid and a random number (uniform distribution) is generated from that bid’s range. The users’ bids are determined by their belief strength, which specifies the mean of a Gaussian distribution, as shown in Figure 1 (e.g Strong belief implies a  $\mu = 0.35$ ). Computing bids in this fashion leads to, on average, users with strong beliefs bidding *highest*, warranted beliefs bidding in the *middle*, and weak beliefs bidding *lowest*. The use of the probability distributions allows us to randomly decide ties between system and user bids.

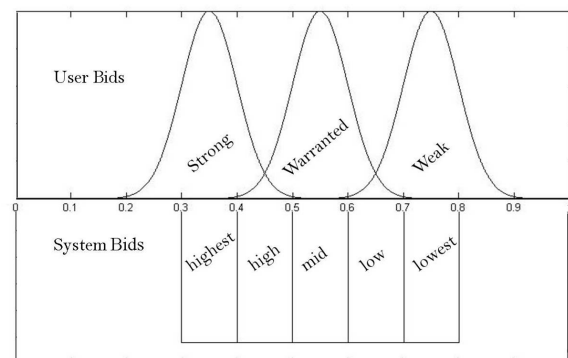


Figure 1: Bid Value Probability Distribution

**Single-Utterance:** The Single-Utterance (SU) approach, as described in Section 2, has a rigid

<sup>5</sup>In this simple domain the next filler is guaranteed to be available if the first is not. We do not model this with belief strength since it is probably not representative of reality.

turn-taking mechanism. After a speaker makes a single utterance the turn transitions to the listener. Since the turn transitions after every utterance the system must only choose appropriate utterances, not turn-taking behavior. Similarly, user agents do not have any turn-taking behavior and slot beliefs are only used to choose between a *query* and an *inform*.

**Keep-Or-Release Model:** The Keep-Or-Release (KR) model, as described in Section 2, allows the speaker to either keep the turn to make multiple utterances or release it. Taking the same approach as English and Heeman (2005), the system learns to keep or release the turn after each utterance that it makes. We also use RL to determine which conversant should begin the dialogue. While the use of RL imparts some importance onto the turn-taking behavior, it is not influencing whether the system gets the turn when it did not already have it. This is an crucial distinction between KR and IDTB. IDTB allows the conversants to negotiate the turn using turn-bids motivated by importance, whereas in KR only the speaker determines when the turn can transition.

Users in the KR environment choose whether to keep or release the turn similarly to bid decisions.<sup>6</sup> After a user performs an utterance, it chooses the slot that would be in the next utterance. A number,  $k$ , is generated from a Gaussian distribution using belief strength in the same manner as the IDTB users' bids are chosen. If  $k \leq 0.55$  then the user keeps the turn, otherwise it releases it.

#### 5.4 Preliminary Turn-Bidding System

We described a preliminary turn-bidding system in earlier work presented at a workshop (Selfridge and Heeman, 2009). A major limitation was an overly simplified user model. We used two user types, expert and novice, who had fixed bids. Experts always bid *high* and had complete domain knowledge, and the novices always bid *low* and had incomplete domain knowledge. The system, using all five bid types, was always able to out bid and under bid the simulated users. Among other things, this situation gives the system complete control of the turn, which is at odds with the negotiative nature of IDTB. The present contribution is a more realistic and mature implementation.

<sup>6</sup>We experimented with a few different KR decision strategies, and chose the one that performed the best.

## 6 Evaluation and Discussion

We now evaluate the IDTB approach by comparing it against the two competing models: Single-Utterance and Keep-Or-Release. The three turn-taking approaches are trained and tested in four user conditions: novice, intermediate, expert, and combined. In the combined condition, one of the three user types is randomly selected for each dialogue. We train ten policies for each condition and turn-taking approach. Policies are trained using Q-learning, and  $\epsilon$ -greedy search for 10000 epochs (1 epoch = 100 dialogues, after which the Q-scores are updated) with  $\epsilon = 0.2$ . Each policy is then ran over 10000 test dialogues with no exploration ( $\epsilon = 0$ ), and the mean dialogue cost for that policy is determined. The 10 separate policy values are then averaged to create the mean policy cost. The mean policy cost between the turn-taking approaches and user conditions are shown in Table 3. Lower numbers are indicative of shorter dialogues, since the system learns to successfully complete the task in all cases.

Table 3: Mean Policy Cost for Model and User condition<sup>7</sup>

Model	Novice	Int.	Expert	Combined
SU	7.61	7.09	6.43	7.05
KR	<b>6.00</b>	6.35	4.46	6.01
IDTB	6.09	<b>5.77</b>	<b>4.35</b>	<b>5.52</b>

**Single User Conditions:** Single user conditions show how well each turn-taking approach can optimize its behavior for specific user populations and handle slight differences found in those populations. Table 3 shows that the mean policy cost of the SU model is higher than the other two models which indicates longer dialogues on average. Since the SU system must respond to every user utterance and cannot learn a turn-taking strategy to utilize user knowledge, the dialogues are necessarily longer. For example, in the expert condition the best possible dialogue for a SU interaction will have a cost of five (three user utterances for each slot, two system utterances in response). This cost is in contrast to the best expert dialogue cost of three (three user utterances) for KR and IDTB interactions.

The IDTB turn-taking approach outperforms the KR design in all single user conditions ex-

<sup>7</sup>SD between policies  $\leq 0.04$

cept for novice (6.09 vs. 6.00). In this condition, the KR system takes the turn first, informs the available fillers for each slot, and then releases the turn. The user can then inform its filler easily. The IDTB system attempts a similar dialogue strategy by using *highest* bids but sometimes loses the turn when users also bid *highest*. If the user uses the turn to query or inform an unavailable filler the dialogue grows longer. However, this is quite rare as shown by small difference in performance between the two models. In all other single user conditions, the IDTB approach has shorter dialogues than the KR approach (5.77 and 4.35 vs. 6.35 and 4.46). A detailed explanation of IDTB’s performance will be given in Section 6.1.

**Combined User Condition:** We next measure performance on the combined condition that mixes all three user types. This condition is more realistic than the other three, as it better mimics how a system will be used in actual practice. The IDTB approach (mean policy cost = 5.52) outperforms the KR (mean policy cost = 6.01) and SU (mean policy cost = 7.05) approaches. We also observe that KR outperforms SU. These results suggest that the more a turn-taking design can be flexible and negotiative, the more efficient the dialogues can be.

**Exploiting User bidding differences:** It follows that IDTB’s performance stems from its negotiative turn transitions. These transitions are distinctly different than KR transitions in that there is information inherent in the users bids. A user that has a stronger belief strength is more likely to be have a higher bid and inform an available filler. Policy analysis shows that the IDTB system takes advantage of this information by using moderate bids —neither *highest* nor *lowest* bids— to filter users based on their turn behavior. The distribution of bids used over the ten learned policies is shown in Table 4. The *initial* position refers to the first bid of the dialogue; *final* position, the last bid of the dialogue; and *medial* position, all other bids. Notice that the system uses either the *low* or *mid* bids as its *initial* policy and that 67.2% of dialogue *medial* bids are moderate. These distributions show that the system has learned to use the entire bid range to filter the users, and is not seeking to win or lose the turn outright. This behavior is impossible in the KR approach.

Table 4: Bid percentages over ten policies in the Combined User condition for IDTB

Position	H-est	High	Mid	Low	L-est
Initial	0.0	0.0	70.0	30.0	0.0
Medial	20.5	19.4	24.5	23.3	12.3
Final	49.5	41.0	9.5	0.0	0.0

## 6.1 IDTB Performance:

In our domain, performance is measured by dialogue length and solution quality. However, since solution quality never affects the dialogue cost for a trained system, dialogue length is the only component influencing the mean policy cost.

The primary cause of longer dialogues are unavailable filler inform and query (UFI-Q) utterances by the user, which are easily identified. These utterances lengthen the dialogue since the system must inform the user of the available fillers (the user would otherwise not know that the filler was unavailable) and then the user must then inform the system of its second choice. The mean number of UFI-Q utterance for each dialogue over the ten learned policies are shown for all user conditions in Table 5. Notice that these numbers are inversely related to performance: the more UFI-Q utterances, the worse the performance. For example, in the combined condition the IDTB users perform 0.38 UFI-Q utterances per dialogue (u/d) compared to the 0.94 UFI-Q u/d for KR users. While a KR user will release the turn if its planned

Table 5: Mean number of UFI-Q utterances over policies

Model	Novice	Int.	Expert	Combined
KR	<b>0.0</b>	1.15	0.53	0.94
IDTB	0.1	<b>0.33</b>	<b>0.39</b>	<b>0.38</b>

utterance has a weak belief, it may select that weak utterance when first getting the turn (either after a system utterance or at the start of the dialogue). This may lead to a UFI-Q utterance. The IDTB system, however, will outbid the same user, resulting in a shorter dialogue. This situation is shown in Tables 6 and 7. The dialogue is the same until utterance 3, where the IDTB system wins the turn with a *mid* bid over the user’s *low* bid. In the KR environment however, the user gets the turn and performs an unavailable filler inform, which the system must react to. This is an instance of the second deficiency of the KR approach, where

Table 6: Sample IDTB dialogue in Combined User condition; Cost=6

Sys	Usr	Spkr	Utt
1 low	mid	U:	inform burger b1
2 h-est	low	S:	inform burger have b3
3 mid	low	S:	inform side have s1
4 mid	h-est	U:	inform burger b3
5 mid	high	U:	inform drink d1
6 l-est	h-est	U:	inform side s1
7 high	mid	S:	bye

Table 7: Sample KR dialogue in Combined User condition; Cost=7

Agent	Utt	Turn-Action
1 U:	inform burger b1	Release
2 S:	inform burger have b3	Release
3 U:	inform side s1	Keep
4 U:	inform drink d1	Keep
5 U:	inform burger b3	Release
6 S:	inform side have s2	Release
7 U:	inform side s2	Release
8 S:	bye	

the speaking system should not have released the turn. The user has the same belief in both scenarios, but the negotiative nature of IDTB enables a shorter dialogues. In short, the IDTB system can win the turn when it should have it, but the KR system cannot.

A lesser cause of longer dialogues is an instance of the first deficiency of the KR systems; the listening user cannot get the turn when it should have it. Usually, this situation presents itself when the user releases the turn, having randomly chosen the weaker of the two unfilled slots. The system then has the turn for more than one utterance, informing the available fillers for two slots. However, the user already had a strong belief and available top filler for one of those slots, and the system has increased the dialogue length unnecessarily. In the combined condition, the KR system produces 0.06 unnecessary informs per dialogue, whereas the IDTB system produces 0.045 per dialogue. The novice and intermediate conditions mirror this (IDTB: 0.009, 0.076 ; KR: 0.019, 0.096 respectively), but the expert condition does not (IDTB: 0.011, KR: 0.0014). In this case, the IDTB system wins the turn initially using a *low* bid and informs one of the strong slots, whereas the expert user initiates the dialogue for the KR environment and un-

necessary informs are rarer. In general, however, the KR approach has more unnecessary informs since the KR system can only infer that one of the user's beliefs was probably weak, otherwise the user would not have released the turn. The IDTB system handles this situation by using a *high* bid, allowing the user to outbid the system as its contribution is more important. In other words, the IDTB user can win the turn when it should have it, but the KR user cannot.

## 7 Conclusion

This paper presented the Importance-Driven Turn-Bidding model of turn-taking. The IDTB model is motivated by turn-conflict studies showing that the interest in holding the turn influences conversant turn-cues. A computational prototype using Reinforcement Learning to choose appropriate turn-bids performs better than the standard KR and SU approaches in an artificial collaborative dialogue domain. In short, the Importance-Driven Turn-Bidding model provides a negotiative turn-taking framework that supports mixed-initiative interactions.

In the previous section, we showed that the KR approach is deficient for two reasons: the speaking system might not keep the turn when it should have, and might release the turn when it should not have. This is driven by KR's speaker-centric nature; the speaker has no way of judging the potential contribution of the listener. The IDTB approach however, due to its negotiative quality, does not have this problem.

Our performance differences arise from situations when the system is the speaker and the user is the listener. The IDTB model also excels in the opposite situation, when the system is the listener and the user is the speaker, though our domain is not sophisticated enough for this situation to occur. In the future we hope to develop a domain with more realistic speech acts and a more difficult dialogue task that will, among other things, highlight this situation. We also plan on implementing a fully functional IDTB system, using an incremental processing architecture that not only detects, but generates, a wide array of turn-cues.

## Acknowledgments

We gratefully acknowledge funding from the National Science Foundation under grant IIS-0713698.



## References

- J.E. Allen, C.I. Guinn, and Horvitz E. 1999. Mixed-initiative interaction. *IEEE Intelligent Systems*, 14(5):14–23.
- Jennifer Chu-Carroll and Sandra Carberry. 1995. Response generation in collaborative negotiation. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 136–143, Morristown, NJ, USA. Association for Computational Linguistics.
- David DeVault, Kenji Sagae, and David Traum. 2009. Can i finish? learning when to respond to incremental interpretation results in interactive dialogue. In *Proceedings of the SIGDIAL 2009 Conference*, pages 11–20, London, UK, September. Association for Computational Linguistics.
- S.J. Duncan and G. Niederehe. 1974. On signalling that it's your turn to speak. *Journal of Experimental Social Psychology*, 10:234–247.
- S.J. Duncan. 1972. Some signals and rules for taking speaking turns in conversations. *Journal of Personality and Social Psychology*, 23:283–292.
- M. English and Peter A. Heeman. 2005. Learning mixed initiative dialog strategies by using reinforcement learning on both conversants. In *Proceedings of HLT/EMNLP*, pages 1011–1018.
- G. Ferguson, J. Allen, and B. Miller. 1996. TRAINS-95: Towards a mixed-initiative planning assistant. In *Proceedings of the Third Conference on Artificial Intelligence Planning Systems (AIPS-96)*, pages 70–77.
- A. Gravano and J. Hirschberg. 2009. Turn-yielding cues in task-oriented dialogue. In *Proceedings of the SIGDIAL 2009 Conference: The 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 253–261. Association for Computational Linguistics.
- C.I. Guinn. 1996. Mechanisms for mixed-initiative human-computer collaborative discourse. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 278–285. Association for Computational Linguistics.
- P.A. Heeman. 2007. Combining reinforcement learning with information-state update rules. In *Proceedings of the Annual Conference of the North American Association for Computational Linguistics*, pages 268–275, Rochester, NY.
- Gudny Ragna Jonsdottir, Kristinn R. Thorisson, and Eric Nivel. 2008. Learning smooth, human-like turntaking in realtime dialogue. In *IVA '08: Proceedings of the 8th international conference on Intelligent Virtual Agents*, pages 162–175, Berlin, Heidelberg. Springer-Verlag.
- S. Larsson and D. Traum. 2000. Information state and dialogue management in the trindi dialogue move engine toolkit. *Natural Language Engineering*, 6:323–340.
- E. Levin, R. Pieraccini, and W. Eckert. 2000. A stochastic model of human-machine interaction for learning dialog strategies. *IEEE Transactions on Speech and Audio Processing*, 8(1):11 – 23.
- A. Raux and M. Eskenazi. 2009. A finite-state turn-taking model for spoken dialog systems. In *Proceedings of HLT/NAACL*, pages 629–637. Association for Computational Linguistics.
- H. Sacks, E.A. Schegloff, and G. Jefferson. 1974. A simplest systematics for the organization of turn-taking for conversation. *Language*, 50(4):696–735.
- R. Sato, R. Higashinaka, M. Tamoto, M. Nakano, and K. Aikawa. 2002. Learning decision trees to determine turn-taking by spoken dialogue systems. In *ICSLP*, pages 861–864, Denver, CO.
- E.A. Schegloff. 2000). Overlapping talk and the organization of turn-taking for conversation. *Language in Society*, 29:1 – 63.
- E. O. Selfridge and Peter A. Heeman. 2009. A bidding approach to turn-taking. In *1st International Workshop on Spoken Dialogue Systems*.
- G. Skantze and D. Schlagen. 2009. Incremental dialogue processing in a micro-domain. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 745–753. Association for Computational Linguistics.
- N. Ström and S. Seneff. 2000. Intelligent barge-in in conversational systems. In *Sixth International Conference on Spoken Language Processing*. Citeseer.
- R. Sutton and A. Barto. 1998. *Reinforcement Learning*. MIT Press.
- S. Sutton, D. Novick, R. Cole, P. Vermeulen, J. de Villiers, J. Schalkwyk, and M. Fenty. 1996. Building 10,000 spoken-dialogue systems. In *ICSLP*, Philadelphia, Oct.
- M. Walker and S. Whittaker. 1990. Mixed initiative in dialogue: an investigation into discourse segmentation. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, pages 70–76.
- M. Walker, D. Hindle, J. Fromer, G.D. Fabbrizio, and C. Mestel. 1997. Evaluating competing agent strategies for a voice email agent. In *Fifth European Conference on Speech Communication and Technology*.
- Fan Yang and Peter A. Heeman. 2010. Initiative conflicts in task-oriented dialogue”. *Computer Speech Language*, 24(2):175 – 189.